DESIGN AND DEVELOPMENT OF A

COMPOSITE FREQUENCY RESPONSE ANALYSER

DAVID LEE JONES

A thesis submitted in partial fulfilment of the requirements of the Council for National Academic Awards for the degree of Doctor of Philosophy

July 1990

Polytechnic of Wales

ACKNOWLEDGEMENTS

The author wishes to thank Dr. J.D. Davies, Director of the Polytechnic of Wales and Professor P.A. Witting, Head of Department, Electronics and Information Technology, for permitting the investigation to be undertaken.

I am indebted to my supervisors, Dr. D. Rees, Deputy Head of Department of Electronics and Information Technology at the Polytechnic of Wales, and Mr. W.A. Evans, Dean of Faculty/Reader, Department of Electrical and Electronic Engineering, University College of Swansea, for their helpful guidance, encouragement and enthusiasm during the course of this investigation.

I am grateful to a number of my colleagues, particularly Mr. R. Stone and Mr. S. Gardner, for interesting and stimulating discussion.

I am thankful to Mrs. S. Morgan and Miss C. Bell, secretarial staff at the Department of Electronics and Information Technology, for their help in the typing of this thesis.

My final thanks go to my wife and children for the patience they have shown during the many hours I have spent on the investigation, which otherwise would have been spent with them.

CONTENTS

DESIGN AND DEVELOPMENT OF A COMPOSITE FREQUENCY

RESPONSE ANALYSER

D.L.JONES, B.Sc.,C.Eng.,M.I.E.E.

SYNOPSIS

This thesis examines the role that composite frequency test signals play in system identification. The advantages and disadvantages of composite frequency response analysers over conventional analysers are discussed. Two low peak-factor composite frequency test signals are reviewed which largely overcome the problems associated with frequency response analysers which employ more established test waveforms.

A novel, high performance frequency response analyser incorporating these two composite waveforms has been fully designed and developed. The instrument which makes use of a 256 point Cooley-Tukey Fast Fourier Transform, demonstrates the feasibility of producing an analyser based upon the new waveforms which has a performance equal to, or greater than existing commercial analysers.

The relative performance of the two test waveforms has been evaluated by applying them to a number of real systems and comparing the results with those obtained when using a commercial monotonic analyser. In addition to linear systems, tests have been carried out on systems exhibiting non-linear behaviour in an attempt to establish the ability of the instrument to minimise harmonic distortion in the measurement process.

The theoretical basis for the frequency response analysis has been developed and the potential problems associated with aliasing have been analysed. In order to avoid the complexity and expense associated with anti-aliasing filters however, no such filter has been incorporated within the instrument. In place of a filter, a new anti-aliasing compensation technique has been developed. This technique is largely based upon a software algorithm and offers significant advantages over existing analogue/digital filtering techniques. The theoretical basis for the technique is developed, and a method for its mechanisation within the developed instrument is presented.

This anti-aliasing algorithm has been incorporated within the analyser and has been applied to the frequency response measurement of known systems using a composite frequency test waveform. The results are compared to those obtained without using the algorithm to demonstrate the effectiveness of the new technique.

NOMENCLATURE

ADC	Analogue to Digital Converter
Ai	Amplitude at i^{th} frequency component
ATE	Automatic Test Equipment
CPU	Central Processing Unit
DAC	Digital to Analogue Converter
DFT	Discrete Fourier Transform
$d_0(t)$, $d_1(t)$	Train of impulse functions
$D_0(f)$, $D_0(f)$	Spectrum of impulse function train
E_n	RMS value of quantisation noise
E_t	RMS value of test waveform
EPROM	Erasable Programmable Read Only Memory
FFT	Fast Fourier Transform
FIFO	First In First Out
f	Frequency variable
g(t)	Impulse Response of SUT
G	Constant gain
G(f)	Transfer function of the SUT in the frequency domain
h(t)	Impulse response of zero order hold DAC
H(f)	Transfer function of zero order hold DAC
I/O	Input/Output
I/P	Input
K	System gain
k_1, k_2	Cross quadrature correlator outputs
MPRBS	Modified Pseudo Random Binary Signal
MPU	Microprocessing Unit
n	Sample number variable

N	Sample size
PRBS	Pseudo Random Binary Sequence
Q	Digitiser resolution
RAM	Random Access Memory
SUT	System Under Test
s(t)	Instrument test waveform
S(f)	Spectrum of the instrument test waveform
S*(f)	Spectrum of sampled instrument test waveform
$S_i(f)$	Spectrum of periodic pulse waveform
$S_i^*(f)$	Spectrum of sampled periodic pulse waveform
T_c	Shift register clock period
T_o	Waveform period
T	Sample waveform period
t	Time variable
UART	Universal Asynchronous Receive and Transmit
u(t)	Sampled output signal of the SUT
U(f)	Spectrum of sampled output signal of the SUT
VDU	Visual Display Unit
w	Angular frequency
x(t)	Output signal of the SUT
X(f)	Spectrum of the output signal of the SUT
y(t)	Reconstructed signal applied to the SUT
Y(f)	Spectrum of the reconstructed signal applied to the SUT
β	Saturation figure
δ	The impulse function
θ	Phase angle
γ	System time constant

CHAPTER 1

INTRODUCTION

1.1 Identification of Linear Systems

A fundamental component of any control system design is the identification of a model of the dynamic characteristic of the system. This model may be expressed in the time domain as an impulse response function or in the frequency domain as a transfer function. Although both representations give complementary information about the dynamic behaviour of the system it is the transfer function of the system which is most widely adopted for use in traditional control design methodologies.

Generally, the transfer function of a system may be directly identified from a knowledge of the frequency response of a system, which in turn may be obtained by measurement of the systems input and output signals. The choice of input signal in such an analysis of the system's dynamic behaviour is critical, since it can affect the accuracy and the speed at which the system response is obtained.

In some applications the choice of input stimulus is restricted by the operating environment and as such it is impossible to excite the system with an arbitrary input signal. However, in the main, the only restriction on the excitation signal is its maximum allowed peak to peak amplitude.

The conventional method for measuring the frequency response of a system employs a sinusoid as the input stimulus.

Developments in signal processing algorithms and hardware however, have made it possible to utilise more complex input test signals which have a broadband spectrum (Composite frequency test signals). These signals allow several spectral estimates to be measured simultaneously and this results in an important reduction in the overall measurement time.

The mechanisation of such composite waveform test procedures however, rely on the use of digital to analogue conversion techniques for test signal generation, and analogue to digital conversion techniques for the capture of the system output. Care must be taken in the application of such techniques since an undesirable loss of accuracy can result unless special precautions are taken.

In general therefore, a successful identification method must incorporate an accurate measurement of the system frequency response which should be obtained rapidly, with little disturbance to the process. Furthermore, identification is often undertaken on systems which are noisy and/or exhibit slight non-linear behaviour and as such it is desirable for the measurement procedure to provide both noise and harmonic rejection capability.

1.2 Identification Techniques

1.2.1 Monotonic Frequency Response Analysis.

Conventional monotonic frequency response analysers (1,2,3) are widely used in the measurement of system dynamics.

The monotonic technique utilised by these analysers however has the disadvantage of a long measurement time when testing systems with large time constants. Using this approach, the system to be tested is excited by a sine wave of known amplitude and phase. After a delay which is governed by the settling time of the system, the system output is captured and from this a single spectral estimate of the system frequency response is calculated. This procedure needs to be repeated for every frequency of interest. The technique is therefore serial by nature and the time taken to obtain the response of a system is given by

$$\text{Measurement time} > mt_s + t_i \qquad 1.1$$

where

m = the number of spectral estimates required

t_s = the settling time of the system

t_i = the period of the waveform at the i^{th} measurement

The above expression demonstrates that if the system under test has large time constants then obtaining the response of the system over a number of frequencies can be time consuming and even prohibitive.

Since the test waveform employed is periodic then the effect of random noise on the measurement may be minimised by integrating over many cycles of the test waveform (4). This averaging, however, results in a further increase in measurement time.

One benefit of the single frequency testing approach is that should the system under test exhibit non-linear behaviour (a quasi-linear system) then corruption of the spectral estimate due to harmonic distortion can be avoided. This is achieved by employing quadrature cross-correlation to calculate the system response (4).

1.2.2 Impulse Response Analysis

To overcome the larger testing times associated with monotonic frequency response analysis much attention in the literature has been directed towards obtaining the time domain equivalent of the frequency response - the impulse response.

To directly measure the impulse response, the system under test should be perturbed by an impulse function.

There are two main difficulties associated with direct impulse response testing however

(a) The impulse function (δ function) is defined by the following mathematical expressions:-

$$\delta(t-t_0) = 0 \; ; \quad t <> t_0 \qquad 1.2$$

$$\int_{-\infty}^{\infty} \delta(t-t_0)\, dt = 1 \qquad 1.3$$

In practice only an approximation to the above function can be generated.

(b) Because of the limited dynamic range of real systems difficulty is experienced in testing systems at high enough energy levels to overcome the effects of noise in the measurement channel.

As a result of these practical difficulties commercial analysers offering an impulse response option often do so as a convenient "quick look" facility (5).

An alternative approach to impulse response testing which is often used to obtain the impulse response of a system is based upon the use of correlation (6,7).

Using this method the system under test is excited with a test waveform having an autocorrelation function approximating to an impulse. The output of the system is then cross-correlated with the input signal to yield the impulse response of the system.

Although white noise has an impulse as an autocorrelation function its statistical nature means that long averaging times must be used in order to achieve acceptable results (8). To avoid this, use may be made of Psudo-Random Binary Sequences (PRBS)(9,10,11,12). These signals have an autocorrelation function which approximate to an impulse and yet they are deterministic and easily generated. In addition these signals are also periodic. This means that the system response may be measured rapidly and that periodic averaging can be used to eliminate the effect of noise on the measurement.

The problem with using direct cross-correlation is that it provides the impulse response of the system under test. However many of the techniques used to design dynamic systems are based in the frequency domain and as such most system designers tend to 'think' in the frequency domain.

1.2.3 Composite Frequency Response Analysis

In order to reduce testing times a parallel approach in the frequency domain may be taken. The system in this case is subjected to a composite test signal comprising of a number of component frequencies. After a single delay, again dependant upon the settling time of the system under test, the output is captured. The spectral content of the output wave is then calculated and compared to the injected signal spectrum. In this manner a large number of spectral estimates of the system response may be obtained simultaneously. The measurement time in this case is given by

$$\text{Measurement time} = t_s + t_f \qquad 1.4$$

where

t_s = settling time of the system under test

t_f = period of the test waveform

as can be seen this provides for a much more rapid method of system testing when compared to a monotonic technique (eqn 1.1).

In order to fully exploit the advantages of composite frequency testing the test waveform should possess the following attributes.

(a) The waveform should have a rectangular magnitude envelope in the frequency domain i.e. the waveform should contain components of equal energy. This avoids weak components which may be susceptible to noise in the system under test.

(b) The waveform should have a large RMS value with respect to its peak-to-peak value ie. a low peak factor (13) as described by

$$\text{Peak Factor} = \frac{\text{Maximum Amplitude} - \text{Minimum Amplitude}}{2\sqrt{2} \times \text{RMS Value}} \qquad 1.5$$

A low peak factor allows system testing at high energy levels without requiring a large dynamic range from the system under test.

A number of composite frequency waveforms have been suggested as suitable test signals :-

Impulse and square waveforms : Although composite frequency signals these waveforms are not ideal since neither of them possess both of the attributes outlined above.

Random Noise : Since this signal is not periodic it is impossible to avoid the effects of leakage. To achieve good results it is necessary to make use of windowing techniques (30). In addition to this it is necessary to average over a large number of measurements to minimise errors. This in turn can lead to large overall measurement times.

Swept Sine : The monotonic approach outlined in Section 1.2.1 may be automated by sweeping the frequency up and/or down in one measurement period. This procedure provides a swept sine signal with a period equal to the total measurement time (14). Swept Sine signals possess good Peak Factors but the amplitude envelope is not flat and as a consequence may require averaging to produce acceptable results in the presence of noise.

Multi-frequency Binary Signals : Much attention has recently been given to the use of Multi-Frequency Binary Signals (15,16,17,18). The bit pattern associated with these binary signals is chosen in an attempt to by and large concentrate the energy of the signal in a discrete set of spectral lines. However, not only the spectral lines of interest are produced but a significant number of unwanted spectral lines are also generated. The result of this is a waveform which has a spectral amplitude envelope which is only an approximation to that desired, and an effective Peak Factor (considering only the RMS power at the measurement harmonics) which is greater than that of Psudo-Random Binary Signals.

Maximum Length Psudo-Random Binary Signals (PRBS) : It has been shown (19,20,21) that short sequence length PRBS waveforms can provide much more appropriate test signals having peak factors of 0.707 and a relatively flat envelope over a wide range of frequencies.

Since all real systems exhibit some degree of non-linear behaviour it is a further constraint that any composite test frequency waveform should be capable of producing meaningful frequency estimates when testing systems exhibiting slight non-linear effects. It is this fact which largely curtails the use of PRBS, since spectral estimates obtained by this test signal can be heavily corrupted by harmonic inter-modulation components, generated by non-linearities present in real systems.

Non-Binary Composite Test Waveforms : In order to reduce the effect of harmonics generated by non-linear behaviour a new composite test waveform has been proposed - the Prime Composite Test Signal (22).

The general composite test waveform can be expressed mathematically as

$$s(t) = \sum_{i=1}^{N} A_i \sin(w_i t + \Theta i) \qquad 1.6$$

In the case of the prime composite signal however only prime harmonically related frequencies are included, with the fundamental frequency removed so that

$$s(t) = \sum_{\substack{i<>1 \\ i<>2 \\ i=\text{ prime number}}}^{N} A_i \sin(w_i t + \Theta i) \qquad 1.7$$

The amplitudes of the sine waves are chosen to be equal and the number of sinusoids included in the signal is determined by the frequency range that is to be tested.

It has been shown (22) that this signal provides total immunity from harmonic distortion due to even power non-linearities. Also in the case of odd power non-linearities a prime sinusoid signal made up of 20 prime frequencies can be expected to give a reduction of 96% in the number of spurious harmonics that fall on the component frequencies over PRBS.

The choice of phases in the above equation is critical in that it determines the peak factor of the signal. Composing the prime composite signal with zero phase results in large peak factors. For example it has been shown that a prime composite test waveform containing 20 harmonics with zero phases results in a peak factor of 3.297.

Considerable attention has been given in the literature to optimising the peak factors of non-binary ,multi-frequency test signals (23,24,25,26). Recently a 20 harmonic prime multi frequency test signal has been designed which possesses a peak factor of 1.14 (27).

1.3 Objectives of the Investigation

It is clear from the preceding discussion that the use of composite frequency response techniques can offer significant advantage over more conventional monotonic techniques in respect of overall measurement time. Two types of composite test waveform have been identified as possessing the required attributes of a good composite test waveform.

(a) Maximum length PRBS signals

(b) Prime composite signals

It is an objective of this investigation to demonstrate the feasibility of developing a high performance frequency response analyser based upon these signals.

This thesis describes the work undertaken in order to achieve this objective. The techniques and procedures developed during the design of the instrument are laid out along with the results obtained from an evaluation of their performance.

In addition, an assessment is made of the relative effectiveness of the two waveforms incorporated into the developed instrument.

The significant contribution of this work is:

(a) A study into the attributes of the incorporated composite test waveforms.

(b) A presentation of the theoretical basis for the test procedures adopted for use in the new analyser.

(c) A description of the techniques developed for the mechanisation of the incorporated test procedures.

(d) The design and development of a new, high performance, FFT based frequency response analyser.

(e) The design of the instrument software and signal processing routines.

(f) An evaluation of the performance of the new analyser when used to determine the frequency response of a number of physical systems.

(g) An assessment of the relative performance of the incorporated composite test frequency test signals.

During the execution of the above research program much attention was given to the aliasing problems associated with the mechanisation of a composite frequency response strategy. As a result of this, new anti-aliasing compensation algorithms have been developed. These are predominantly software algorithms which significantly reduce the complexity of the developed frequency response analyser since they remove the necessity of using more elaborate anti-aliasing filters in the measurement channel.

The significant contribution of this work is:

(a) The study into the effects of aliasing in the implementation of a composite frequency response test strategy.

(b) The development of new software algorithms which compensate for the effects of aliasing on the frequency response measurements.

(c) The incorporation of these developed algorithms into the new analysers test procedures.

(d) An evaluation of the effectiveness of the new algorithms.

The following chapter lays down the theoretical basis for the test waveforms utilised during the investigation.

CHAPTER 2

THEORETICAL BASIS FOR FREQUENCY RESPONSE IDENTIFICATION TECHNIQUES

2.1 The Conventional Test Procedure

The experimental procedure adopted by conventional frequency response analysers is depicted in Figure 2.1.

A sinusoid of known amplitude and frequency (A sinwt) is applied to the system under test. After a fixed time delay, governed by the settling time for the system, the output of the system is captured and correlated simultaneously with both the input sinusoid and a quadrature signal (A coswt).

It can be shown (28) that for linear systems, the steady state system output will be a sinusoid of frequency w, amplitude $A|G(jw)|$ phase shifted by $\angle G(jw)$. The measured correlator outputs k_1 and k_2 will be given by

$$k_1 = \frac{[A]^2 \; |G(jw)|}{NT_0} \int_0^{NT_0} \sin(wt + \angle G(jw)) \sin wt \; dt \qquad 2.1$$

and

$$k_2 = \frac{[A]^2 \; |G(jw)|}{NT_0} \int_0^{NT_0} \sin(wt + \angle G(jw)) \cos wt \; dt \qquad 2.2$$

so that

$$k_1 = \frac{[A]^2 |G(jw)|}{2} \cos\angle G(jw) \qquad 2.3$$

$$k_2 = \frac{[A]^2 |G(jw)|}{2} \sin\angle G(jW) \qquad 2.4$$

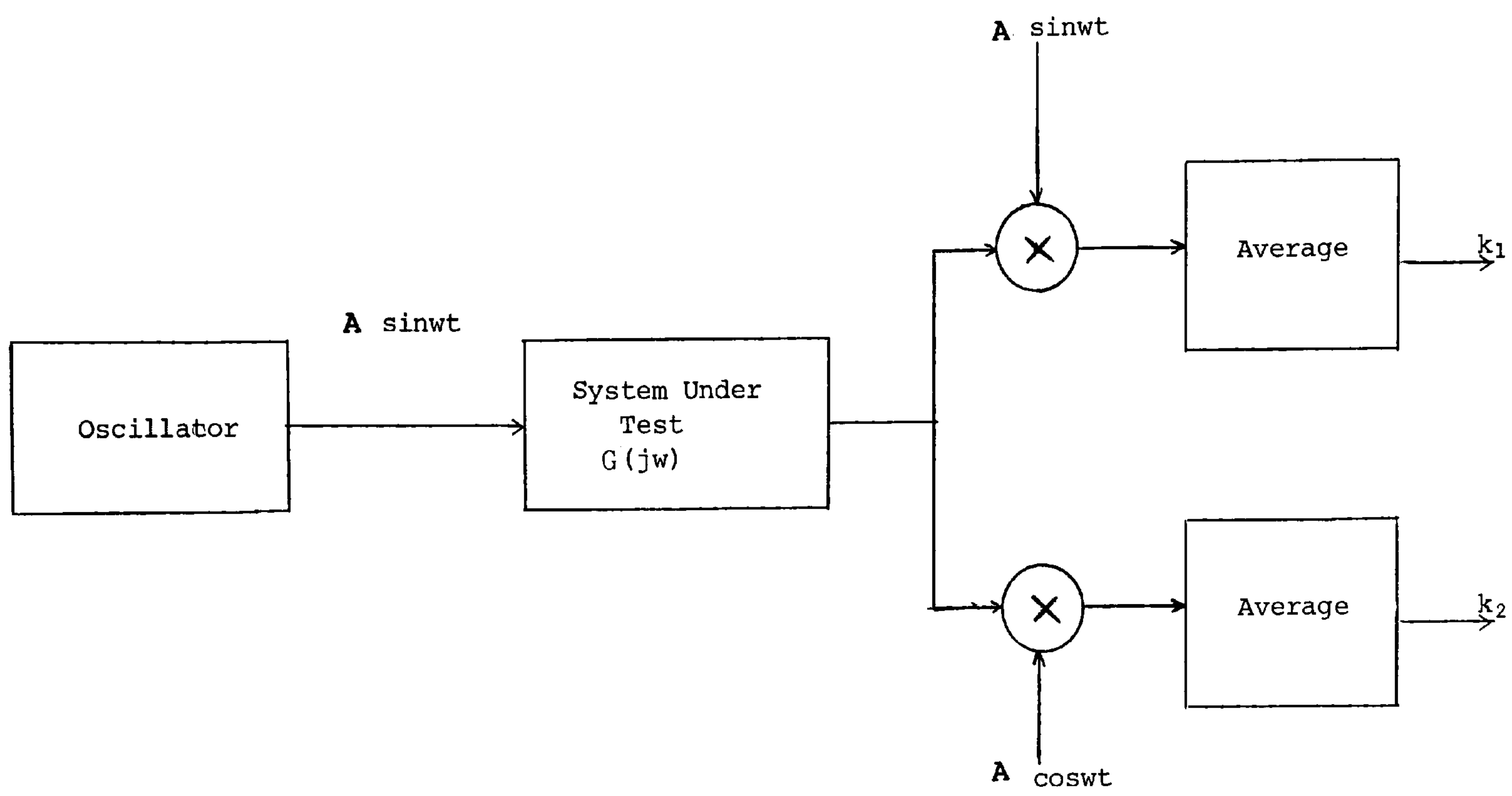


Figure 2.1 <u>**Conventional Test Procedure**</u>

From the preceding equations it can be shown that

$$k_1^2 + k_2^2 = \frac{([A]^2 G(jw))^2}{4} (\cos^2 \angle G(jw) + \sin^2 \angle G(jw)) \qquad 2.5$$

so that

$$|G(jw)| = \frac{\sqrt{4(k_1^2 + k_2^2)}}{\sqrt{A^2}} \qquad 2.6$$

and

$$k_2 / k_1 = \frac{\frac{A^2 |G(jw)|}{2} \sin \angle G(jw)}{\frac{A^2 |G(jw)|}{2} \cos \angle G(jw)} \qquad 2.7$$

i.e $\angle G(jw) = \tan^{-1} k_2 / k_1$ 2.8

Equations 2.6, 2.8 therefore demonstrate how the system transfer function can be obtained at a single frequency (w)

A major advantage of the above method for determining the transfer function of a system, is the immunity that is provided to any harmonic distortion in the output signal which may be generated as a result of non-linear system behaviour.

Furthermore, it has been demonstrated (4) that the technique can be used effectively in relatively "noisy" systems, since cross correlation of the output over multiple periods of the test waveform drastically reduces the effect of noise on measured spectral estimates.

2.2 Composite Test Procedures

Figure 2.2 shows the test procedure adopted in composite frequency based identification schemes. The system under test is excited in this case by a waveform composed of an assemblage of sinusoids of known amplitude and phase. After allowing all transitory effects to die away, the output of the system may be captured and its spectral content evaluated. Assuming the system is linear then the spectral estimates can be identified at a number of frequency points simultaneously.

Two methods may be employed in evaluating the spectral content of the output signal - multi-channel correlation (29) or the Discrete Fourier Transform (DFT) which can be efficiently computed using the Fast Fourier Transform (FFT) algorithm.

The requirement of a large number of test frequencies and rapid measurement times, constrain the viability of using a multi-channel cross-correlator for such applications. Consequently the DFT provides the most appropriate tool for use in a composite frequency based analyser.

2.2.1. The Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is a special case of the continuous Fourier Transform (30) which allows the spectral content of a signal to be calculated accurately and quickly by the digital computer.

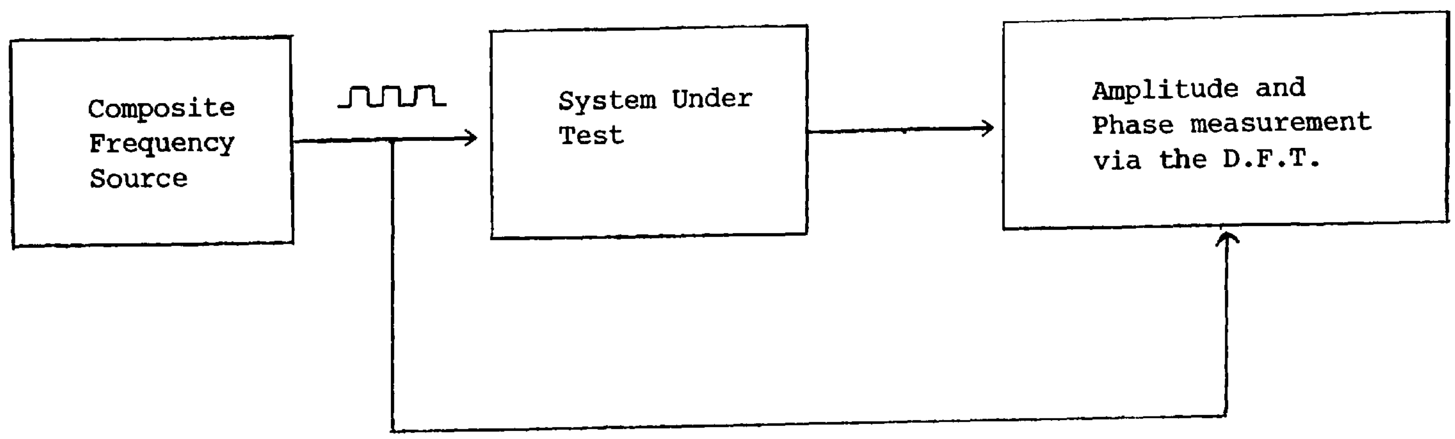


Figure 2.2 Composite Frequency Test Procedure

Figure 2.3 shows how the DFT may be used in system identification. The input and output signals of the system are sampled at a fixed period T and over a fixed time window to give two sequences $y^*(t)$ and $x^*(t)$ of length N where

$$y^*(t) = \sum_{k=0}^{N-1} y(t)\delta(t-kT) \qquad 2.9$$

$$x^*(t) = \sum_{k=0}^{N-1} x(t)\delta(t-kT) \qquad 2.10$$

It can be further shown (30) that the frequency content of these signals can be evaluated at N spectral points via the Discrete Fourier Transform and is given by

$$Y^*\left(\frac{n}{NT}\right) = \sum_{k=0}^{N-1} y(kT)\, e^{-j2\pi nk/N} \qquad 2.11$$

$$X^*\left(\frac{n}{NT}\right) = \sum_{k=0}^{N-1} x(kT)\, e^{-j2\pi nk/N} \qquad 2.12$$

where $n = 0, 1, \ldots\ldots N-1$

This result is of importance to this investigation since this sampled spectrum can be shown to be identical to within a scaling factor to the frequency spectra of the continuous signals y(t), and x(t) provided the following conditions hold :-

1. The signals must be periodic and band limited.
2. The sampling rate 1/T must be at least two times greater than the largest frequency component of y(t) and x(t).

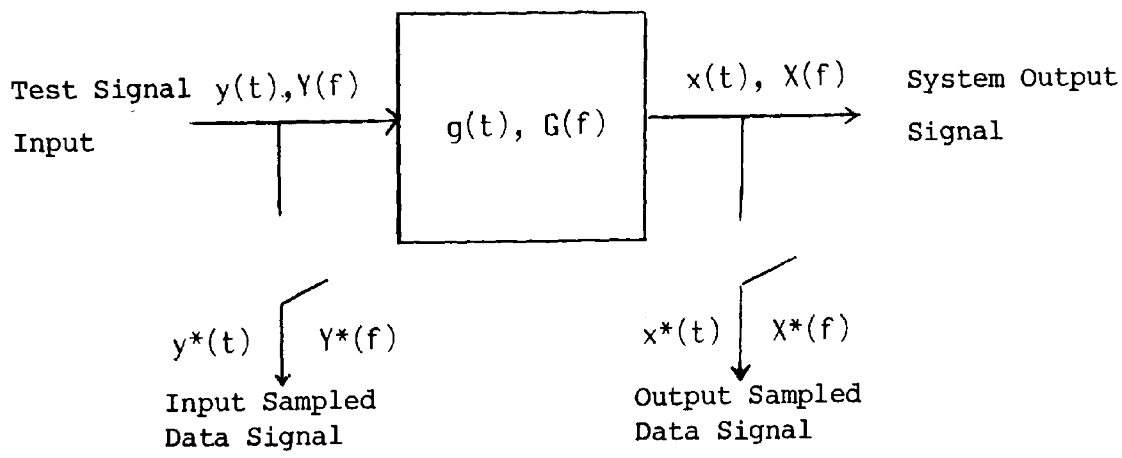


Figure 2.3 <u>System Identification using the DFT</u>

3. The time window must be an exact integer multiple of the period of U(t) and Y(t).

4. To avoid time domain aliasing the boundary of the sample window must not coincide with the sample points.

In such an identification scheme as shown in Figure 2.3 the transfer function at N points may be evaluated using the DFT and is given by

$$G^*\left(\frac{n}{NT}\right) = \frac{X^*\left(\frac{n}{NT}\right)}{Y^*\left(\frac{n}{NT}\right)} \qquad 2.13$$

Although the DFT provides a method of evaluating the spectral content of signals using a digital computer the process is still a computational intensive task involving N^2 complex multiplications and N(N-1) complex additions. If the number of frequency points at which the transfer function is to be evaluated is large then the time taken to calculate the result of a frequency response test may be prohibitive.

In order to reduce processing time use is made of the Fast Fourier Transform algorithm. Originally proposed in a paper by Cooley and Tukey (31) in 1965 this algorithm takes advantage of the properties of modulo arithmetic to reduce the number of calculations to $N\log_2 N/2$ complex multiplications and $N\log_2 N$ complex additions - a significant reduction for large N.

2.2.2 Pseudo Random Binary Sequences

The merits of using Pseudo Random Binary Sequences (PRBS) in composite frequency response analysis were outlined in Chapter 1. The time domain representation of a typical sequence is shown in Figure 2.4.

These sequences may be generated by a shift register of length n in which an exclusive OR of the m^{th} and n^{th} bits of the register is fed back to the input as shown in Figure 2.5. The shift register should never contain zero so that the maximum number of distinct binary numbers of length n which can be generated is 2^n-1. A given feedback configuration may lead to fewer than n distinct numbers. In order to obtain a maximum period PRBS m and n are chosen so that $1 + x^m + x^n$ is an irreducible polynomial. Fortunately these polynomials are well documented (32).

The power density spectrum Φ_{xx} of a PRBS sequence is illustrated in Figure 2.6. It is a line spectrum with a $(\sin x/x)^2$ envelope.

$$\Phi_{xx}(w) = a^2\frac{(N+1)}{N^2} \sum_{r=1}^{N} \frac{\sin(r\pi/N)^2}{(r\pi/N)^2} \qquad 2.14$$

where

a = amplitude of signal

N = sequence length

T_c = shift register clock period

r = spectral line number

This is a good approximation to the desired rectangular

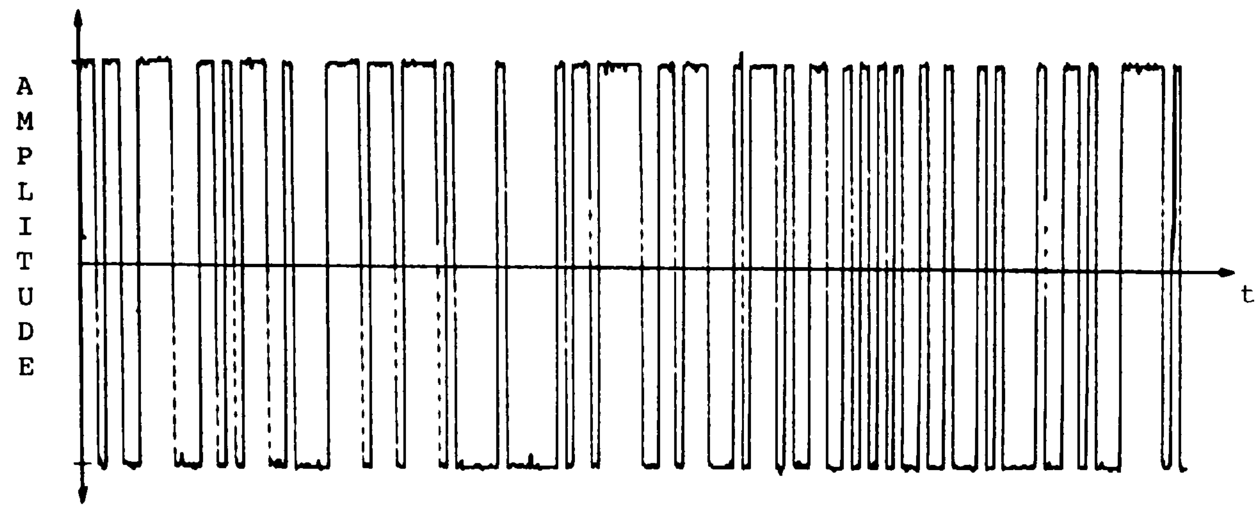


Figure 2.4 PRBS Time Domain Waveform

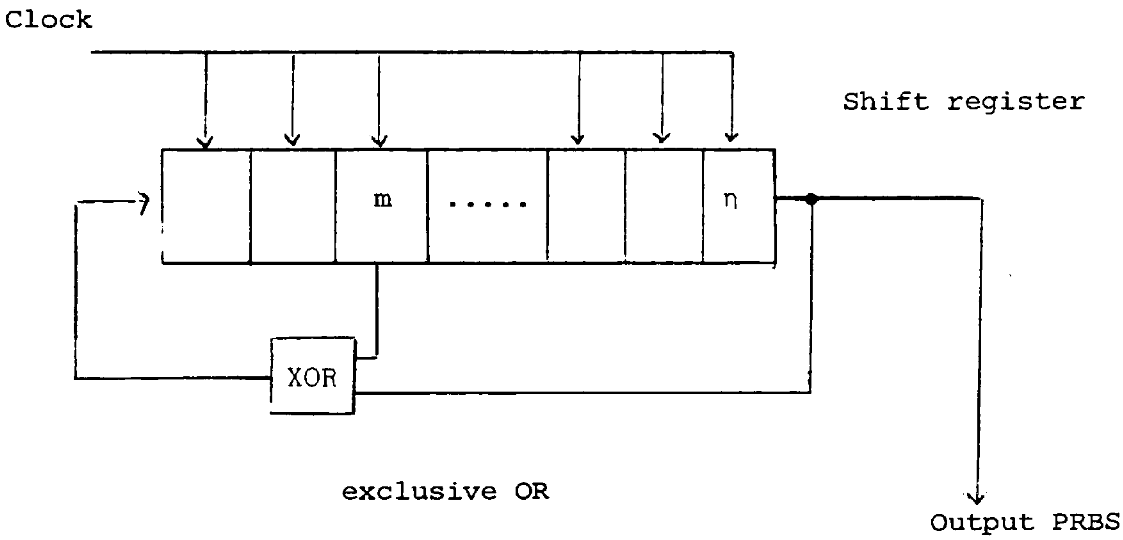


Figure 2.5 PRBS Generator

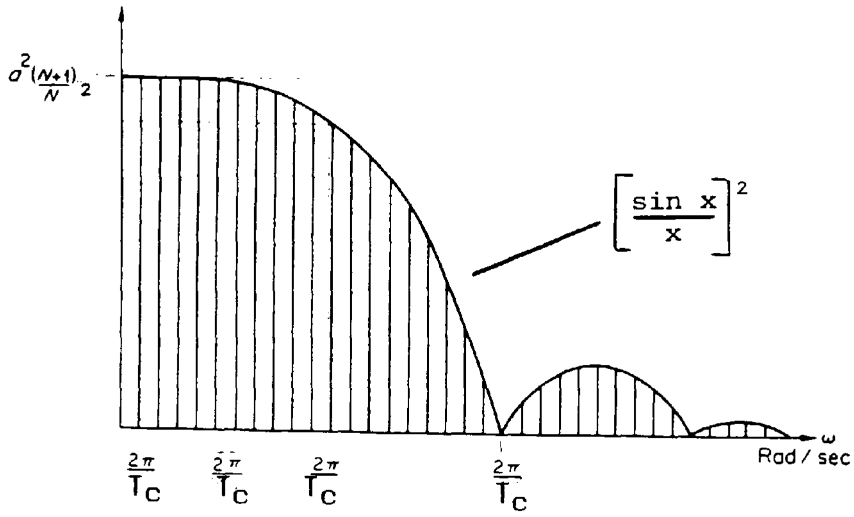


Figure 2.6

Power Density Spectrum of a maximum length PRBS signal frequency envelope, since its -3dB point occurs when

$$\frac{\sin(r\pi/N)}{(r\pi/N)} = 0.707 \qquad 2.15$$

$$\text{i.e. when} \quad r \approx N/2.5 \qquad 2.16$$

The effective frequency range covered by a maximum length PRBS signal is therefore approximately from

$$f = \frac{1}{NT} \text{ to } \frac{1}{2.25T} \text{ Hz} \qquad 2.17$$

It is clear therefore that both the period and the frequency range covered by a maximum length PRBS signal are directly dependent upon the sequence length N. However, since the PRBS test signal has spectral components at every harmonic frequency the value of N should in practice be restricted. Too large a value of N will for many systems give redundant response data. Since the peak-factor of a maximum length PRBS signal is fixed, the presence of these redundant spectral components can effectively reduce the signal to noise ratio at the measurement points.

Furthermore by adopting a convenient PRBS sequence length a complete period of the signal can be employed, thereby satisfying the third restriction on the use of the D.F.T. as put forward in Section 2.2. This in turn avoids the complexity introduced by using elaborate windowing techniques (30).

2.2.3 Prime Composite Waveform

The general equation of a Prime Composite Waveform has been presented in the time domain as

$$x(t) = \frac{1}{N} \sum_{\substack{i \neq 1 \\ i \neq 2 \\ i = \text{prime number}}}^{N} A_i \sin(w_i t + \Theta_i) \qquad 2.18$$

By putting A_i = K a constant for all i, the amplitude envelope in the frequency domain for the prime composite would approach the ideal rectangular shape as shown in Figure 2.7.

As outlined in Chapter 1 composite frequency test signals should possess low peak factor values. It can be shown that the peak factor for such a test signal is directly dependent on the phases chosen for the component sinusoids. Figure 2.8 shows the resulting time domain representation when such a signal is built out of 20 sinusoids with a zero phase relationship (Θ_i = 0 for all i). The peak factor for the signal shown is 3.297.

Much work has been carried out in order to minimise the peak factors of such signals and various algorithms have been applied such as that presented by Shroeder (13). Although these algorithms produced improvements in the peak factor figure they failed to yield an "optimum" set of phases. By making an exhaustive search of phase angle combinations based upon a quasi-Newtonian algorithm for finding the unconstrained minimum of a function $F(x_1, x_2, ----x_n)$ of the n independent variables x_1, x_2, $----x_n$ a set of optimum phases for 20 consecutive primes have been

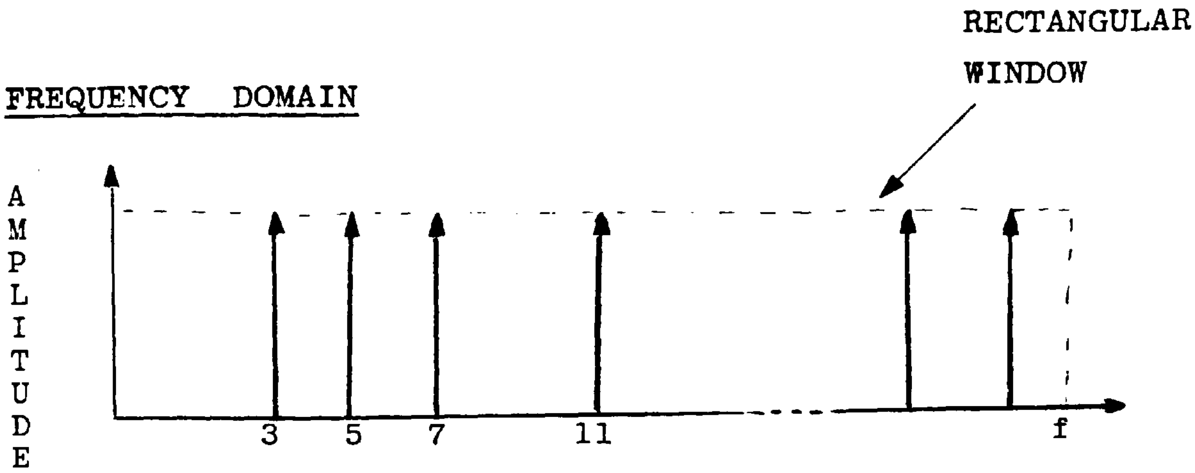


Figure 2.7 Prime Composite Waveform - Frequency Domain

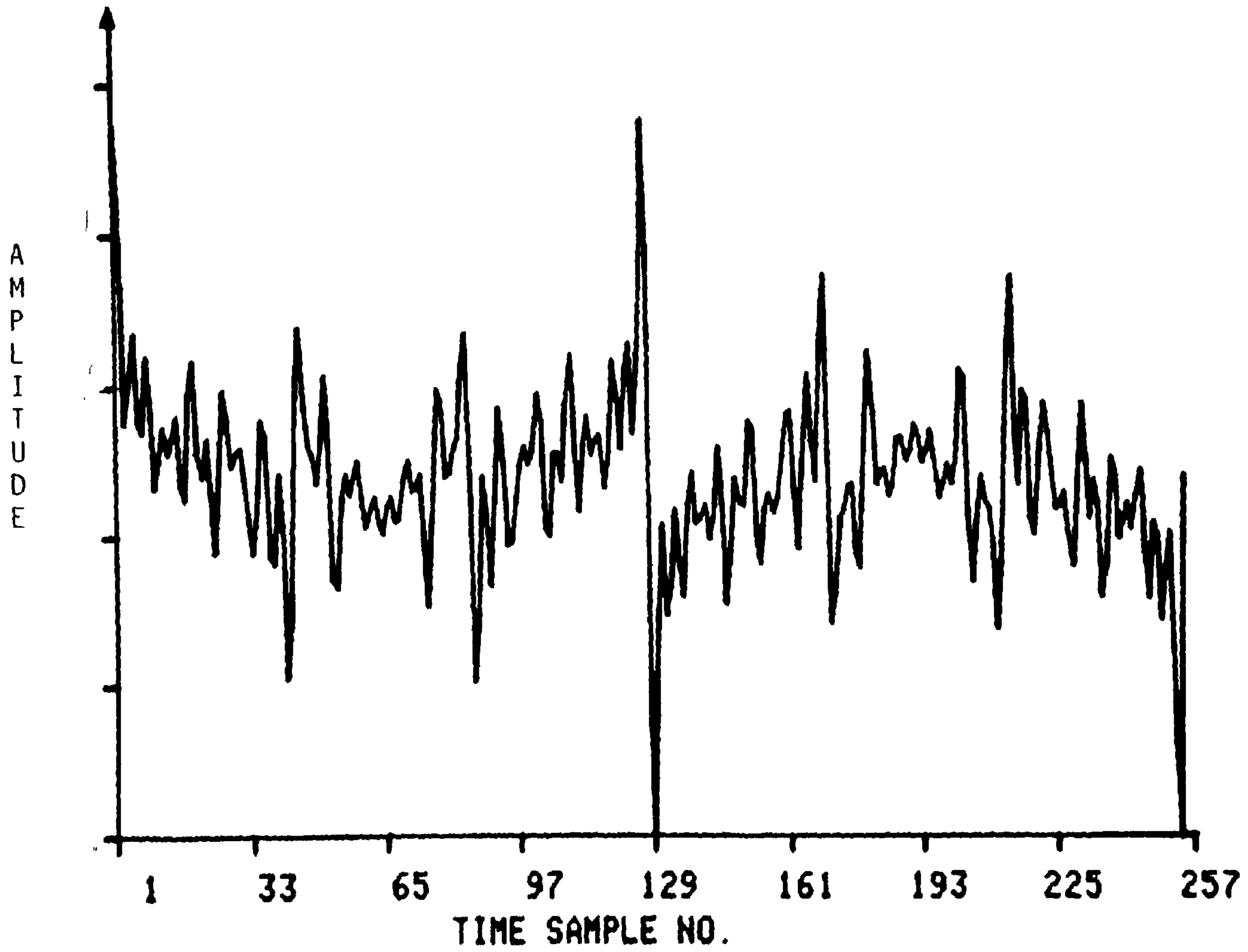


Figure 2.8 Prime Composite Test Waveform - Zero Phases

obtained. A waveform diagram of the resulting prime composite waveform is shown in Figure 2.9 having a peak-factor of 1.14 (27).

The importance of the prime composite test waveform however does not solely lie in its frequency domain envelope or in its peak factor values but in its ability to produce useful spectral estimates even when the system under test is exhibiting non-linear behaviour.

The harmonic rejection property of the prime composite waveform is hard to quantify since the amplitude distribution of the signal, the dynamics and type of non linearity exhibited by the system under test can all effect its performance. From the very nature of the waveform, however, it can be shown that harmonic distortion as a result of even type non-linear behaviour will not affect any spectral measurement taken at prime frequencies. In the case of "odd type non-linearities" however this is not necessarily the case and the possibility of corruption of readings still remains. Results of simulations on the other hand have provided encouraging results. For example it has been shown (22) that for a cubic non-linearity the number of distorting harmonics falling on the measurement frequency lines is less than 5% of that which can be expected when using a PRBS waveform with the same band coverage.

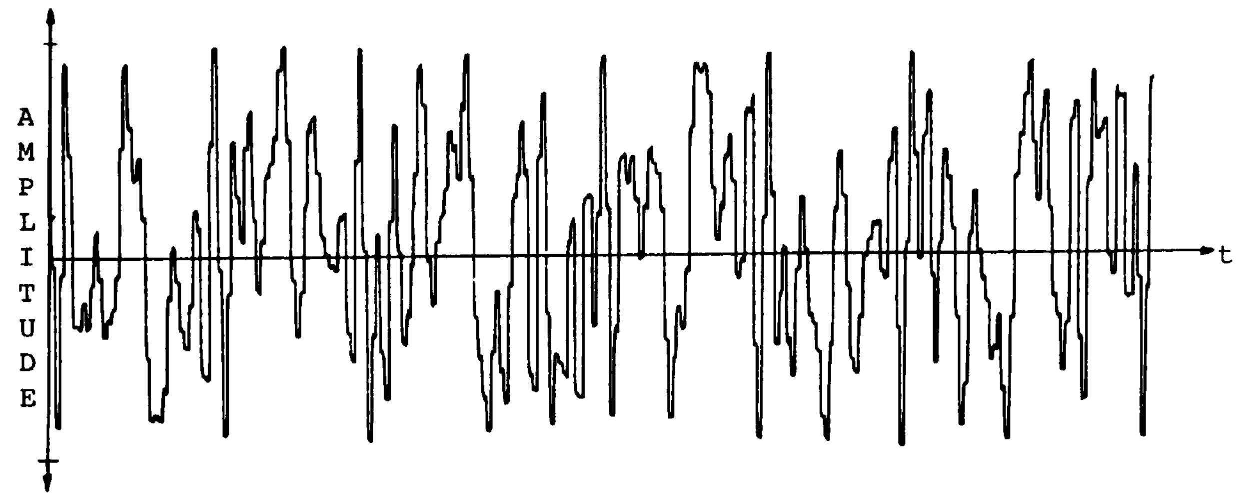


Figure 2.9 <u>Composite Test Waveform - Optimum Phases</u>

This chapter has presented the theoretical basis for the composite frequency test procedure. In addition two classes of composite test waveforms have been identified which possess the required attributes of a good composite test signal. These are Prime Composite test waveforms and short sequence length Pseudo Random Binary Signals.

The next two chapters deal with the practical problems encountered in implementing test procedures based upon these waveforms, and give an overview of a new F.F.T. based analyser which incorporates two composite frequency test waveforms.

CHAPTER 3

A FREQUENCY RESPONSE ANALYSER BASED UPON COMPOSITE TEST WAVEFORMS

The first two chapters of this thesis have described the theory upon which composite frequency response measurement is based. The inherent advantages of using composite test waveforms have been presented along with a description of two suitable waveforms, the PRBS waveform and the Prime Composite Waveform.

In order to further this area of work and to demonstrate the relative effectiveness of these waveforms for the testing of real systems, a design and development program was undertaken to produce a frequency response analyser capable of employing a variety of test strategies based upon these signals.

From the outset of the program it was decided that in order to establish the possibility of producing a viable commercial product based upon these techniques the prototype frequency response analyser should have a basic performance at least comparable if not superior to the current generation of commercially available analysers. In addition to this requirement, a further constraint was placed upon the design of the instrument. It was important that the instrument should provide a vehicle for research into frequency response measurement techniques and in particular to establish and quantify the respective performance of the test waveforms in the identification of physical systems with all the problems of non-linear behaviour and noisy environments. To this end the instrument was designed to be versatile and capable of carrying a number of test waveforms.

3.1 Instrument Specification

The basic specification for the new instrument is presented in Fig 3.1

These figures were devised so that the instrument would have a basic specification equal to or greater than the commercial, high performance, F.F.T. based analysers which were available at the outset of the investigation (1983). An overview of the equivalent performance figures of three such analysers (5,33,34) is given in Appendix G. The analysers described, however, offer many analytical features beside that of basic frequency response analysis which are not included in the specification of the analyser developed in this research program. This fact coupled with the elimination in the design of an elaborate anti-aliasing filter (discussed in chapters 4 and 9) results in an overall component cost of the prototype analyser of less than £800. The commercial analysers presented in Appendix G all have (1983) list prices in excess of £13,500. This in turn points to the the commercial viability of the new analyser in meeting a market need for a low cost, commercial, F.F.T. analyser.

The reader should be aware that during this research programme there has been major developments in F.F.T. based frequency response analysers. At the time of publication a number of new analysers (35,36,37) are available which offer higher performance at lower cost (less than £8,000).

The figures for bandwidth and resolution of the instrument dictate the spread of frequencies over which system measurement can be made and the accuracy of the amplitude measurement. These figures are consistent with the current generation of modern, high performance analysers and allow the analyser to be readily applied to a wide range of systems including process plant, servo and audio frequency systems with a high degree of resolution (72dB).

In order that the instrument may be directly interfaced to a variety of systems without the necessity of designing custom interface circuitry, the input/output systems should provide a high degree of versatility. An eight level programmable input sensitivity is provided for in the range $\pm$125mV to $\pm$8V.

An autorange facility on the waveform capture allows the 12 bit resolution of the analyser to be more fully utilised without the need of user intervention during test sequences.

The provision for including up to 4 different test waveforms has been made to give the analyser a high degree of flexibility. This can be used for two purposes. It allows the user to tailor his test signal to the system under test. In doing so a trade-off between speed of measurement, band coverage and effectiveness in the presence of non-linear behaviour can be made.

<u>Basic Prototype Specification</u>

Bandwidth	DC to 50 KHz
Resolution	12 bits
Waveform Generation Circuitry	Programmable ±125 mV to ±8V Maximum current 100 mA Short Circuit Protected
Waveform Capture Circuitry	Autoranged input ±125 mV to ±8V full scale Impedance 1MΩ ,40pF shunt
Test Waveforms	A short sequence length PRBS signal, and a composite frequency waveform comprised of 20 primely related harmonics
RS232 Serial Interface	To provide an interface to a alpha numeric terminals Baud Rates supported 110,150, 300, 600, 1200, 2400, 4800, 9600, 19200.

Figure 3.1 Basic Instrument Specification

In addition to this, by allowing the easy incorporation of multiple test signals, the analyser becomes a very useful tool for further research into frequency response analysis techniques. - The one instrument providing a common test environment for simultaneously evaluating the performance of different test strategies when applied to real systems.

As stated in Chapter 2, composite frequency response analysis relies upon the D. F T. to provide an efficient method of evaluating the system response. This dictates that the instrument must contain a powerful processing unit. Furthermore unnecessary delay in the presentation of results are to be avoided and thus the use of a Fast Fourier Transform (F.F.T.) algorithm was critical.

An RS232 interface is specified in order that an alpha numeric terminal can be used to provide a channel for machine/user dialogue. Through this terminal a user can exercise control of the analyser's various functions and display results of frequency response measurements. A programmable baud rate was deemed necessary in order to allow the instrument to be used with as wide a range of alpha numeric terminals as possible.

3.2 Overview of the New Instrument

An instrument adhering to the specifications outlined has been developed. Fig 3.2 shows the layout of the new instrument. The instrument is housed in a 19" rack and consists of five single Eurocard circuit boards.

(i) System Processor Module: The System Processor Module comprises of a single Eurocard high performance 8 MHz 68000 processor card which was bought "off the shelf" from Gespac (38). This card supplies the main system requirements for memory, serial I/O (RS232) as well as the main system processor.

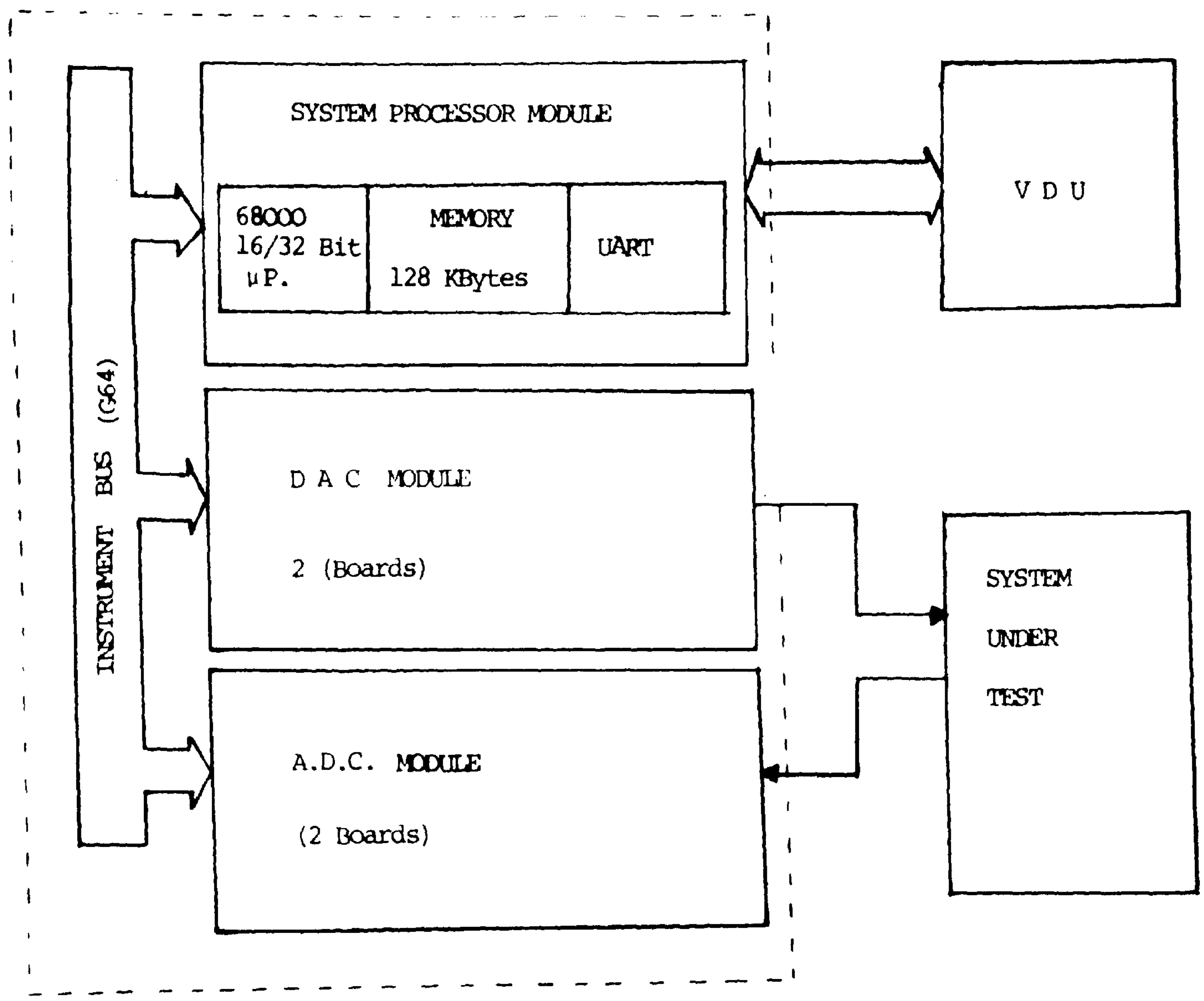


Figure 3.2 New Frequency Response Analyser

(ii) DAC Module: Consisting of two purpose designed and produced single Eurocard wire-wrapped boards this module contains the test waveform memory in which the composite test waveforms are stored in sampled and digitised form. Also resident within this module is a programmable timer, a high speed Digital to Analogue Converter and a programmable amplifier.

(iii) ADC Module: As was the case with the DAC module it required two purpose designed and produced, single Eurocard circuit boards to realise the ADC function. Besides providing analogue conditioning circuitry in the form of an autoranging input amplifier, a sample and hold amplifier and high performance Analogue to Digital Converter, this module has enough semiconductor memory to store a complete time record of the captured waveform.

(iv) Backplane: The G64 Bus was used for the interconnection between the various instrument subsystems. This bus offers 16 bit Data Bus capability, a synchronous transfer, single eurocard mechanical standard and a high reliability DIN 41612 indirect connector. These features coupled with the wide availability of G64 based "off the shelf" parts including the 68000 module were responsible for its adoption for the instrument backplane.

3.3 Functional Overview

In its current configuration the instrument supports two composite frequency waveforms.

1) A prime composite frequency waveform comprised of 20 sequential primely related harmonics (22) - this waveform can provide spectral estimates ranging in frequency from the third harmonic to the seventy third harmonic of the fundamental test waveform frequency. This waveform has been optimised to provide a peak factor of 1.34 (Lower peak factors for this type of signal did not become available until 1986 and these have yet to be incorporated within the instrument). A list of the respective magnitude and phases of the constituent harmonics is given in Appendix F.

2) A Modified PRBS waveform. As explained in Chapter 2 the sequence length and frequency domain signature of a PRBS signal is determined by the length of the shift register employed in generating it. A sequence length of 255 was chosen for the PRBS test waveform incorporated in the instrument. This gives a comparable measurement band coverage to that of the prime waveform discussed above. The (sin x)/x magnitude envelope results in a 3dB attenuation in the power of the component frequencies at the 113^{th} harmonic of the fundamental test waveform frequency (the highest frequency contained within the prime waveform is the 73^{rd} harmonic of the the test waveform frequency).

The PRBS sequence was generated by an eight stage shift register with the outputs of the eighth, seventh, second and first stages fed back via a modulo-two addition. However, in order to process the captured data via a radix-2 FFT (thereby avoiding the increase in computation time associated with a mixed radix algorithm) the sequence length of the signal was increased arbitrarily by one. This was found to be an acceptable compromise since it produces only a slight effect on the frequency domain envelope of the signal (Appendix F).

Both of the waveforms detailed above can therefore be used to provide a measurement of the system response over approximately two decades, while giving a reasonably linear band coverage. This range of two decades may be increased by increasing the number of prime frequencies employed in the prime waveform and by increasing the length of the generating shift register in the case of the PRBS waveform. This would in turn reduce the average test time per spectral estimate (See Chapter 1). However, due to essentially linear band coverage of the two waveforms, it would result in too many measurements at the higher end of the band covered than is generally required. The energy dissipated by these 'redundant' test frequency components reduce the effective peak factors of the signal since for a given peak amplitude the power at the frequencies of interest is reduced.

If however two decades does not provide enough information about the system response, the instrument is designed to readily provide a programmable number of consecutive frequency response tests to provide the extra band coverage.

Two hundred and fifty-six samples of each of the waveforms are taken, and digitised to a quantisation level of 12 bits. These samples are stored in semi-conductor memory devices within the DAC module. Under control of the overall system controller (the 68000 MPU) either of these waveforms can be output via the DAC and output analogue conditioning circuitry and applied to the system under test. The readout rate of the waveform is controlled by a timing device resident in the DAC module which in turn is programmed by the user via the system controller. Thus the period of the test waveform and hence the frequency band over which the identification is made is directly under user control.

The output from the system under test is captured via the autoranging analogue conditioning input circuitry, digitised and stored in semiconductor read/write memory in the ADC module. Two hundred and fifty-six samples are taken over a complete period of the captured waveform and these are stored as 12 bit values.

In the post test period the measurement data is retrieved from the ADC module and analysed. A 256 point Cooley-Tukey FFT algorithm is used to evaluate the spectral content of the system response. By comparing this with the spectral content of the output test waveform stored in ROM in the system processor module, the frequency response of the system is evaluated. This information is then relayed to the user terminal display in tabular form.

3.4 Design Considerations

3.4.1 Quantization Noise

This is a source of error which occurs both in the initial sampling of the test waveforms and during the capture of the system output using the analogue to digital converter. The magnitude of the quantisation error is directly dependent upon the resolution of the digitiser. It can be shown (39) that the RMS value of the quantisation noise is given by:

$$E_n = \frac{Q}{2\sqrt{3}} \text{ volts} \qquad 3.1$$

Where Q is the resolution of the digitiser which for a 12 bit converter = V/4096, where V is the fullscale voltage reading of the convertor.

The RMS values for the test waveforms are in turn given by:

$$E_t = \frac{V}{2\sqrt{2}\ \text{Peak factor}} \qquad 3.2$$

The Signal/Quantisation ratio is therefore low :

77dB for the PRBS waveform

71dB for the Prime waveform.

Another source of quantisation or rounding error is that which occurs as a result of the type of binary arithmetic used for the frequency response calculation (34). In order that this may be kept to a minimum and to avoid elaborate measures to avoid overflow, floating point arithmetic has been used throughout all calculations.

3.4.2. The Capture Window

The response of the system under test consists of two components - a transient element which for stable systems decays to zero and a "steady state" element. It is this latter component of the response which is used to gather information on the transfer function of the system.

Therefore in order that the steady state information can be collected without interference by transient effects, it is necessary that the system under test should be allowed to "settle" before testing begins. To accommodate this the instrument has been designed to generate two consecutive test waveform periods during any test.

The response of the system to the first waveform is captured but used solely for the purpose of autoranging the programmable input amplifiers. It is the data captured from the response of the system to the second period which is used for subsequent spectral estimation. In this way it is reasonable to expect that all transient effects will have "died away" and will not interfere with test results, since the test waveform period will generally be several times the value of the dominant system time constant.

Two hundred and fifty-six samples over exactly one period of the test waveform are used as input to the DFT calculation. For this reason, there is no need to apply any special window function to the samples and no time domain aliasing results.

3.4.3. Isolation

Frequency response analysis on mechanical and servo applications sometimes requires that the analyser provides a floating ground. It is intended that this will be provided by running the analyser on floating power supplies with the supply ground driven by the shield of the input cables. This approach provides very effective dc common-mode rejection.

3.4.4. System Control

One of the main design decisions to be made was the approach adopted for the control of the DAC and ADC modules. The input and output of the test waveform may primarily be described as a Direct Memory Access (DMA) activity. A number of possible options were considered for control of this operation. The required sampling rate in order to achieve a maximum test frequency of 50 KHz precluded the use of the main system processor while the advantages of using a commercial DMA processor were removed by the demands of a number of non-DMA control activities (eg autorange control). It was therefore decided to build a dedicated I/O processor. In order to maintain a degree of versatility and yet retain a structured approach this I/O processor was designed using three micro-programmable sequencers. A description of the design of these sequencers is given in Chapters 5 and 6.

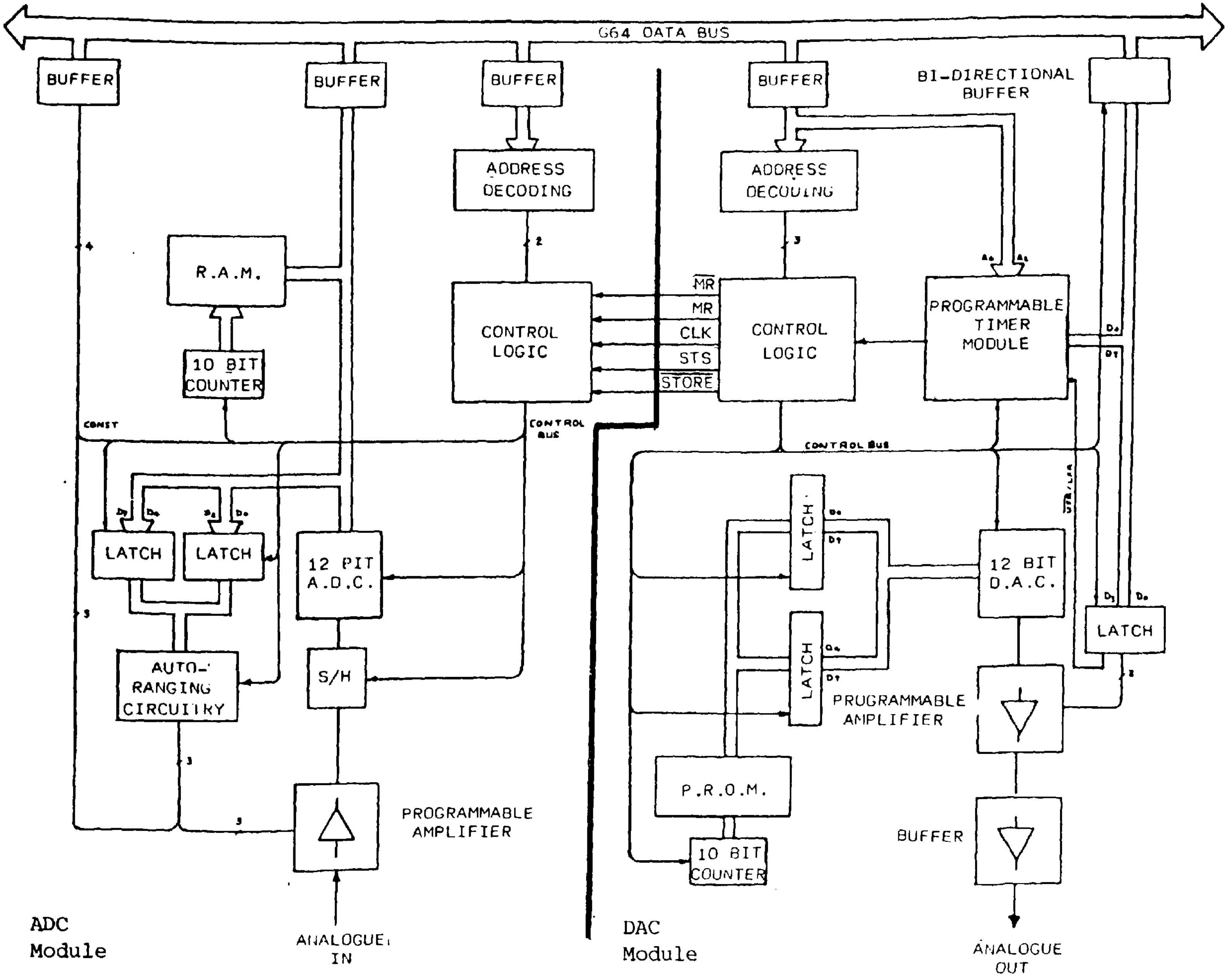


Figure 3.3 Detailed Block Diagram of the ADC and DAC Modules

3.5 A Modular Design

In order to obtain the benefits of a modular design, the DAC and ADC modules were designed to be as independent as possible. Fig 3.3 shows a functional block diagram for the two modules.

As can be clearly seen from this figure the interconnection between the two modules has been minimised to the five signal lines which are necessary to ensure that both the input and output of the test waveform are synchronised. The sample timing sequence, or STS signal, determines the sample rate of the system while the STORE signal determines whether the captured data should be stored in the RAM of the ADC module or processed by its autoranging circuitry.

The other three signals consist of the ADC/DAC module Master reset signals and the high frequency clock which provides synchronisation to the programmable sequencers. The main system processor has responsibility for initialising the programmable timer and programmable output amplifiers, as well as for the selection of the test signal and initialisation of the test sequence.

The output and capture of the test waveform is on the other hand carried out under control of the microprogrammable controllers as an independent operation.

On starting the independent operation of the analogue I/O system; a 12 bit data word is presented at the input of the DAC. This 12 bit word represents the first sample output of the test waveform. When the control circuitry observes a positive edge of the periodic STS signal, the first sample of the test waveform will be converted into its analogue representation. This procedure is repeated until two cycles of the test waveform have been output, which corresponds to 512, 12 bit samples.

At the same time as the DAC module outputs the test signal the ADC module samples the output of the system under test. The actual sampling procedure in this case is initiated by the negative edge of the STS signal. The ADC module sampling procedure continues until 512 samples are taken. Under control of the STORE signal the first 256 samples are processed by the autoranging circuitry while the second 256 samples are store in RAM. Once all 512 samples have been taken the control logic sets a conversion status flag (CONST) in the ADC module status register. The contents of this register is available to the main system processor through the G64 bus

This chapter has described the main features of the new frequency response analyser together with a functional overview of the design. The majority of the hardware design effort for the instrument lay in the in house design of the DAC and ADC modules. These designs are discussed in further detail in Chapters 5 and 6. The following chapter however presents a theoretical basis for the analyser and in particular addresses the effects of aliasing on measurements taken.

Chapter 4

ALIASING EFFECTS ON THE MEASUREMENT PROCESS

The preceding chapter of this thesis has given an overview of the basic structure and principles of operation of the new analyser. This chapter details the limitation of the practical implementation and gives the theoretical basis for the analyser design.

As stated in Chapter 3, both test waveforms are sampled, digitised and stored within ROM. During a frequency response measurement, waveform samples are read out of this memory, put through a digital to analogue convertor and applied directly to the system under test. It will be shown that this regeneration procedure can result in waveforms which differ markedly from the original. Furthermore, the reconstructed signals will not be band-limited. It can clearly be seen therefore, that if the system response does not effectively band-limit the signals then the results yielded by the DFT will be distorted by aliasing effects.

It is the aim of this section to analyse this potential aliasing problem in detail. Due to the different spectral nature of the two implemented waveforms, this is done separately for both test signals.

4.1 Prime Composite Waveform Test Strategy

As discussed in Chapter 3 the prime composite test waveform is a multifrequency signal made up of twenty primely related harmonics. The waveform may be described mathematically in the time domain by

$$s(t) = \sum_{n=3,5 \text{ prime nos}}^{73} \sin\left(2\pi \frac{nt}{NT} + \Theta n\right) \qquad 4.1$$

where N = 256, NT is the fundamental period of the test waveform and n is the harmonic number.

In the complex frequency domain this waveform is described by

$$S(f) = 1/2 \sum_{n=\pm 3,\pm 5 \text{ prime nos}}^{73} \delta\left(f-\frac{n}{NT}\right) e^{-j\Theta n} \qquad 4.2$$

When the signal is employed within the instrument to perform a frequency response analysis, the test set up is as depicted in Figure 4.1

The composite waveform is sampled, digitised and stored within the Read Only Memory (ROM) of the instrument. This is effected by multiplying waveform by a sequence of impulse functions $d_0(t)$ where

$$d_0(t) = \sum_{r=-\infty}^{\infty} \delta(t-rT) \qquad 4.3$$

and

$$D_0(f) = \frac{1}{T} \sum_{r=-\infty}^{\infty} \delta\left(f-\frac{r}{T}\right) \qquad 4.4$$

where T is the sample period.

The sampling rate was chosen so that the number of samples was 256 over the fundamental period of the waveform. This ensures that the spectrum of the ROMed samples are unaffected by aliasing (30).

From equation 4.2 it can be seen that the frequency of the highest harmonic component is given by

$$\frac{73}{NT} \text{ or } \frac{73}{256\ T} \qquad 4.5$$

The spectrum of the resulting waveform which is stored in ROM is shown in figure 4.2a and is given by

$$S^{*}(f) = S(f) * D_0(f) \qquad 4.6$$

therefore $$S^{*}(f) = \frac{1}{T} \sum_{r=-\infty}^{\infty} \delta(f-\frac{r}{T}) * S(f) \qquad 4.7$$

and $$S^{*}(f) = \frac{1}{T} \sum_{r=-\infty}^{\infty} S(f+\frac{r}{T}) \qquad 4.8$$

When performing a frequency response measurement this ROMed waveform is continuously read out of memory at a rate of one sample every T seconds and applied to the System Under Test via a zero order hold Digital to Analogue Convertor (DAC).

The time and frequency domain representation of the DAC may be given as

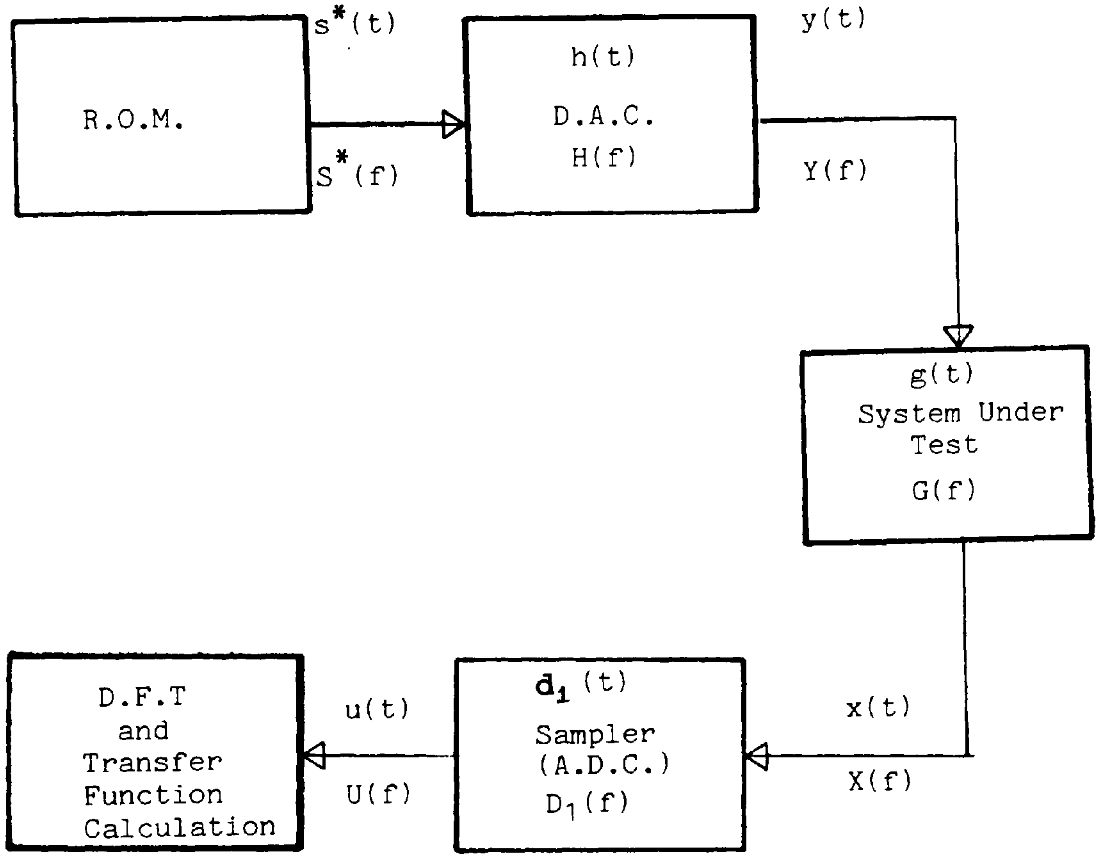


Figure 4.1 Instrument Test Procedure

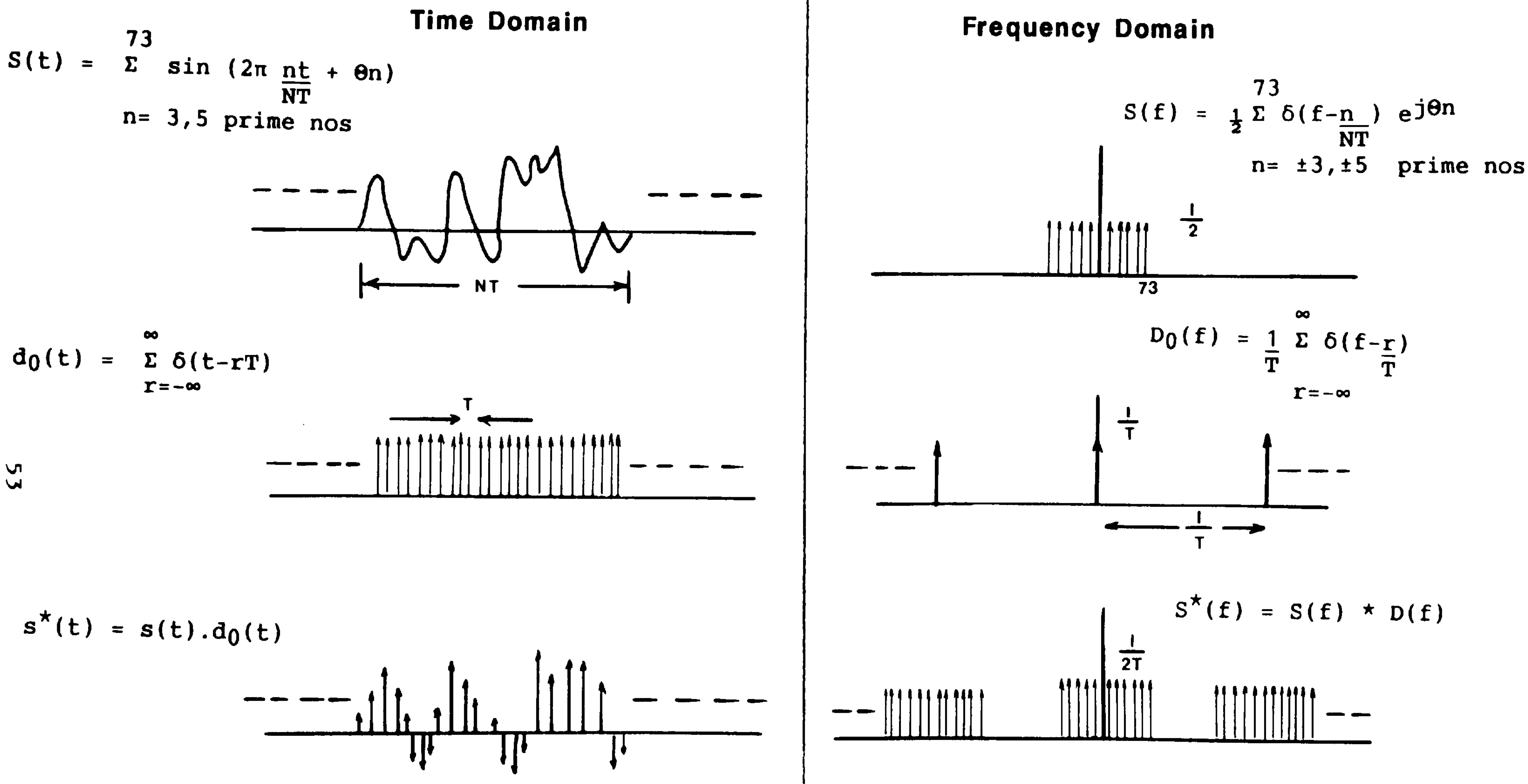
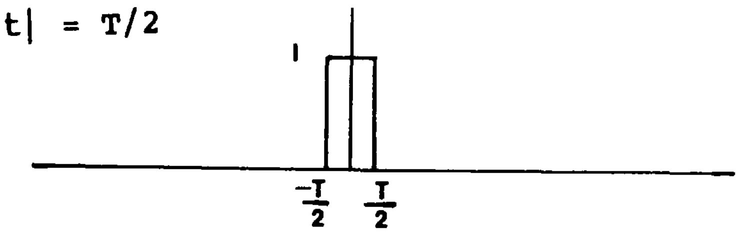


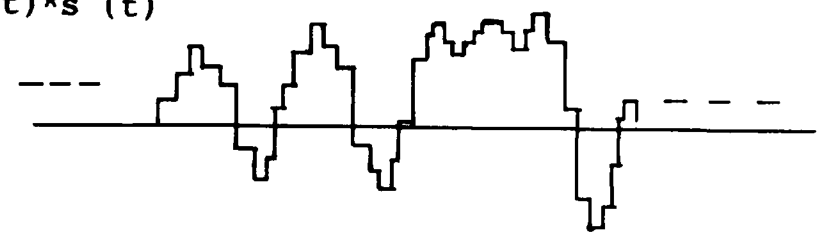
Figure 4.2a The Prime Waveform Generation Sequence

Time Domain

$$h(t) = 0 \quad |t| > T/2$$
$$= 1 \quad |t| < T/2$$
$$= \tfrac{1}{2} \quad |t| = T/2$$

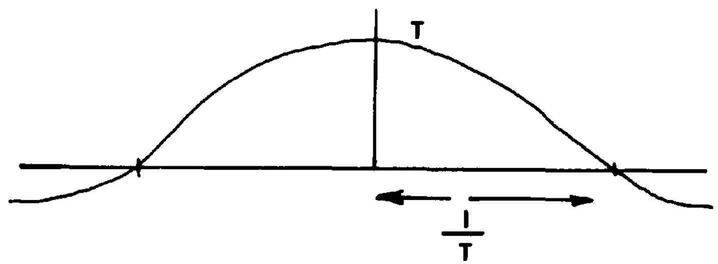


$$y(t) = h(t)*s^*(t)$$



Frequency Domain

$$H(f) = 2(T/2)\,\frac{\sin(2\pi(T/2)f)}{2\pi(T/2)f}$$



$$Y(f) = S^*(f).H(f)$$

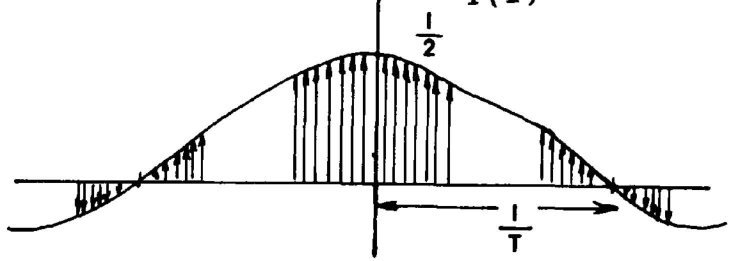


Figure 4.2b The Prime Waveform Generation Sequence

$$h(t) = 0 \quad |t| > T/2 \tag{4.9}$$
$$= 1 \quad |t| < T/2$$
$$= \tfrac{1}{2} \quad |t| = T/2$$

and

$$H(f) = 2(T/2)\,\frac{\sin(2\pi(T/2)f)}{2\pi(T/2)f} \tag{4.10}$$

The reconstituted signal applied to the system under test is shown in Figure 4.2b and is given by

$$y(t) = h(t)*s^{*}(t) \tag{4.11}$$

and

$$Y(f) = S^{*}(f).H(f) \tag{4.12}$$

$$Y(f) = \frac{1}{T}\sum_{r=-\infty}^{\infty} S(f+\frac{r}{T})\ H(f) \tag{4.13}$$

From equation 4.2

$$Y(f) = \frac{1}{T}\sum_{r=-\infty}^{\infty}\ \sum_{n=\pm3,\pm5,\text{prime no}}^{73} \delta(f-(\frac{n}{NT}+\frac{r}{T}))e^{-j\Theta n}\ H(f) \tag{4.14}$$

therefore

$$Y(f) = \frac{1}{T}\sum_{r=-\infty}^{\infty}\ \sum_{n=\pm3,\pm5,\text{prime nos}}^{73} \delta(f-(\frac{rN+n}{NT}))e^{-j\Theta n}\ H(f) \tag{4.15}$$

As can be seen from the above equation the reconstituted signal varies markedly from the original prime composite signal given in equations 4.1 and 4.2. The signal is no longer band-limited and the original component harmonics have been shaped by the DAC (H(f)).

It is this reconstituted signal which is applied to the System Under Test, whose transfer function G(f), is to be measured.

The output of the system (X(f)) is therefore described by

$$x(t) = y(t) * g(t) \qquad 4.16$$

and

$$X(f) = Y(f).G(f) \qquad 4.17$$

From equation 4.13

$$X(f) = \frac{1}{T} \sum_{r=-\infty}^{\infty} S(f+\frac{r}{T}).H(f).G(f) \qquad 4.18$$

This output signal is synchronously sampled by the Analogue to Digital Convertor (ADC) at the same rate (T) at which the input signal (y(t)) to the System Under Test is generated. This sampling function is given by

$$d_1(t) = \sum_{k=-\infty}^{\infty} \delta(t-kT) \qquad 4.19$$

and

$$D_1(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta(f-\frac{k}{T}) \qquad 4.20$$

The resulting sampled waveform u(t) therefore given by

$$u(t) = x(t).d_1(t) \qquad 4.21$$

and

$$U(f) = X(f)*D_1(f) \qquad 4.22$$

From equation 4.20

$$U(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta(f-\frac{k}{T})*X(f) \qquad 4.23$$

It thus follows that

$$U(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X(f+\frac{k}{T}) \qquad 4.24$$

and from equation 4.18

$$U(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \frac{1}{T} \sum_{r=-\infty}^{\infty} S(f+\frac{r}{T}+\frac{k}{T})\ H(f+\frac{k}{T})\ G(f+\frac{k}{T}) \qquad 4.25$$

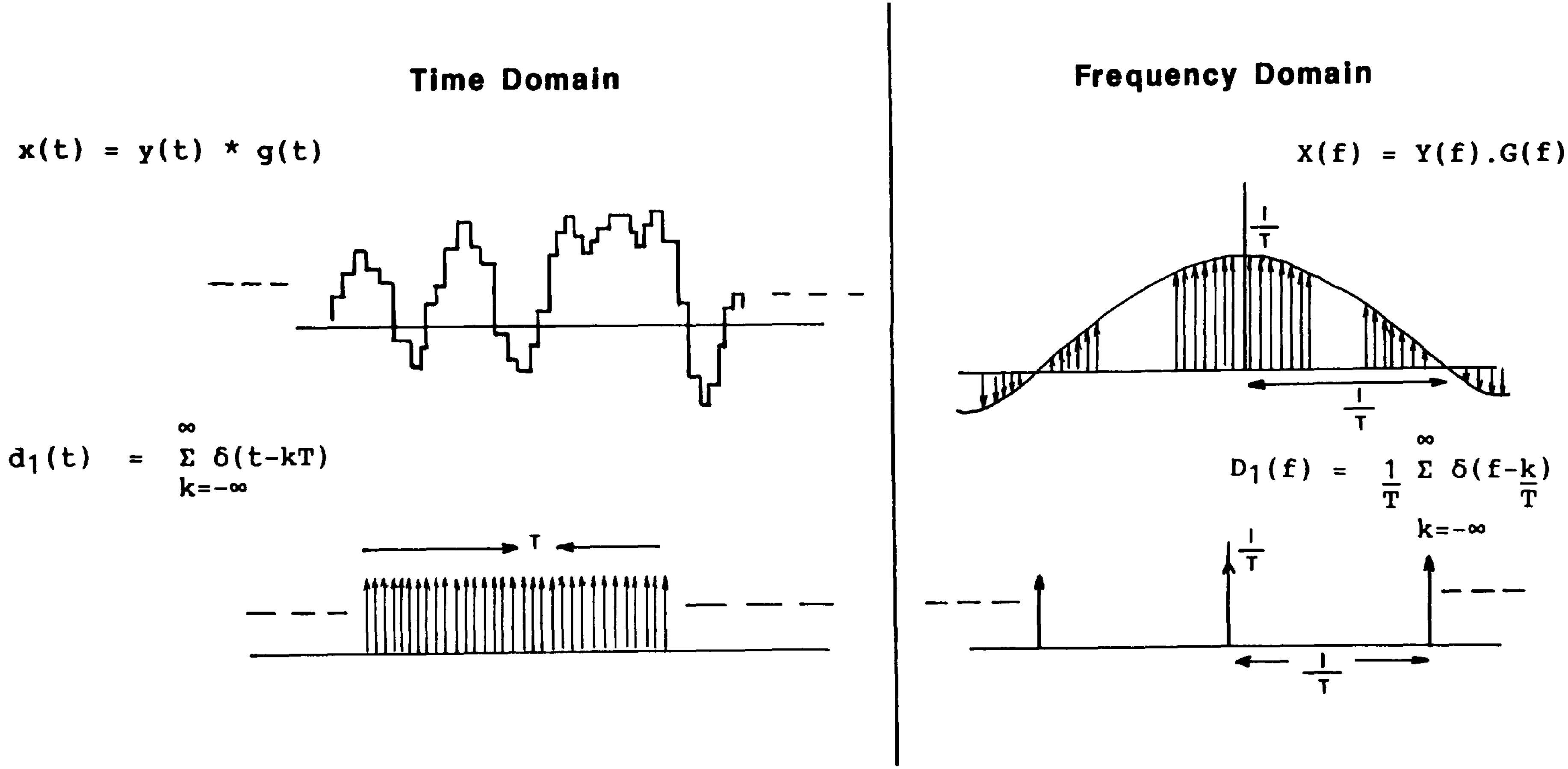
These time domain samples of the system output are transformed using a Fast Fourier Transform to yield the Discrete Fourier Transform (DFT) of the output signal of the system. This procedure is depicted in figure 4.3

By comparing the calculated spectral content of this output signal with the injected signal spectrum for n = 3, 5 ....73 the frequency response of the system can be evaluated at 20 spectral points since

$$X(f) = Y(f).G(f) \qquad 4.26$$

or $$G(f) = \frac{X(f)}{Y(f)} \qquad 4.27$$

In order that the DFT evaluates correctly at discrete frequency intervals the spectrum of x(t) it is necessary that the sampling rate is more than twice the frequency of the highest frequency component of the signal.



* Note for illustrative purposes it has been assumed that the system under test is an all pass filter

Figure 4.3a The Output Waveform Capture Sequence – Prime Composite Test Waveform

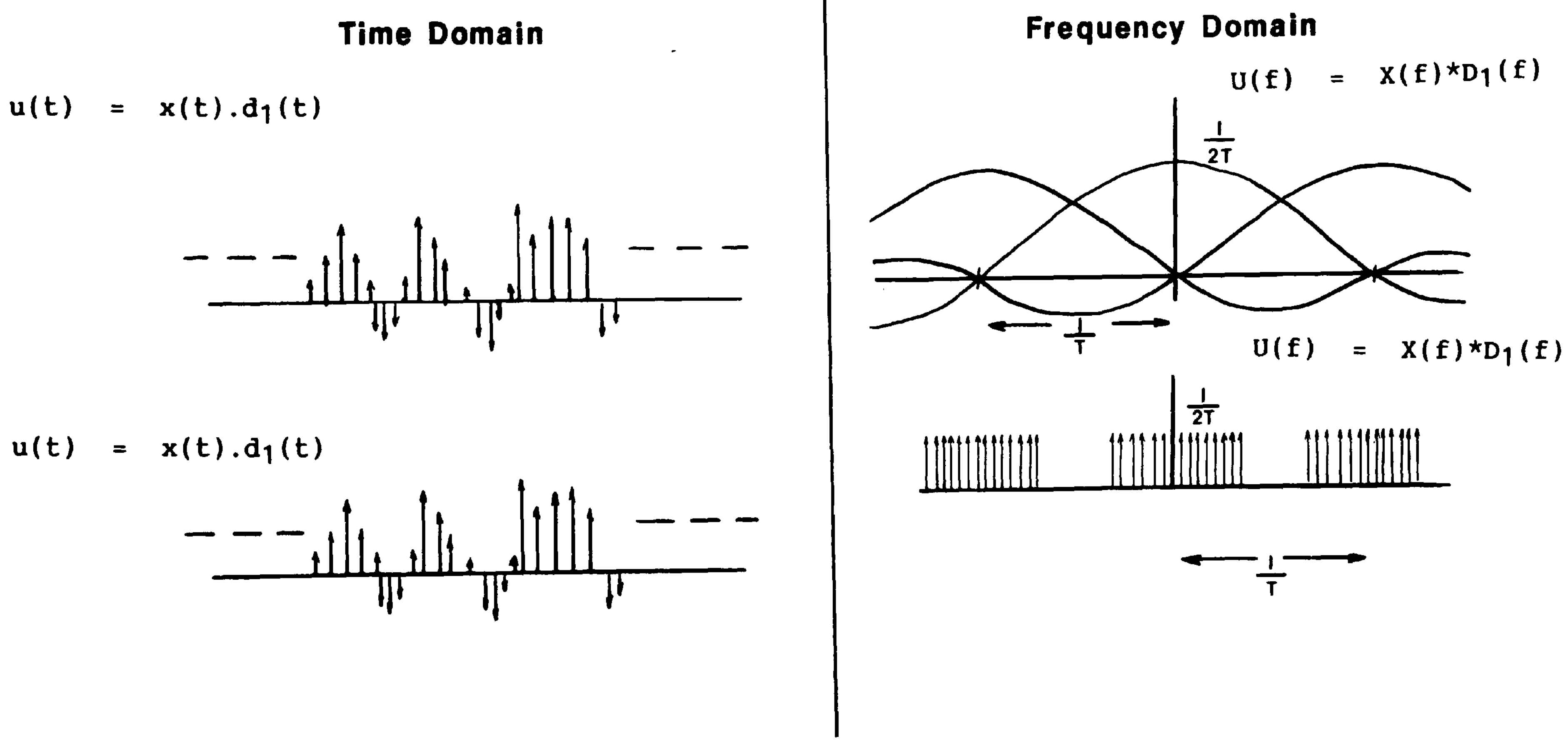


Figure 4.3b The Output Waveform Capture Sequence

– Prime Composite Test Waveform

Equation 4.15 however shows that the signal applied to the system under test is not band limited and therefore this condition cannot be guaranteed.

As a result the calculated spectrum U(f) may, be markedly different from X(f) in the range n = 3 to n = 73

4.1.1 Antialiasing Filters

For this reason it is current practice to band limit the output signal X(f) prior to sampling. This is achieved using either an analogue or digital filter or a combination of both.

The specification for the analyser under development however places stringent constraints upon the performance of such a filter. It is required that the filter has a 72dB attenuation and a cut-off frequency which may be programmed to vary from a few milliHertz to many tens of kiloHertz (See the instrument specification given in Chapter 3).

The use of analogue filters, however, are limited as they are susceptible to drift and noise and their low frequency performance is difficult to predict due to leakage effects. The anti-aliasing filter therefore, is one of the most critical parts of an analyser design (40) and has to be manufactured to very tight tolerances. Typically in the passband region the filter is required to have a flat response of around ± 0.1 dB (1.15%) while the phase response is required to be known within ± 2 degrees. In some analysers (37) aliasing is addressed by inserting a different analogue filter depending on the input sampling frequency. This means that a number of accurately defined switchable filters are required which are difficult to produce and can be expensive. Furthermore, the specification details of an analyser depend on the accuracy of the anti-aliasing filter and the input amplifiers (34). Therefore, to a large extent the overall accuracy of the frequency response data is limited by the performance of the anti-aliasing filter.

Digital filters, however, do not suffer the drawbacks of drift and noise. The main problem with such implementations lies in producing filters with high cut-off frequencies.

The digital filter must sample at a frequency which is at least twice as high as the highest frequency component of the signal to be filtered in order to avoid frequency domain aliasing. If, as is the case with the current analyser, the highest frequency component to be measured lies at 50kHz, then the filter must sample at least 100kHz. This gives less than 10μs for each filter output to be calculated.

This processing requirement is, in general, impossible to meet using low cost, general purpose, microprocessing units. The result is that special purpose Digital Signal Processing hardware (33) must be employed which leads to an increase in cost.

If, as is the case with the described reconstituted test waveform, the signal contains components above 50KHz then an additional analogue filter with a fixed cut-off frequency must be employed to avoid aliasing.

To avoid the drawbacks associated with both analogue and digital filtering techniques an alternative approach to the aliasing problem is presented.

4.1.2 The Effect of Aliasing on the Response Calculation

From equation 4.25 the spectrum of the waveform analysed by the FFT algorithm is given by

$$U(f) = \frac{1}{T}\sum_{k=-\infty}^{\infty} \frac{1}{T}\sum_{r=-\infty}^{\infty} S(f+\frac{r}{T}+\frac{k}{T}).H(f+\frac{k}{T})\ G(f+\frac{k}{T}) \qquad 4.28$$

This is a discrete and periodic spectrum with a repetition interval of 1/T as shown in Figure 4.3

As can be seen from the above equation, the calculated value at any frequency f, is a sum of an infinite number terms.

However, S(f) is a discrete band limited signal which has non zero values for f = 3/NT-73/NT i.e. for frequencies m/NT where m = 3..73 (Prime).
As a consequence of this

$$S(\frac{m}{NT}+\frac{r}{T}+\frac{k}{T}) = S(\frac{m}{NT}) \qquad k = -r \qquad 4.29$$

$$= 0 \qquad k <> -r \qquad 4.30$$

Therefore from equation 4.28 the calculated spectrum at any measurement harmonic m/NT is given by

$$U\left(\frac{m}{NT}\right) = \frac{1}{T^2} \sum_{k=-\infty}^{\infty} S\left(\frac{m}{NT}\right).H\left(\frac{m}{NT}+\frac{k}{T}\right).G\left(\frac{m}{NT}+\frac{k}{T}\right) \qquad 4.31$$

therefore

$$U\left(\frac{m}{NT}\right) = \frac{1}{T^2} \sum_{k=-\infty}^{\infty} S\left(\frac{m}{NT}\right) H\left(\frac{kN+m}{NT}\right) G\left(\frac{kN+m}{NT}\right) \qquad 4.32$$

or

$$U\left(\frac{m}{NT}\right) = \frac{1}{T^2} S\left(\frac{m}{NT}\right).H\left(\frac{m}{NT}\right).G\left(\frac{m}{NT}\right)$$

$$+ \frac{1}{T^2} \sum_{\substack{k=-\infty \\ k<>0}}^{\infty} S\left(\frac{m}{NT}\right).H\left(\frac{kN+m}{NT}\right).G\left(\frac{kN+m}{NT}\right) \qquad 4.33$$

From the above equation it can be seen that the calculated spectrum of the sampled system output is affected by an infinite number of aliasing terms. That is, the calculated spectrum of the system output at any measured frequency m/NT differs from the actual spectrum due to the extra summation terms on the right-hand side of equation 4.33.

Rearranging this equation the frequency response of the system at any test harmonic frequency m/NT is given by

$$G\left(\frac{m}{NT}\right) = \frac{U\left(\frac{m}{NT}\right) - \frac{1}{T^2} \sum_{\substack{k=-\infty \\ k<>0}}^{\infty} S\left(\frac{m}{NT}\right) H\left(\frac{kN+m}{NT}\right) G\left(\frac{kN+m}{NT}\right)}{\frac{1}{T^2} S\left(\frac{m}{NT}\right) H\left(\frac{m}{NT}\right)} \qquad 4.34$$

The above equation cannot be used directly to calculate the frequency response of the system under test since the equation contains terms for the response of the system at the 'aliasing' frequencies (kN+m)/NT, and these are unknown.

One approach to this problem is to make an assumption about the response of the system at these frequencies. For example, if it is assumed that the system exhibits a constant gain and phase response over all frequencies of interest the equation 4.32 may be rewritten as

$$U\left(\frac{m}{NT}\right) = \frac{1}{T^2} S\left(\frac{m}{NT}\right) G\left(\frac{m}{NT}\right) \sum_{k=-\infty}^{\infty} H\left(\frac{kN+m}{NT}\right) \qquad 4.35$$

From equation 4.10

$$\sum_{k=-\infty}^{\infty} H\left(\frac{kN+m}{NT}\right) = \sum_{k=-\infty}^{\infty} T \frac{\sin\left(\frac{kN+m}{N}\right)\pi}{\left(\frac{kN+m}{N}\right)\pi} \qquad 4.36$$

and furthermore it can be shown that

$$\sum_{k=-\infty}^{\infty} T \frac{\sin\left(\frac{kN+m}{N}\right)\pi}{\left(\frac{kN+m}{N}\right)\pi} = T \qquad 4.37$$

Therefore equation 4.32 can be rewritten

$$U\left(\frac{m}{NT}\right) = \frac{1}{T} S\left(\frac{m}{NT}\right) G\left(\frac{m}{NT}\right) \qquad 4.38$$

or

$$G\left(\frac{m}{NT}\right) = T \frac{U\left(\frac{m}{NT}\right)}{S\left(\frac{m}{NT}\right)} \qquad 4.39$$

That is the response of the system under test at any spectral point m/NT can be evaluated by dividing the spectral estimate of the system output as calculated by the D.F.T., by the known spectral value of the test signal. It can also be shown that by sampling the test waveform by $d_1(t)$ and using the D.F.T. to evaluate the discrete spectrum of the signal, the term T in the above equation may be eliminated. This is due to the scaling effect of the D.F.T. as applied to band-limited periodic waveforms with truncation interval equal to the period of the waveform (30).

Equation 4.39 effectively predicts the effect of aliaising on the measurements taken and in effect compensates for them. The formula is based on the premise that the system under test exhibits a constant gain and phase response over the range of frequencies. However, should the system show dynamic behaviour at these frequencies the use of equation 4.39 will result in an error being introduced into the calculated system response. To quantify the introduced error a worse case situation may be considered where the actual response of a system is given by

$$G(m/NT)_{act} = G.e^{-j\Theta}$$

$$G(kN+m/NT)_{act} = 0 \qquad 4.40$$

Using expression 4.39 and 4.33 the frequency response calculated would be

$$G(m/NT)_{calc} = \frac{U(m/NT)}{S(m/NT)} = \frac{S(m/NT)H(m/NT)G.e^{-j\Theta}}{S(m/NT)}$$

$$= H(m/NT)G.e^{-j\Theta} \qquad 4.41$$

The percentage error incurred in the calculated transfer function is therefore given by

$$\text{Error} = \frac{G.e^{-j\Theta} - H(m/NT)G.e^{-j\Theta}}{G.e^{-j\Theta}} \times 100\% \qquad 4.42$$

Therefore

$$\text{Error} = (1 - H(m/NT)) \times 100\% \qquad 4.43$$

This error will be a maximum for the case where m = 73 since H(m/NT) decreases as m increases in the range 3 ....73.

That is the maximum error in any response calculation is

$$\text{Error} = (1 - H(73/NT)) \times 100\% = 14\% \qquad 4.44$$

Although this maximum error figure of 14% is substantial, this error occurs at the higher end of the test band where it is reasonable to expect the gain of most real systems under test to be "rolling off" and as such a 14% error may be approaching the lower resolution capability of the instrument. It is suggested therefore that this figure may be acceptable given the enormous advantages gained by the exclusion of an elaborate antialiasing filter.

Initially therefore the instrument was developed using equation 4.39 for transfer function calculations when using the Prime Composite Test Waveform.

The results of instrument evaluation tests are presented in Chapter 8. Chapter 9, however, readdresses the above theoretical discussion and shows how an antialiasing compensation algorithm may be developed and implemented within the new frequency response analyser.

4.2 The Modified PRBS Waveform Test Strategy

4.2.1 Spectral Content of the Modified PRBS Waveform

The MPRBS Waveform is a repetitive pulse train with a fundamental pulse period T and a sequence length of N where N equals 256. In order to derive the spectral content of this signal it is convenient to regard the MPRBS signal as a composite signal which is the sum of 128 repetitive pulse waveforms, each of duration T and period NT, shifted with respect to each other by T seconds.

The spectral content of this signal is therefore given by

$$S(f) = \frac{1}{N} \sum_{i=1}^{256} \sum_{n=-\infty}^{\infty} A_i \frac{\sin(Tf\pi)}{(Tf\pi)} \delta\left(f - \frac{n}{NT}\right) e^{-j2\pi in/N} \qquad 4.45$$

where A_i = the amplitude of the i^{th} component periodic pulse waveform, and has a binary value (1,0).

In order to develop an expression for the effects of aliasing on measurements taken using the MPRBS signal, it is necessary to consider firstly a single periodic pulse waveform and then use superposition to develop the argument for the complete MPRBS signal.

Consider the i^{th} component pulse waveform

$$S_i(f) = \frac{1}{N} \sum_{n=-\infty}^{\infty} A_i \frac{\sin(Tf\pi)}{(Tf\pi)} \delta\left(f - \frac{n}{NT}\right) e^{-j2\pi in/N} \qquad 4.46$$

When the signal is employed within the instrument to perform a frequency response analysis the test set up is as shown in Figure 4.1.

The waveform is sampled, digitised and stored within the Read Only Memory of the instrument. This is effected by multiplying the waveform by a sequence of impulse functions $d_0(t)$ where

$$d_0(t) = \sum_{r=-\infty}^{\infty} \delta(f - rT) \qquad 4.47$$

and

$$D_0(f) = \frac{1}{T} \sum_{r=-\infty}^{\infty} \delta(f - \frac{r}{T}) \qquad 4.48$$

where the sample period T is equal to the fundamental pulse duration.

This results in sub-Nyquist sampling of the test waveform since from equation 4.46, $S_i(f)$ is not a band limited signal.

The resulting spectrum of the stored samples is therefore given by

$$S_i^*(f) = S_i(f) * D_0(f) \qquad 4.49$$

$$S_i^*(f) = \frac{1}{T} \Sigma\, \delta\, (f - \frac{r}{T}) * S_i(f) \qquad 4.50$$

$$S_i^*(f) = \frac{1}{T} \sum_{r=-\infty}^{\infty} S_i(f + \frac{r}{T}) \qquad 4.51$$

From equation 4.46 the pulse waveform has a discrete spectrum and its value at any harmonic component frequency (n/NT) is given by

$$S_i\left(\frac{n}{NT}\right) = A_i/NT \quad \frac{\sin\left(T\,\frac{n}{NT}\,\pi\right)}{\left(T\,\frac{n}{NT}\,\pi\right)} \quad e^{-j2\pi in/N} \qquad 4.52$$

$$= A_i/NT \quad \frac{\sin\left(\frac{n\pi}{N}\right)}{\left(\frac{n\pi}{N}\right)} \quad e^{-j2\pi in/N} \qquad 4.53$$

Therefore the spectrum of the stored samples is given by

$$S_i^*\left(\frac{n}{NT}\right) = \frac{1}{T} \sum_{r=-\infty}^{\infty} S_i\left(\frac{n}{NT} + \frac{r}{T}\right) \qquad 4.54$$

$$= \frac{1}{NT} \sum_{r=-\infty}^{\infty} A_i \frac{\sin\left[T\left(\frac{n}{NT} + \frac{r}{T}\right)\pi\right]}{T\left(\frac{n}{NT} + \frac{r}{T}\right)\pi} \quad e^{-j2\pi in/N} \qquad 4.55$$

$$= \frac{1}{NT^2} \sum_{r=-\infty}^{\infty} A_i \frac{\sin\left[r\pi + \frac{n\pi}{N}\right]}{\left[r\pi + \frac{n\pi}{N}\right]} \quad e^{-j2\pi in/N} \qquad 4.56$$

$$= \quad A_i/NT^2 \; e^{-j2\pi\, in/N} \qquad 4.57$$

This is an important result since it shows that the discrete spectrum of the stored samples of the periodic pulse waveform is of uniform amplitude and constant phase.

When performing a frequency response measurement this ROMed waveform is continuously read out of memory at a rate of one sample every T seconds and applied to the System Under Test via a zero order hold Digital to Analogue Converter.

The time and frequency domain representation of the D.A.C. may be given by equations 4.9 and 4.10 as

$$h(f) = \begin{cases} 0 & |t| > T/2 \\ 1 & |t| < T/2 \\ \tfrac{1}{2} & |t| = T/2 \end{cases} \qquad 4.58$$

and

$$H(f) = 2T/2 \frac{\sin(2\pi(T/2)f)}{2\pi(T/2)f} \qquad 4.59$$

$$= T \frac{\sin(\pi Tf)}{\pi Tf} \qquad 4.60$$

The reconstituted signal applied to the System Under Test is shown in figure 4.4 and is given by

$$y(t) = h(t) * S_i^*(t) \qquad 4.61$$

and

$$Y(f) = S_i^*(f).H(f) \qquad 4.62$$

$$Y(f) = \frac{1}{T} \sum_{r=-\infty}^{\infty} S_i(f+\frac{r}{T}) .H(f) \qquad 4.63$$

and $$Y(\frac{n}{NT}) = S_i^*(\frac{n}{NT})\ H(\frac{n}{NT}) \qquad 4.64$$

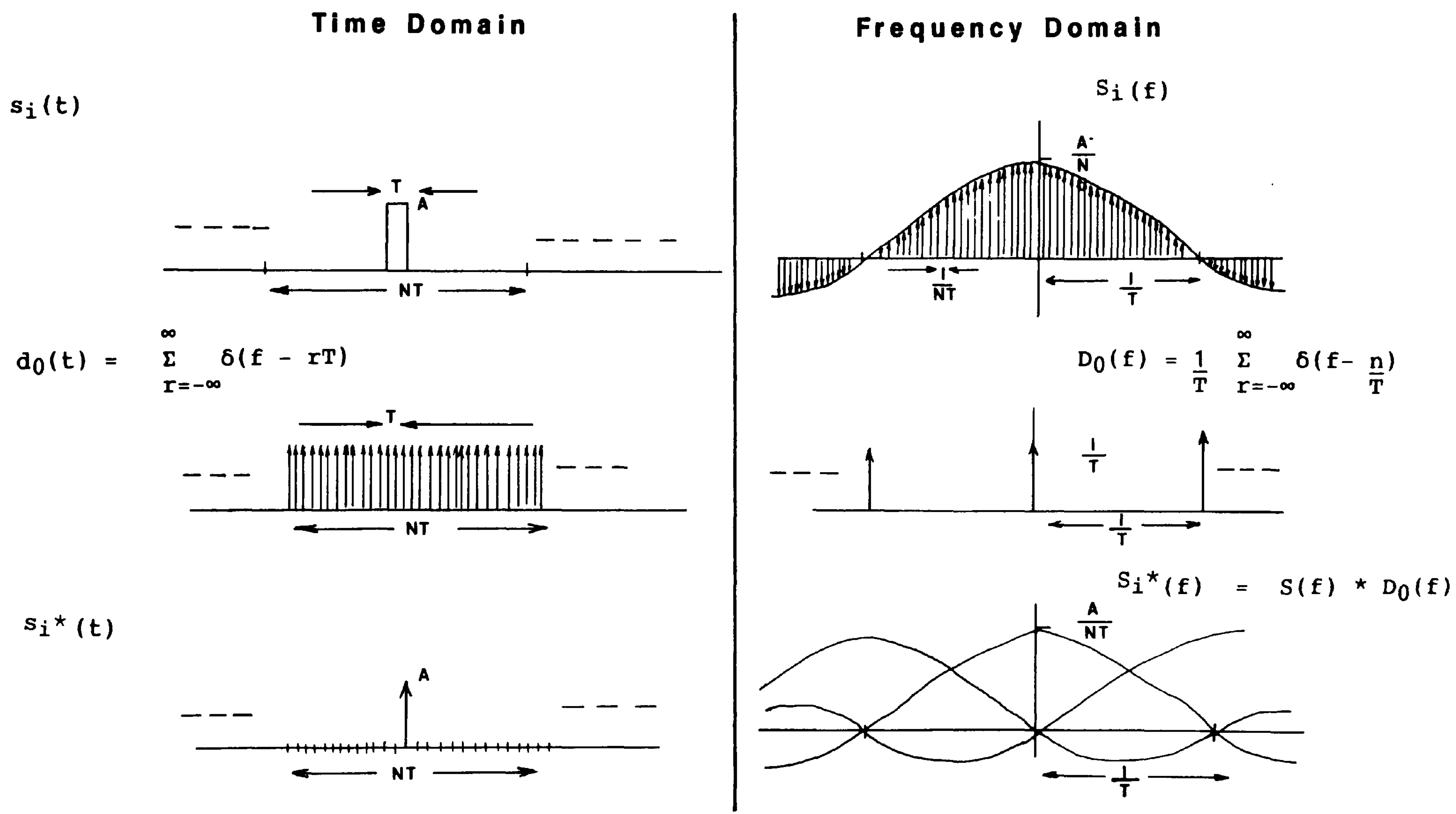


Figure 4.4a The Modified P.R.B.S Generation Sequence

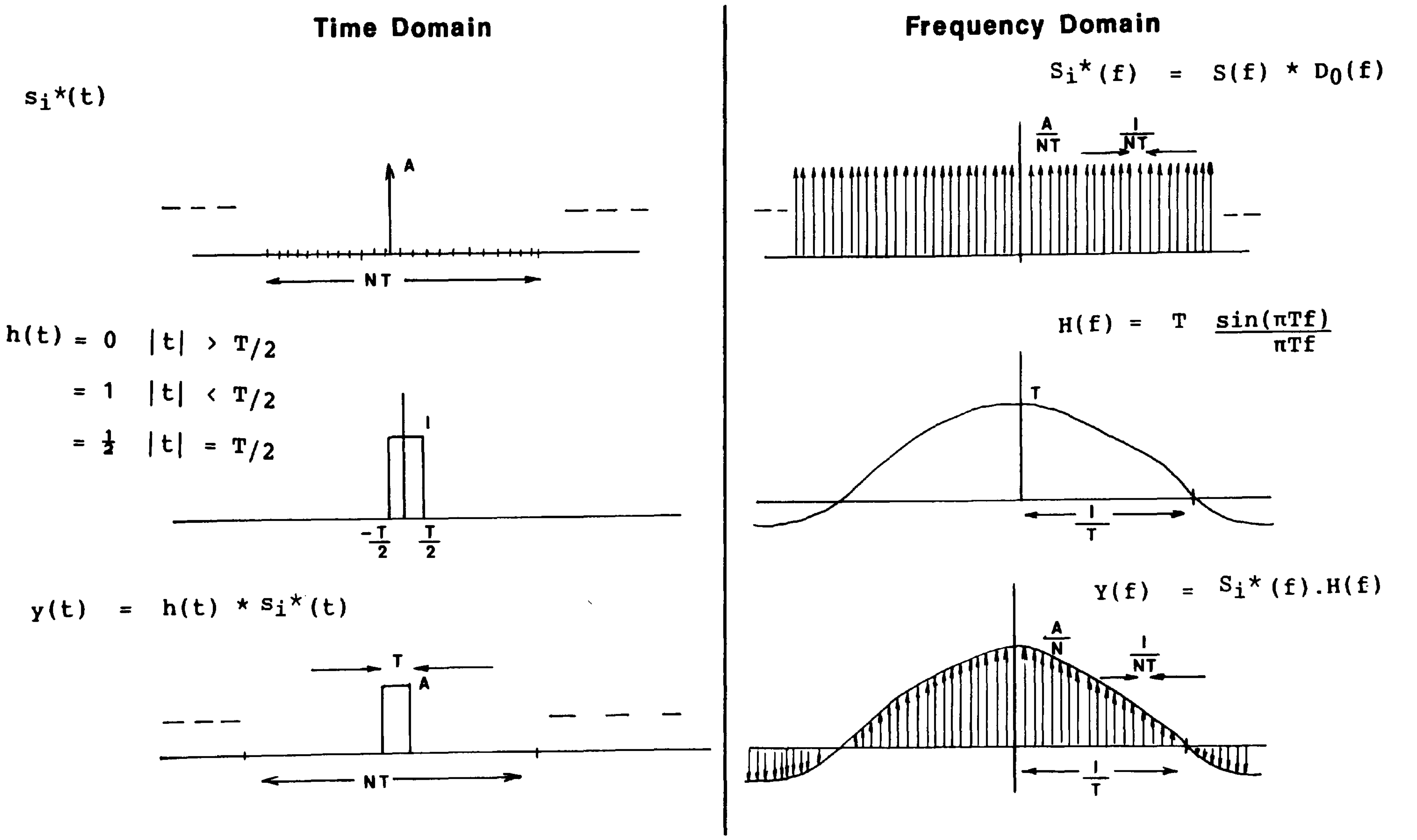


Figure 4.4b The Modified P.R.B.S Generation Sequence

Using equations 4.57 and 4.60

$$Y\left(\frac{n}{NT}\right) = A_i/NT^2 \; e^{-j2\pi \; in/N} \;\; T \; \frac{\sin(\pi Tn/NT)}{\pi Tn/NT} \qquad 4.65$$

$$Y\left(\frac{n}{NT}\right) = A_i/NT \;\; \frac{\sin(\pi n/N)}{\pi n/N} \; e{-}j2\pi in/N \qquad 4.66$$

Comparing this equation with 4.53 it can be seen that the signal applied to the System Under Test is identical to the theoretical test signal $S_i(f)$. Therefore, by superposition it can be deduced that the signal applied to the system when testing with the MPRBS waveform is identical to the original signal as given in equation 4.45.

It is this reconstituted signal which is applied to the System Under Test, whose transfer function G(f) is to be measured.

The output of the system (x(t)) is therefore described by

$$x(t) = y(t) * g(t) \qquad 4.67$$

and

$$X(f) = Y(f).G(f) = S_i(f).G(f) \qquad 4.68$$

This output signal is synchronously sampled by the Analogue to Digital Converter (A.D.C.) at the same rate (T) at which the input signal (y(f)) to the System Under Test is generated.

This sampling function is given by

$$d_1(t) = \sum_{k=-\infty}^{\infty} \delta\ (f-kT) \qquad 4.69$$

$$D_1(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta(f-\frac{k}{T}) \qquad 4.70$$

The resulting sampled waveform U(f) is therefore given by

$$u(t) = x(t).d_1(t) \qquad 4.71$$

and

$$U(f) = X(f) * D_1(f) \qquad 4.72$$

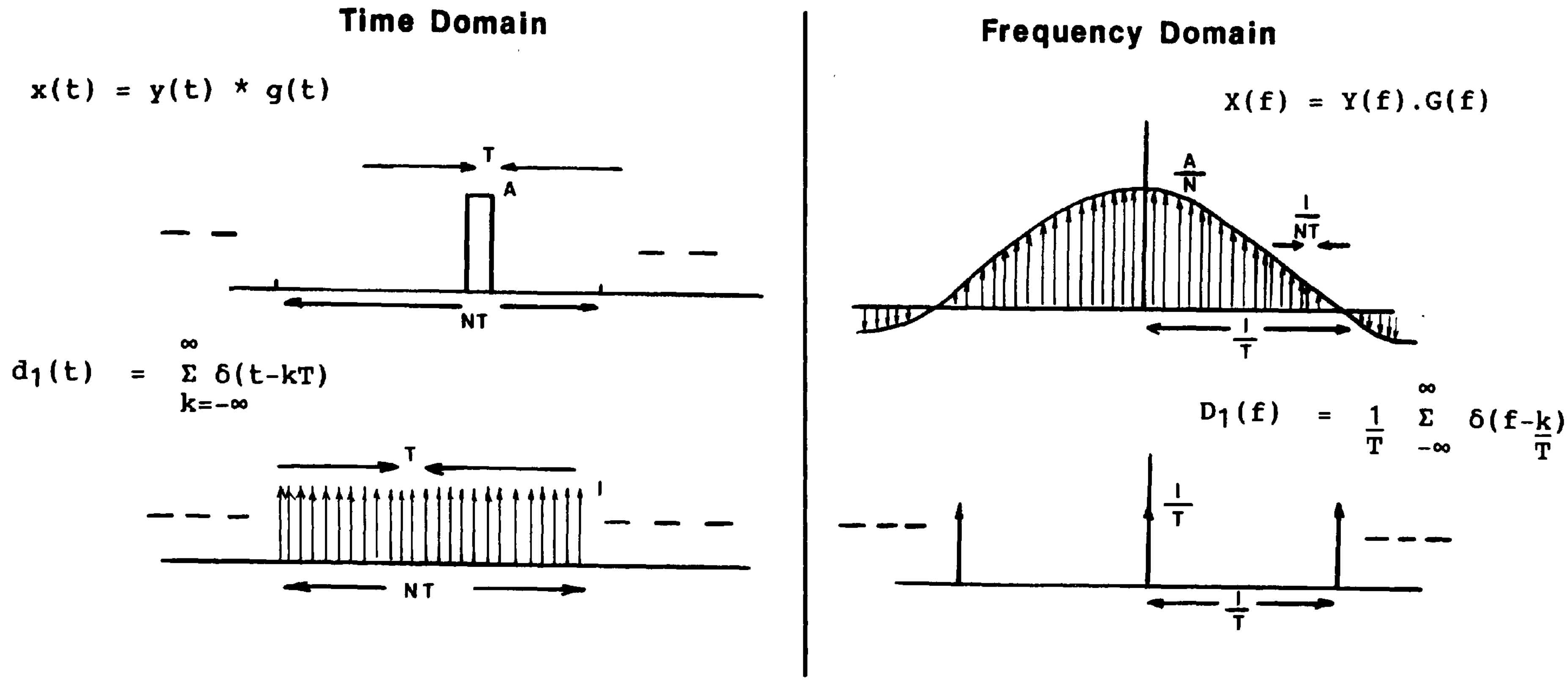
From equation 4.70

$$U(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta(f - \frac{k}{T}) * X(f) \qquad 4.73$$

$$U(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X(f + \frac{k}{T}) \qquad 4.74$$

$$U(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} S_i(f + \frac{k}{T}) \quad G(f + \frac{k}{T}) \qquad 4.75$$

The time domain samples of the system output are transformed using a Fast Fourier Transform to yield the Discrete Fourier Transform of the output signal of the system. This procedure is depicted in figure 4.5. By comparing the calculated spectral content of this output signal with the injected signal spectrum the transfer function of the system



* Note for illustrative purposes it has been assumed that the system under test is an all pass filter

Figure 4.5a The Output Waveform Capture Sequence – Modified PRBS Test Waveform

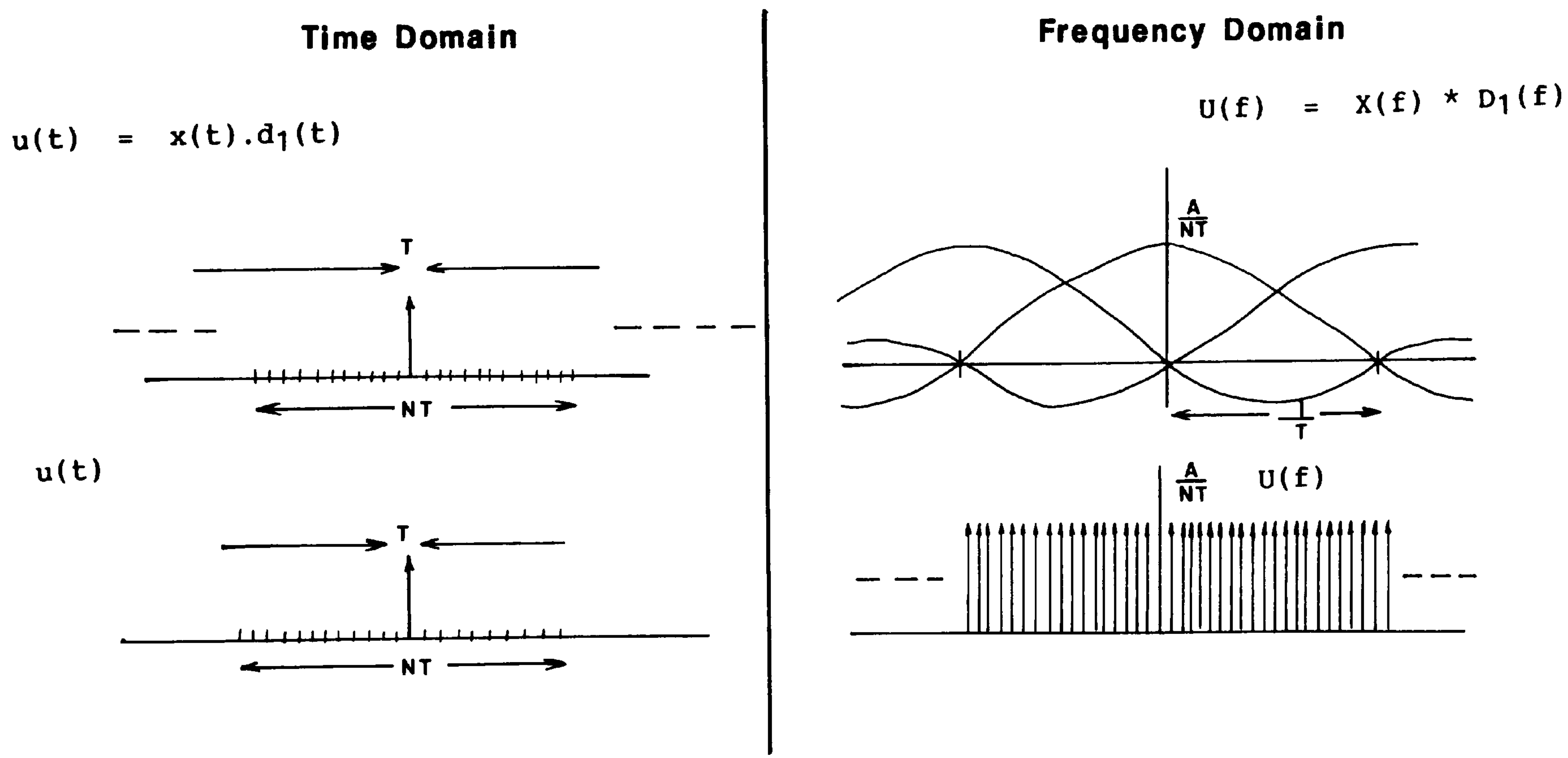


Figure 4.5b The Output Waveform Capture Sequence

- Modified P.R.B.S Test Waveform

can be evaluated since

$$X(f) = S_i(f).G(f) \qquad 4.76$$

$$G(f) = \frac{X(f)}{S_i(f)} \qquad 4.77$$

In order that the DFT evaluates correctly, at discrete frequency intervals, the spectrum of X(f), it is necessary that the sampling rate is at least twice the frequency of the highest frequency component of the signal.

Equation 4.46 however shows that the signal applied to the System Under Test is not band-limited and therefore this condition cannot be guaranteed.

As a result the calculated spectrum U(f) may be markedly different from X(f).

From 4.75 the spectrum of the waveform applied to the FFT algorithm is given by

$$U(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} S_i(f + \frac{k}{T}) \quad G(f + \frac{k}{T}) \qquad 4.78$$

This is a discrete and periodic spectrum with a repetition interval of 1/T as shown in figure 4.5.

The calculated value at the measurement harmonic m is a sum of infinite number of components and is given by $U(\frac{m}{NT})$

$$U(\frac{m}{NT}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} S_i(\frac{m}{NT} + \frac{k}{T}) \quad G(\frac{m}{NT} + \frac{k}{T}) \qquad 4.79$$

$$U\left(\frac{m}{NT}\right) = \frac{1}{T} \sum_{k=-\infty}^{\infty} S_i \left(\frac{kN+m}{NT}\right) G \left(\frac{kN+m}{NT}\right) \qquad 4.80$$

or

$$U\left(\frac{m}{NT}\right) = \frac{1}{T} S_i \left(\frac{m}{NT}\right) G \left(\frac{m}{NT}\right) \qquad 4.81$$

$$+ \frac{1}{T} \sum_{\substack{k<>0 \\ k=-\infty}}^{\infty} S_i\left(\frac{kN+m}{NT}\right) G\left(\frac{kN+m}{NT}\right) \qquad 4.82$$

From the above equation it can be seen that, as with the Prime Composite test procedure, the calculated spectrum of the sampled system output is affected by an infinite number of aliaising terms.

Rearranging equation 4.82 the transfer function of the system at any test harmonic frequency $\frac{m}{NT}$ is given by

$$G\left(\frac{m}{NT}\right) = \frac{U\left(\frac{m}{NT}\right) - \frac{1}{T} \sum_{\substack{k=-\infty \\ k<>0}}^{\infty} S_i \left(\frac{kN+m}{NT}\right) G\left(\frac{kN+m}{NT}\right)}{\frac{1}{T} S_i \left(\frac{m}{NT}\right)} \qquad 4.83$$

This expression shows that in order to obtain a highly accurate value for the system transfer function at the harmonic test frequency (m/NT) then the response of the system at the aliasing frequencies $\left(\frac{kN+m}{NT}\right)$ must be known.

Since the response of the system at these frequencies is not immediately available for an arbitrary system then equation 4.83 cannot be directly used to calculate the system response at the measurement frequency $\left(\frac{m}{NT}\right)$.

4.2.2 The Effect of Aliasing on the Response Calculation

Again one approach to this problem is to make an assumption about the response of the system at these frequencies. For example if it is assumed that the system exhibits a constant gain and phase over all frequencies of interest then equation 4.80 may be written as

$$U\left(\frac{m}{NT}\right) = \frac{1}{T} \; G\left(\frac{m}{NT}\right) \sum_{k=-\infty}^{\infty} S_i\left(\frac{kN+m}{NT}\right) \qquad 4.84$$

but from equation 4.53

$$S_i\left(\frac{n}{NT}\right) = A_i/NT \; \frac{\sin\left(\frac{n\pi}{N}\right)}{\left(\frac{n\pi}{N}\right)} \; e^{-2\pi in/N} \qquad 4.85$$

and by applying a similar argument to that given in equations 4.54 to 4.57 it can be shown that

$$\frac{1}{T} \sum_{k=-\infty}^{\infty} S_i\left(\frac{kN+m}{NT}\right) = A_i/NT \;\; e^{-j2\pi \, in/N} \qquad 4.86$$

As can be seen this is identical to the stored spectrum of the test waveform samples which is given by 4.57

Hence equation 4.81 can be rewritten as

$$U\left(\frac{m}{NT}\right) = G\left(\frac{m}{NT}\right) S_i^*\left(\frac{m}{NT}\right) \qquad 4.87$$

or

$$G\left(\frac{m}{NT}\right) = \frac{U\left(\frac{m}{NT}\right)}{S_i^*\left(\frac{m}{NT}\right)} \qquad 4.88$$

Equation 4.88 effectively predicts the effect of aliasing on the measurements taken and in effect compensates for them. The formula is based on the premise that the system under test exhibits a constant gain and phase response over the range of frequencies.

It is clear, however, that should the system under test exhibit dynamic behaviour then equation 4.88 will no longer fully compensate for the aliaising effects. In order to quantify this in a worse case situation, consider a perfect low pass filter where

$$G\left(\frac{m}{NT}\right)_{act} = Ge^{-j\Theta} \qquad 4.89$$

$$G\left(\frac{kN+m}{NT}\right)_{act} = 0 \qquad 4.90$$

From equations 4.88 and 4.82 the calculated system transfer function is given by

$$G\left(\frac{m}{NT}\right)_{calc} = \frac{U\left(\frac{m}{NT}\right)}{S_i^*\left(\frac{m}{NT}\right)} \qquad 4.91$$

$$= \frac{1/T\ S_i(m/NT)\ Ge^{-j\Theta}}{S_i^*(m/NT)} \qquad 4.92$$

But equation 4.53 gives

$$S_i\left(\frac{m}{NT}\right) = A_i/NT\ \frac{\sin(n\pi/N)}{(n\pi/N)}\ e^{-j\Theta m} \qquad 4.93$$

and equation 4.57 gives

$$S_i^*\left(\frac{m}{NT}\right) = \frac{A_i}{NT^2}e^{-j\Theta m} \qquad 4.94$$

Substituting in equation 4.92

$$G\left(\frac{m}{NT}\right)_{calc} = \frac{\frac{A_i}{NT}\ \frac{\sin(n\pi/N)e^{-j\Theta m}}{n\pi/N}\ Ge^{-j\Theta}}{\frac{A_i}{NT}\ e^{-j\Theta m}} \qquad 4.95$$

$$= Ge^{-j\Theta}\ \frac{\sin\ (m\pi/N)}{(m\pi/N)} \qquad 4.96$$

The error introduced in using equation 4.88 for the transfer function calculation would therefore be given by

$$\text{Error} = \frac{Ge^{-j\Theta} - Ge^{-j\Theta}\ \frac{\sin(m\pi/N)}{m\pi/N}}{Ge^{-j\Theta}} \quad \times 100\% \qquad 4.97$$

$$\text{Error} = 1 - \frac{\sin\ (m\pi/N)}{(m\pi/N)} \quad \times\ 100\% \qquad 4.98$$

This error increases with the measurement frequency of interest in the range m = 1...128. Since a 256 point FFT algorithm is employed within the instrument the highest measurement harmonic of interest is for m = 128. Therefore the maximum percentage error in the reading at the 128^{th} measurement harmonic is given by

$$\text{Error}_{max} = 36\% \qquad 4.99$$

It should be noted however that should the PRBS signal only be used to measure frequencies up to the 73^{rd} harmonic of the fundamental then this maximum error would reduce to less than 14%

As with the case of the error introduced in the Prime Composite measurement procedure discussed in Section 4.3

this error occurs at the higher end of the test band where it is reasonable to expect the gain of most real systems to be rolling off and an error of 14% may well approach the resolution limit of the instrument.

Initially therefore the instrument was developed using equation 4.88 for transfer function calculations when using the Modified PRBS Waveform.

The results of instrument evaluation tests are presented in Chapter 8. Chapter 9, however, readdresses the above theoretical discussion and shows how an anti-aliasing compensation algorithm may be developed and implemented within the new frequency response analyser.

CHAPTER 5

THE ADC MODULE DESIGN

This chapter describes the main design features of the ADC module. Detailed circuit diagrams of the design are documented in Appendix A. Figure 5.1 is a functional block diagram of the ADC. As can be seen from this figure, the ADC module consists of five functional units.

Input Processor - This is implemented by two micro-programmed sequencers. Its function is to control and synchronise the operation of the various parts of the ADC module.

Analogue Conditioning Circuitry - In order to utilise the full resolution of the ADC, the signal output of the system under test is buffered and amplified by this unit before it is applied to the analogue to digital converter. Since the size of the captured signal can vary in amplitude while the Full Scale Reading (FSR) of the analogue to digital converter is fixed, (+10V) an autoranging facility is incorporated into this unit, which adjusts the gain of the input amplifiers according to the size of the incoming signal.

Analogue to Digital Converter - This unit consists of a high performance 12 bit A.D.C. and its associated sample and hold amplifier. After being processed by the analogue conditioning circuitry the captured system response is sampled and digitised by this unit.

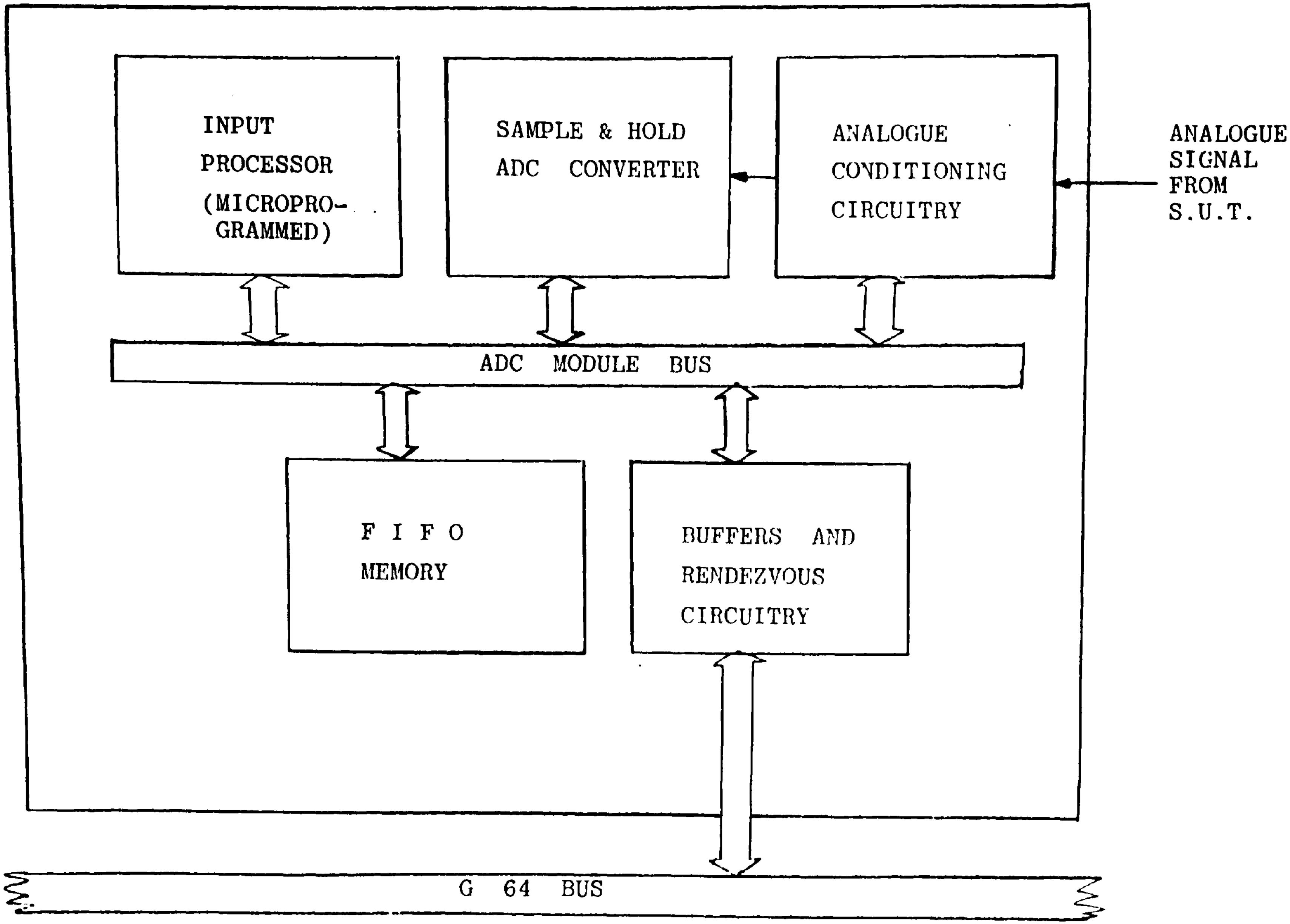


Figure 5.1 Functional Block Diagram of ADC Module

First In First Out (FIFO) Data Memory - This memory based on a 6116, 8 bit nmos RAM device is used to store the sampled response of the system under test.

Buffers and Rendezvous Circuitry - The main system processor communicates with the ADC module through this functional block. The sampled and digitised system responses, the gain adopted by the autoranging amplifiers and the status of the ADC module are made available. In addition to providing a communication channel, this unit contains all the buffer circuitry required to interface the ADC module to the main frequency response analyser bus.

The remaining sections of this chapter describe the operation and design of these units in further detail.

5.1 Autoranging Circuitry

As the system under test is being subjected to the first period of the test waveform the autoranging circuitry takes 256, 12 bit samples of the system output. By processing this sample, the autoranging subsystem determines the gain required for the input amplifiers which will ensure that the signal presented at the input to the ADC extends over its full input scale. This value, calculated during this first test waveform cycle, is used to set the gain of the input amplifiers for the capture of the system output during the second cycle of the test waveform thereby minimising the effect of the ADC quantisation noise on the calculation of the spectral estimates.

The detailed hardware design of the autoranging circuitry is given in Appendix A. Devices U3, U7, U17, U6, U10 and U14 constitute the autorange subsystem.

Fig 5.2 shows that the autorange circuit basically consists of five functional elements

(a) a dual 4 bit latch for the sample data

(b) a mapping EPROM

(c) an octal latch to latch the EPROM output

(d) a set of S-R latches to determine the maximum value output from the EPROM

(e) a priority encoder to decode the maximum output range into a suitable 3 bit gain code which is supplied to the programmable amplifier during the second test cycle.

The output from the ADC appears as a nibble representing the most significant 4 bits of the 12 bit sample followed by a byte which represents the least significant bits. During the autorange cycle of the test waveform the gain of the input amplifier is set to a maximum. Figure 5.3 shows the digital representation of the other ranges in this case. It can be seen from this figure that only the most significant 8 bits are required to evaluate the gain required from the programmable amplifiers.

With the eight most significant bits present at the inputs of the EPROM, an input voltage range is selected using a

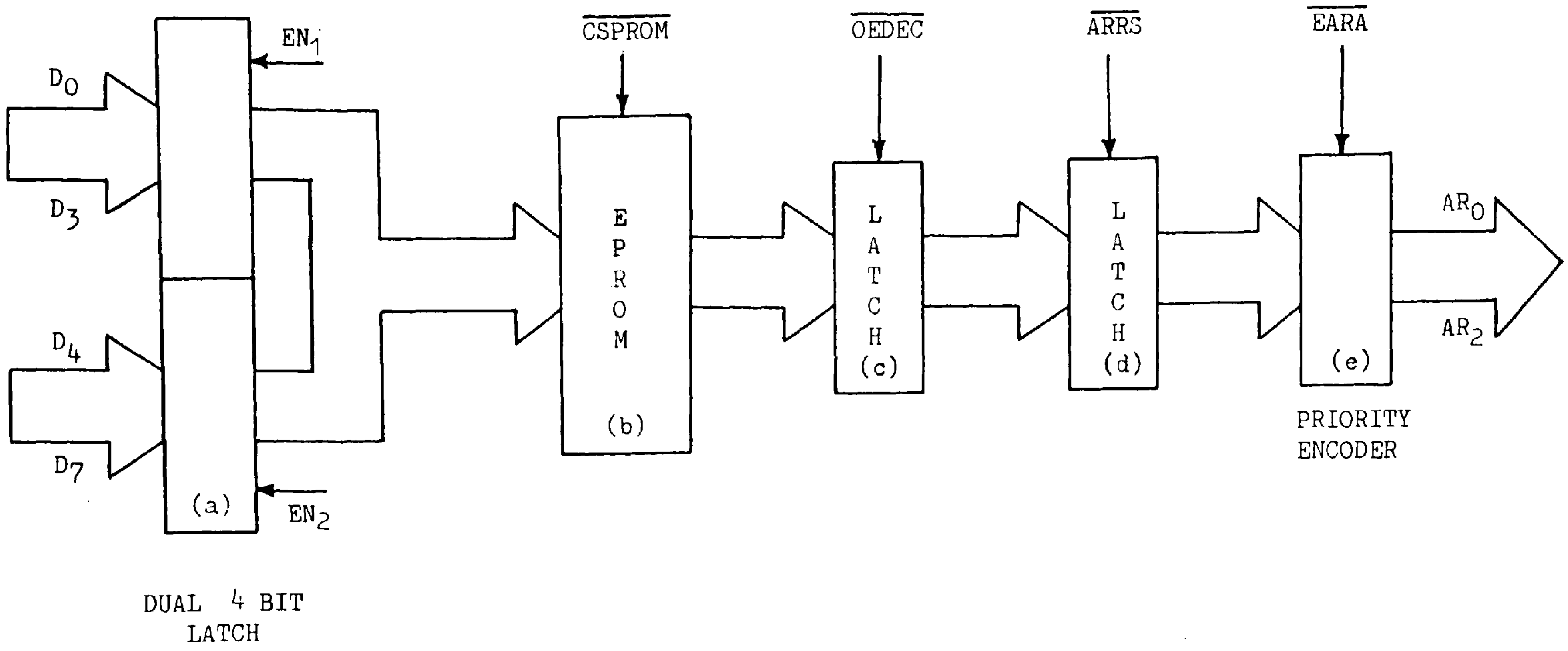


Figure 5.2 BLOCK DIAGRAM FOR AUTO-RANGING CIRCUIT

FFF E00	+ 8V
DFF D00	+ 6V
CFF C00	+ 5V
BFF A00	+ 4V
9FF 900	+ 2V
8FF 880	+ 1V
87F 820	+ 0.5V
81F 800	+ 0.125V
7FF 7E0	- 0.125V
7DF 780	- 0.5V
77F 700	- 1V
6FF 600	- 2V
5FF 400	- 4V
3FF 300	- 5V
2FF 200	- 6V
1FF 000	- 8V

Figure 5.3 RELATIONSHIP BETWEEN 12 BIT ADC OUTPUT AND INPUT VOLTAGE RANGES

suitable mapping algorithm which is encoded into the device. This process is initiated when the CSPROM signal is asserted. The signal remains valid for two clock periods. Once CSPROM has been valid for one clock period, the data at the output of the EPROM is stable. The data is then latched at the inputs of the S-R latches by the positive edge of the OEDEC pulse. Consequently, the S-R latches change state according to the EPROM's encoded gain mapping algorithm. This process is repeated 256 times, at which point, the outputs of the S-R latches represent the maximum autorange voltage gain required. Between the 256^{th} and 257^{th} sample the enable autoranging address (EARA) signal becomes valid, and this value for voltage gain is applied to the programmable input amplifiers.

5.1.1 Determination of the Maximum Input Voltage Range

The maximum input voltage range is determined by the circuit illustrated in Figure 5.5. Initially, the S-R latches are reset by the ARRS signal so that all the Q outputs are in a low state, and they only change state when the S inputs are taken low. With the circuit illustrated in Figure 5.5 and using the functional table for the priority encoder Figure 5.4, it can be seen that each voltage range would require the decoding illustrated in Figure 5.6.

Clearly, if the first sample tested fell into the $\pm$ 4 volt range, the Q outputs of the first four S-R latches would be set high, and remain high until the reset signal ARRS is taken low. Hence, if no other sample exceeds the $\pm$ 4 volt

INPUTS		OUTPUTS	
EI	0 1 2 3 4 5 6 7	A2 A1 AO	GS EO
H	X X X X X X X X	H H H	H H
L	H H H H H H H H	H H H	H L
L	X X X X X X X L	L L L	L H
L	X X X X X X L H	L L H	L H
L	X X X X X L H H	L H L	L H
L	X X X X L H H H	L H H	L H
L	X X X L H H H H	H L L	L H
L	X X L H H H H H	H L H	L H
L	X L H H H H H H	H H L	L H
L	L H H H H H H H	H H H	L H

Figure 5.4 FUNCTIONAL TABLE FOR 74LS148

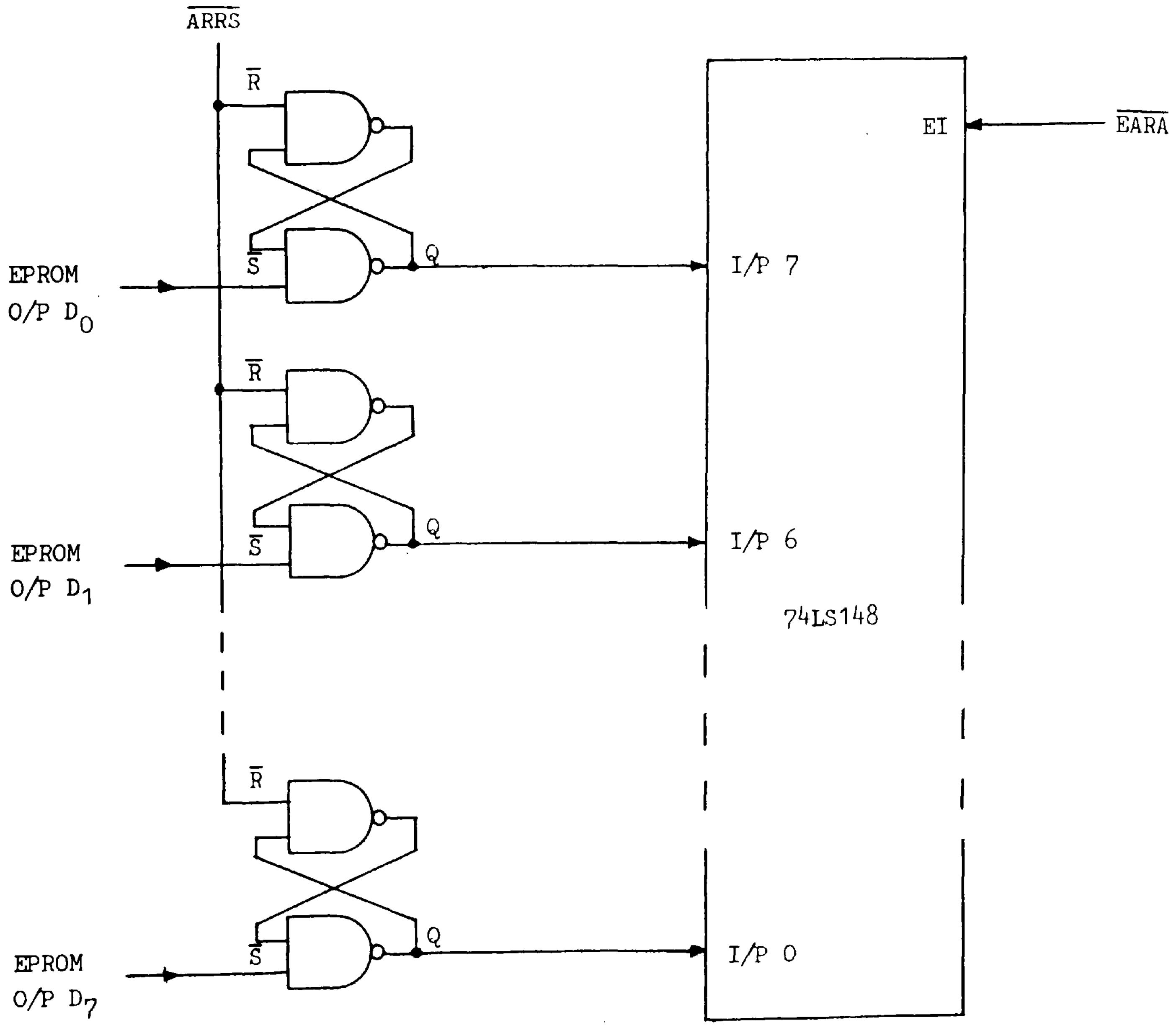


Figure 5.5 <u>CIRCUIT USED FOR SELECTING THE MAXIMUM INPUT VOLTAGE RANGE</u>

VOLTAGE RANGE	EPROM OUTPUT D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$\pm$ 8V	H	L	L	L	L	L	L	L
$\pm$ 6V	H	H	L	L	L	L	L	L
$\pm$ 5V	H	H	H	L	L	L	L	L
$\pm$ 4V	H	H	H	H	L	L	L	L
$\pm$ 2V	H	H	H	H	H	L	L	L
$\pm$ 1V	H	H	H	H	H	H	L	L
$\pm$ 0.5V	H	H	H	H	H	H	H	L
$\pm$ 0.125V	H	H	H	H	H	H	H	H

Figure 5.6 DECODED OUTPUT OF EPROM

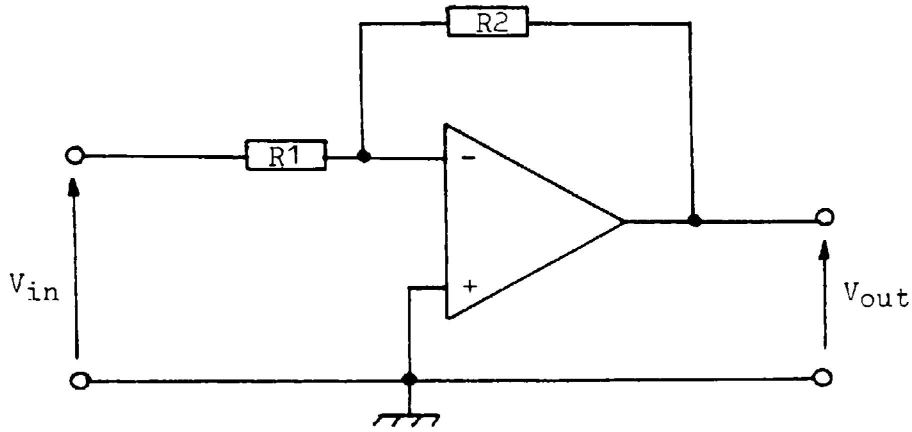
input voltage range, the other Q outputs would not change state. Thus, the maximum input voltage range selected would correspond to address 4, when the EARA signal becomes valid.

5.2 Input Amplifier

The Input Amplifier consists of two operational amplifier stages (U12, U8, U15 on ADC BOARD 1). The first stage is a straight forward inverting amplifier, as illustrated in Figure 5.7 where

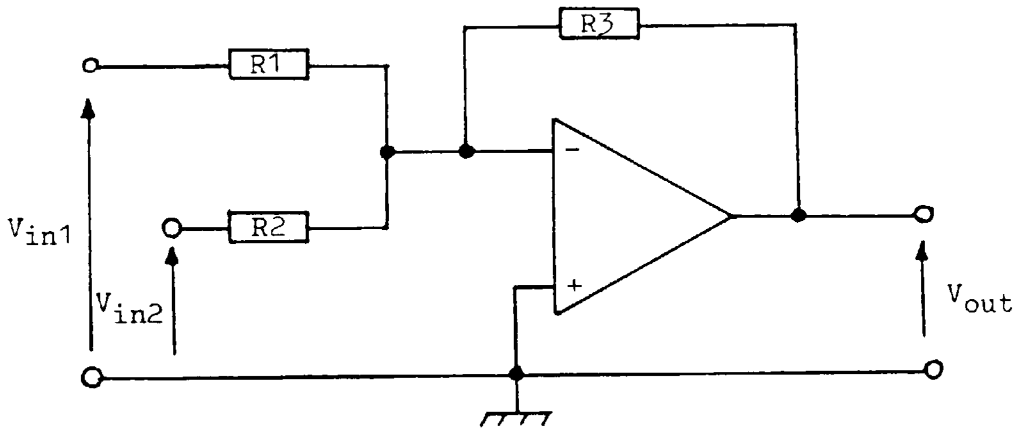
$$V_{out} = -\frac{R2}{R1} V_{in} \qquad 5.1$$

Using equation 5.2 and assuming a value for R1 and a constant output voltage range, it is clear that suitable values of R2 can be calculated to suit the various input gains required. In the design R1 is fixed at 10K and the required output voltage range is $\pm$5volts. Therefore, the value of the feedback resistor R2 can be calculated for each of the given input gains specified in the design specification. The actual values calculated are given in Figure 5.8.



(a)

Inverting Amplifier



(b)

Figure 5.7 Summing Amplifier

INPUT VOLTAGE RANGE	R2 (kΩ)	GAIN
±8V	6.250	- 0.625
±6V	8.333	- 0.833
±5V	10.000	- 1.000
±4V	12.500	- 1.250
±2V	25.000	- 2.500
±1V	50.000	- 5.000
±0.5V	100.000	- 10.000
±0.125V	400.000	- 40.000

FIGURE 5.8

Autoranging is therefore achieved by selecting the appropriate feedback resistor using an eight channel analogue switch (DG508). Each of the input voltage ranges can be selected using the 3 bit address, supplied by the autoranging circuitry. The implementation of this programmable amplifier is shown in the detailed circuit diagram of the ADC module given in Appendix A. As stated, this first amplifier stage will only convert the input voltage range into ±5volts, whereas the ADC has an input voltage range of 0 to 10 volts. Hence the need for the second summing amplifier stage, as illustrated in Figure 5.7b. Equation 5.2 gives the relationship between the input and output voltages.

$$V_{out} = -R3 \left(\frac{V_{in1}}{R1} + \frac{V_{in2}}{R2}\right) \qquad 5.2$$

Thus, using equation 5.3 with R1 = R2 = R3 = 10K , and Vin1 set to - 5 volts then an input voltage range of 5 volts will correspond to an output voltage range of 0 to 10 volts.

5.3 Analogue to Digital Converter

This section outlines the choice of the ADC used and discusses its basic operation in the context of the design given in Appendix A.

5.3.1 Choice of the Analogue Digital Converter

In order to meet the design specifications described in Chapter 3, the maximum sample rate of the ADC module was fixed at 100KHz. This in turn requires the frequency of the sequencer clock to be greater than 5MHz. This sample rate requirement proved to be difficult to meet since it dictates a conversion time of less than 10 microseconds for 12 bit ADC conversion. Most ADC modules with this conversion time are of the flash, or hybrid variety and are normally restricted to 8 bit operation as they are prohibitively expensive for longer word lengths.

The ADC eventually chosen was very high performance monolithic successive approximation converter with a typical conversion time of 6 microseconds and a full scale range of 0-10V - the AM6112 from Advanced Micro Devices.

5.3.2 Control of the Analogue to Digital Converter

The main control logic associated with the ADC is located on ADC BOARD 2, and consists mainly of a programmable sequencer (U5, U1, U3 on ADC BOARD2). Four signals are generated by the sequencer; CSADC, RDADC, WRADC and EN_1. The EN_1 signal becomes active when the most significant data byte is present at the outputs of the ADC. The other signals are used to control the ADC directly.

The timing diagram for the sequencer is illustrated in Fig 5.9 which is mainly based upon the timing requirements for stand alone mode operation for the ADC. The significance of the signals shown is detailed below.

(a) STS - This signal determines the sample rate of the system, the negative edge indicates when an input sample is to be taken.

(b) CSADC - This is the chip-select signal for the Analogue to Digital Convertor.

(c) RDADC, WRADC - These signals start the Analogue to Digital Conversion.

(d) ACK - Becomes active once the Analogue to Digital Conversion process has been completed.

(e) EN1, EN2 - Indicate whether the least or most significant byte is available.

(f) DACK This signal becomes active when the converted data

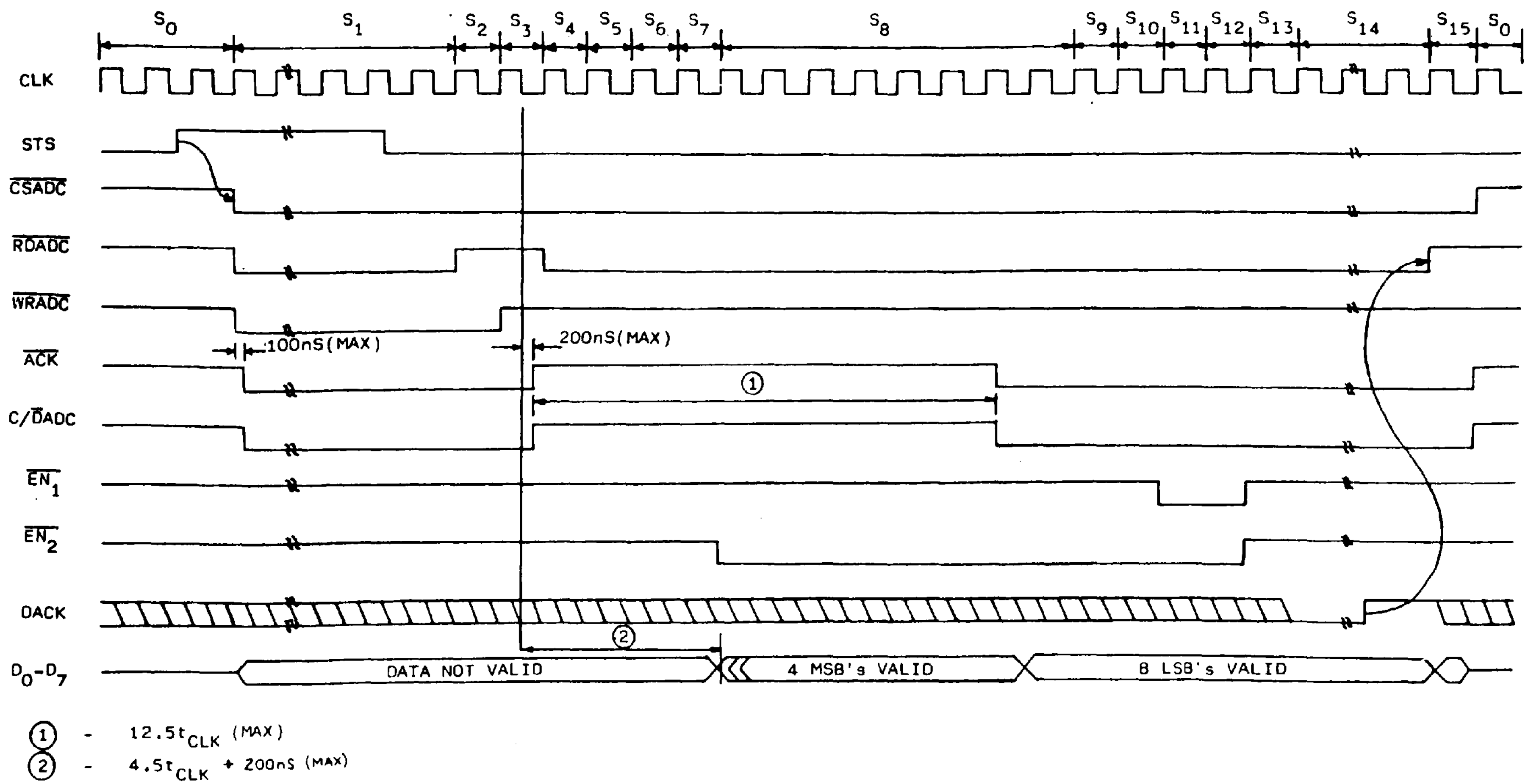


Figure 5.9 TIMING DIAGRAM FOR ADC CONTROL

has been latched by the autoranging circuit or stored in RAM.

5.3.3 Sample and Hold Circuit

When using an ADC to monitor fast changing signals, it is important that during the ADC's conversion time the analogue value remains constant. This is effectively achieved using a sample and hold device. The primary factor to consider in the device is the hold capacitor. Using the performance characteristics given in the data sheet for the LF398, it was determined that a suitable hold capacitor would have a value of 100pF. Factors considered were the acquisition time, hold settling time, aperture time, and the output droop rate.

5.4 The First In First Out Data Memory

During the second period of the test waveform the sampled system output is digitised by the ADC and placed in memory where it is subsequently made available to the main system processor. The memory device used is a 6116 RAM device. In order to simplify the addressing of the memory this device is accessed on a First In First Out basis.

5.4.1 Controlling Data Flow to RAM

As already stated the STORE signal is used to determine the destination of the sampled data. When this signal is invalid all data is transferred to the autoranging circuitry. On the other hand when the STORE signal is asserted all the data is stored in RAM.

The main control signals associated with storing the sampled data in RAM are generated by a second sequencer present on ADC BOARD 2 (U5, U2, U4 on ADC BOARD 2). The timing diagram associated with the sequencer is illustrated in Fig 5.10, the output signals of the sequencer are RS, INCA1, CSRAM1 and WERAM1. Detailed below are the functions of the signals shown.

(a) RS - Reset signal. Resets the Autoranging circuitry and the RAM address counter.

(b) INCA1 - This signal increments the RAM address counter after each byte is stored.

(c) CSRAM1, WERAM - Data is written into the RAM under the control of these waveforms.

(d) DACK1 - Indicates the completion of a two byte write cycle.

(e) CONC - Generated by the RAM adress counter; indicates that the last byte of data has been stored.

(f) STORE - Generated on the DAC board this signal indicates whether data is to be stored in RAM or directed to the autoranging circuitry.

(g) EN1,EN2 - Indicates if the least or most significant data byte is available.

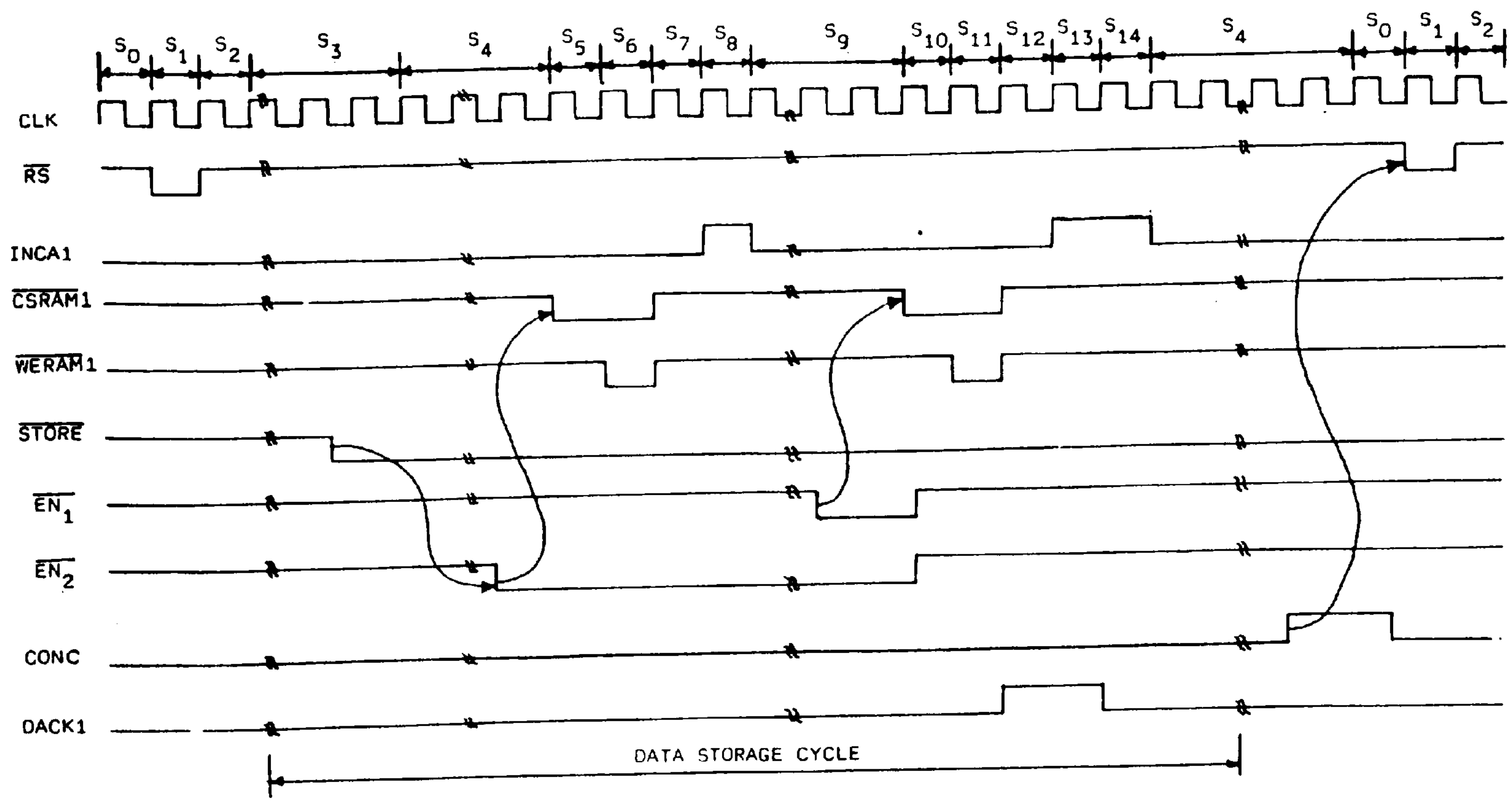


Figure 5.10 TIMING DIAGRAM FOR DATA FLOW CONTROL FOR ADC BOARD

5.5 Address Decoding

The I/O system is selected by decoding the address lines A_0 - A_9 while the processor VPA input is asserted. Redundancy is used to minimise the address decoding requirement.

Address bits A_8 and A_9 are used to divide the peripheral address field into four areas, each area consisting of 256 addresses. These areas are sub-divided again into eight smaller areas using address bits A_5 to A_7. Each of these small areas consists of 32 addresses. The actual memory map implemented for accessing the data buffer and the status register is illustrated in Figure 5.11. It is important to note that the actual addresses used by the 68000 microprocessor based CPU module are the addresses given in the figure displaced by a base address (800000Hex).

5.5.1 Status Register

The status register is the means by which the CPU module interprets when the sample operation has been completed. In addition the status register indicates the current status of the input voltage gain. i.e. the address that is generated by the autoranging circuitry. The format of the status register is illustrated in Fig 5.12.

5.6 Additional Information on the ADC Module

This chapter has described the design of the ADC Module. For further details on the design and structure of the module, reference should be made to Appendices A and B which

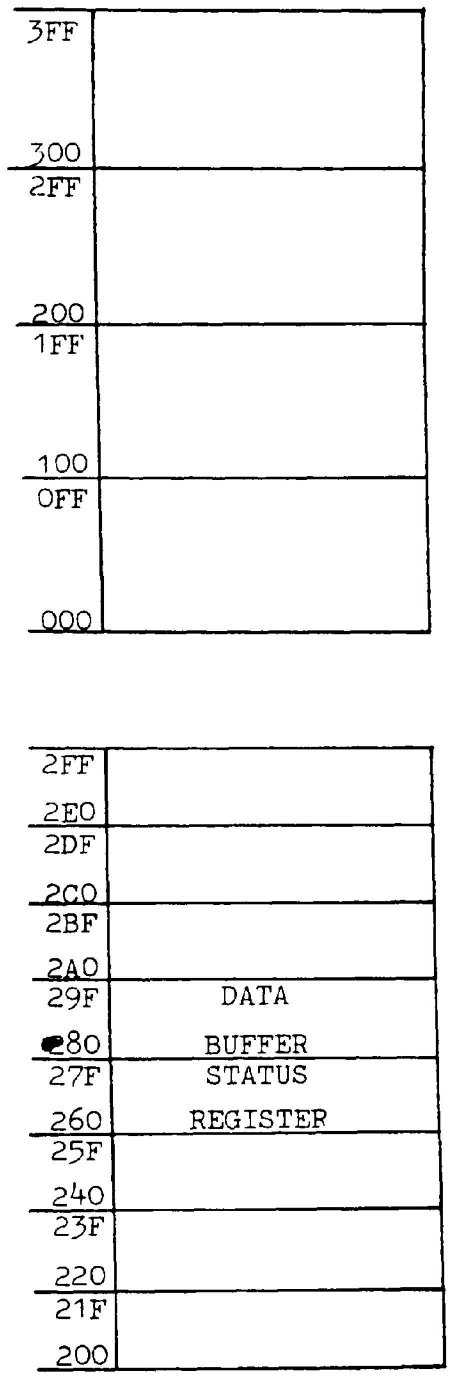


Figure 5.11 ADDRESS DECODING FOR THE DATA ACQUISITION SYSTEM

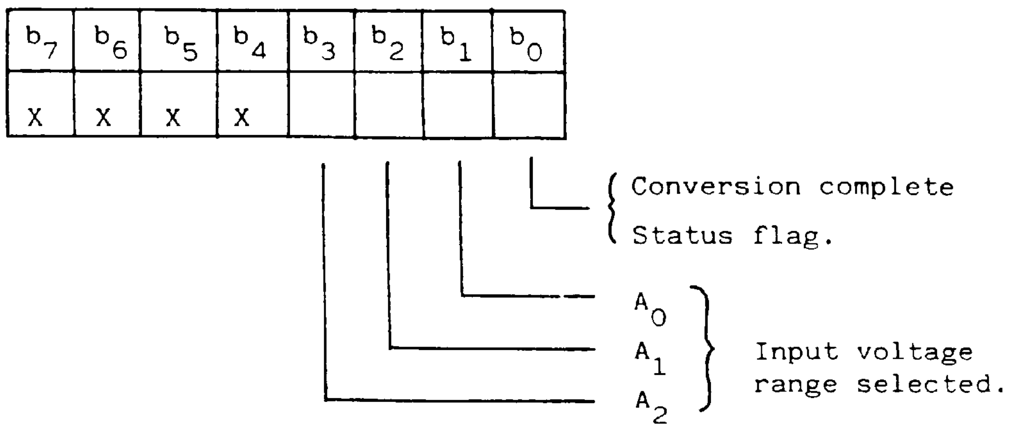


Figure 5.12 ADC Status Register

give further information relating to the circuit design/layout of the module and a high level design of the microprogrammed sequencers (The Input Processor) used to control the module.

The design of the DAC module is discussed in the following chapter.

CHAPTER 6

THE DAC MODULE DESIGN

This chapter describes the main design features of the DAC module. Detailed circuit diagrams of the design are documented in Appendix A. The DAC module consists of six functional elements which are illustrated in Figure 6.1 and outlined below.

Output Processor. This processor is implemented as 16 state microprogrammed sequencer. Its function is to control and synchronise the operation of the DAC module.

Digital to Analogue Converter. This high performance 12 bit DAC converter is the primary functional component of the DAC module. Digital test waveform samples output from the test waveform memory, are converted by this device into analogue form.

Analogue Conditioning Circuitry. The test waveform output from the DAC converter is buffered and amplified by the analogue conditioning circuitry before being applied to the system under test. In order that the instrument may be readily applied to a large number of systems, this stage contains an eight level programmable amplifier.

Test Waveform Memory. This non volatile memory is based upon a 2716 EPROM device. Digitised samples of the composite frequency waveforms have been calculated and permanently stored within this device.

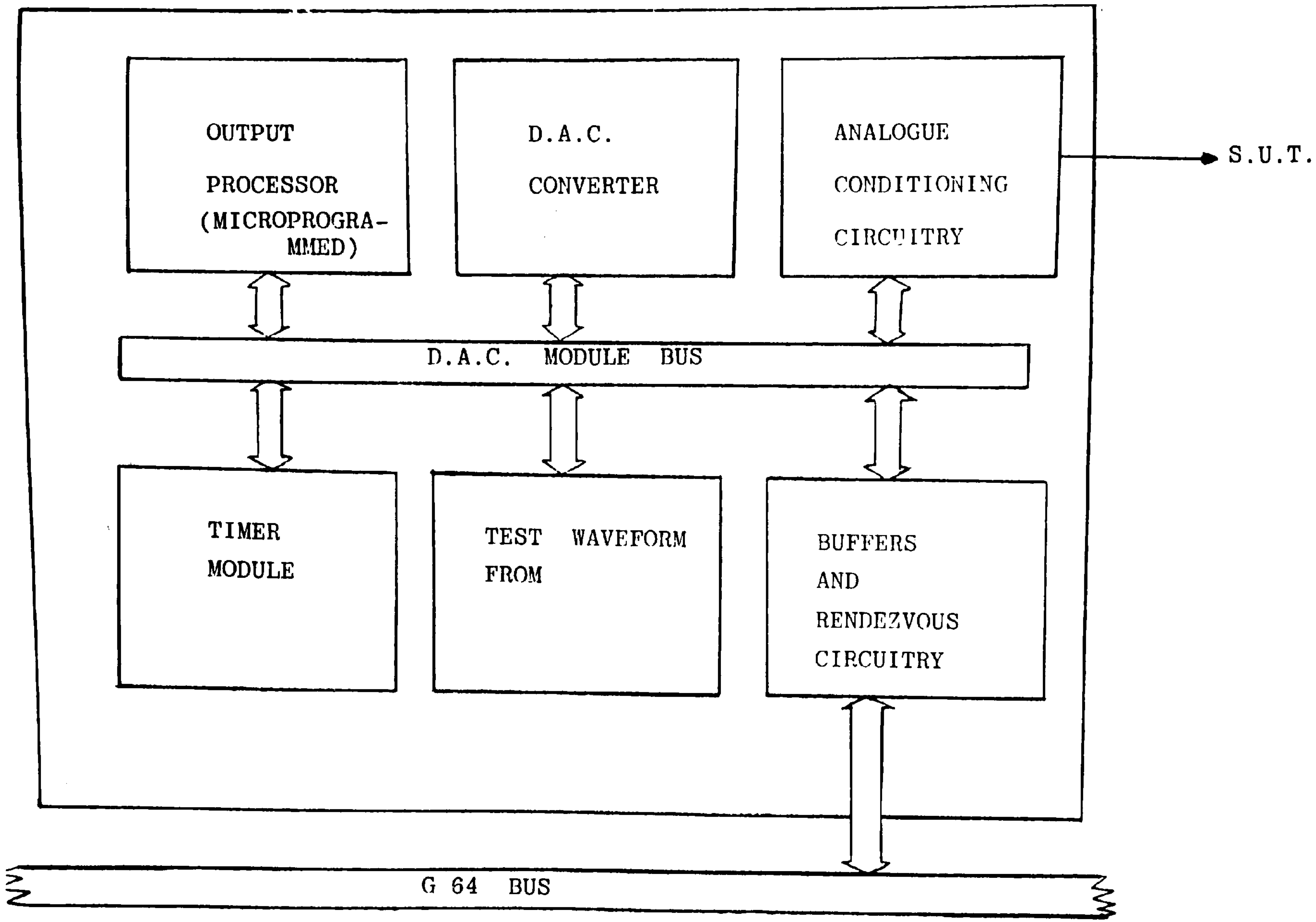


Figure 6.1 Functional Block Diagram of DAC Module

Timer Module. This module consists primarily of a programmable timer. The STS signal generated by this unit determines both the rate at which the DAC module outputs test waveform samples and the rate at which the ADC module captures samples of the system output. It is this unit which determines the frequency range over which the frequency response analysis is made.

Buffer and Rendezvous Circuitry. The main system processor controls the operation of the DAC module through this functional unit by setting the frequency of the Timer Module and the gain of the programmable amplifier in the Analogue Conditioning Circuitry. This unit also contains all the buffer circuitry used to interface the DAC module to the main bus of the frequency response analyser (the G64 bus).

The remaining sections of this chapter describe the operation and design of these units in further detail.

6.1 Overview of the DAC Module

Figure 6.2 shows the basic structure of the Test Waveform Generation System which consists of the following elements

(a) A 10 bit address counter made up of a 74LS393 dual 4 bit counter, and a 74LS93 4 bit counter (U10, U9 on DAC BOARD 1).

(b) A 2716 EPROM in which two cycles of the test signal is to be stored (U8 on DAC BOARD 1).

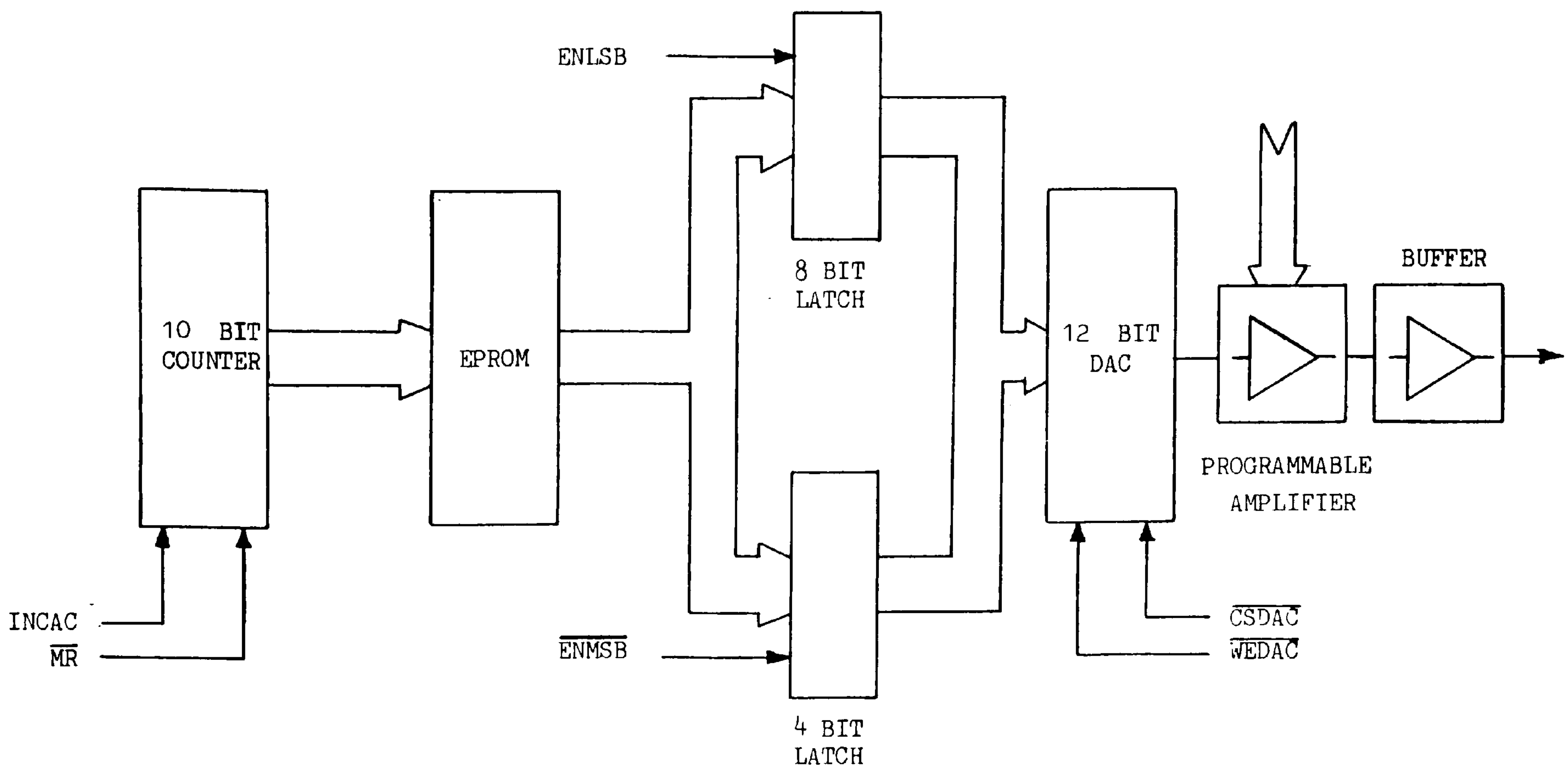


Figure 6.2 Test Waveform Generation System

(c) A 74LS373 octal latch used to latch the eight most significant bits of a 12 bit sample of the test signal.

(d) A 4 bit latch used to latch the four most significant bits of a 12 bit sample of the test signal (U6 on DAC BOARD 1).

(e) A 7545, 12 bit DAC used to convert the test signal into its analogue representation (U5 on DAC BOARD 1).

(f) A programmable amplifier consisting of two operational amplifier stages, used to condition the test signal generated by the DAC (U3, U2, U4 on DAC BOARD 1).

(g) A LM310 voltage follower, which is used as a buffer to produce the necessary driving current (U1 on DAC BOARD 1).

The purpose of the test signal processing circuitry illustrated in Fig 6.2 is to allow the stored test waveform signal to be converted into its corresponding analogue form. Initially the address counter is reset, and points to the first location in the EPROM. The corresponding data at the output of the EPROM, represents the eight least significant bits of the first 12 bit sample of the test waveform. This data is latched at the input of the 12 bit DAC via an 8 bit latch. The counter is then incremented and the four most significant bits of the 12 bit sample of the test signal are output from the EPROM. These four bits are latched at the input of the 12 bit DAC via a 4 bit latch. After a suitable period of time the 12 bit sample of the test signal is converted into its corresponding analogue representation. This process is then repeated for two complete cycles of the test signal, or for 512, 12 bit samples of data.

6.2 Digital to Analogue Convertor

The DAC used in the design of DAC module was a 7545 monolithic 12 bit CMOS multiplying converter. The main features of the DAC are outlined below:

(a) A 12 bit internal data latch.

(b) Totally monotonic over the full operating temperature range.

(c) TTL and CMOS compatible.

(d) 300ns settling time.

(e) An external voltage reference.

6.2.1 Control of the Digital to Analogue Converter

Control of the test waveform generation system is once again based upon a programmable sequencer (U11, U12, U13 on DAC BOARD 1). The timing diagram upon which the programmable sequencer operation is based is presented in Figure 6.3. The significance of the signals described in the timing diagram is given below:

(a) $\overline{\text{CSDAC}}$ - This signal is the chip select for the DAC.

(b) $\overline{\text{WEDAC}}$ - Write and enable data at the inputs of the DAC into an internal 12 bit latch, after which the digital data is

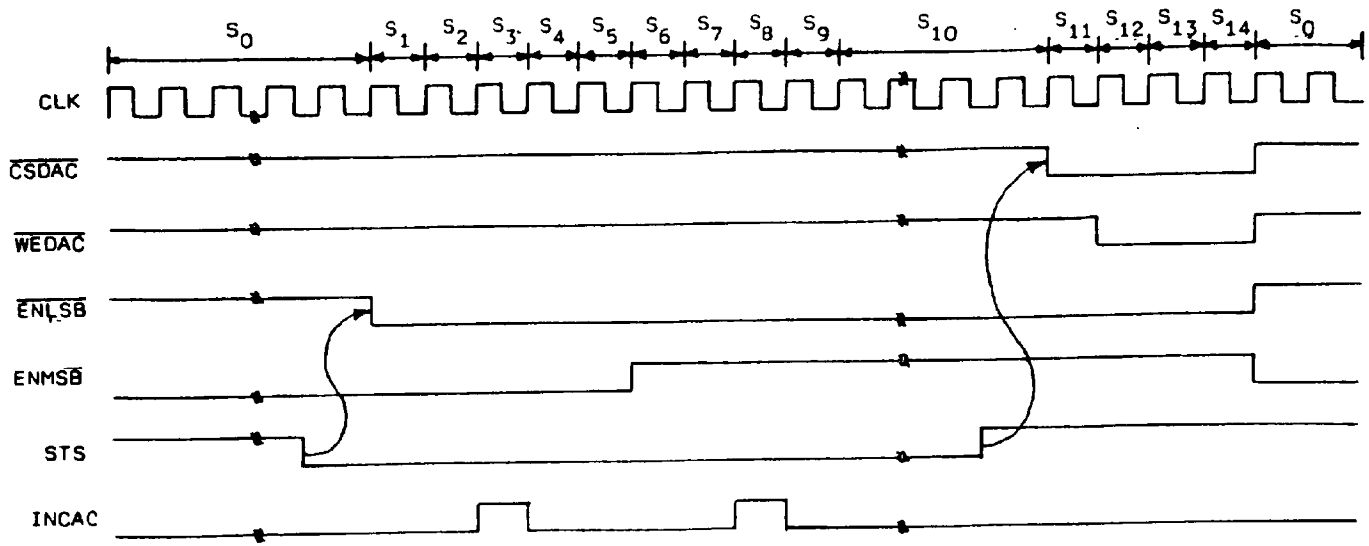


Figure 6.3 TIMING DIAGRAM FOR DATA FLOW CONTROL FOR DAC BOARD

subsequently converted into its analogue representation.

(c) $\overline{\text{ENLSB}}$ - When the least significant byte of a test signal is valid this signal is asserted, and is used to latch the least significant byte at the input of the DAC

(d) ENMSB - When the most significant byte of the test sample is available this signal is asserted and the data is latched at the inputs of the DAC.

(e) STS - The STS signal is an input to the sequencer. Its main function is to control the actual sample rate of the system.

(f) INCAC - This signal increments the address counter. However, since fourteen states of the sequencer have had to be used to produce the four signals described in (a) to (d) this signal has to be decoded via the feedback lines of the sequencer.

6.3 Analogue Conditioning Circuitry

The design of the output analogue conditioning circuitry is shown in Appendix A. This configuration allows for bipolar operation of the DAC. Figure 6.4 represents the relationship between the DAC input value and output voltage V_{OA1}.

DATA INPUT			V_{OA1}
0000	0000	0000	$-V_{REF} \left[\frac{2048}{4096}\right]$
0000	0000	0001	$-V_{REF} \left[\frac{2049}{4096}\right]$
1000	0000	0000	0 Volts
1111	1111	1111	$-V_{REF} \left[\frac{4095}{4096}\right]$
1000	0000	0000	$-V_{REF} \left[\frac{1}{4096}\right]$

FIG. 6.4

LOGIC CODING RELATING THE INPUT DATA TO THE OUTPUT VOLTAGE V_{OA1}

Considering the logic coding described in Fig 6.4 and the circuit given in Appendix A (DAC Board 1) in detail, it can be seen that the output voltage V_{OA1} of the first amplifier (U3), can be represented by equation 6.1, where b_n represents either '1' or '0' depending upon the data bit concerned.

$$V_{OA1} = -V_{REF} \, (\overline{b}_{11}.2^{11} + \sum_{n=0}^{n=10} bn.2^{n}) \qquad 6.1$$

The second stage of the analogue conditioning circuitry is a summing amplifier (U2). The relationship between the input voltage V_{OA1} , and output voltage V_{OA2} will be given by equation 6.2.

$$V_{OA2} = R_V \left[\frac{V_{REF}}{R_{V9}} + \frac{V_{OA1}}{R3}\right] \qquad 6.2$$

Hence, substituting V_{OA1} in equation 6.2;

$$V_{OA2} = R_V \left[\frac{V_{REF}}{R_{V9}} - \frac{V_{REF}}{R3} (\overline{b}_{11}.2^{11} + \sum_{n=0}^{n=10} b_n.2^{n})\right] \qquad 6.3$$

Hence, using equation 6.3 with V_{REF} = 12V, R_V = 20K ,R_{V9} = 20K , and R3 = 10K , it can be shown that the output voltage V_{OA2} varies linearly from -12 volts to +12 volts, given digital input values ranging from -2048 to +2047 respectively. Therefore, by changing the feedback resistor R_V in equation 6.3 the voltage range can be varied to accommodate the output voltage ranges given in the design specification outlined in Chapter 3.

Rearranging equation 6.3 to obtain a relationship in R_V and substituting V_{REF} = 12V, R_{V9} = 20K , R3 = 10K , and V_{OA2} = ±11.997V (maximum value), values for R_V can be calculated for the required output voltage ranges. These values are given in Fig 6.5.

VALUE FOR R_V (K)	OUTPUT VOLTAGE RANGE
13.333	$\pm$8V
10.000	$\pm$6V
8.333	$\pm$5V
6.667	$\pm$4V
3.333	$\pm$2V
1.667	$\pm$1V
0.833	$\pm$0.5V
0.208	$\pm$0.125

Figure 6.5

FEEDBACK RESISTANCE VALUES REQUIRED

FOR THE VARIOUS OUTPUT VOLTAGE RANGES

The analogue conditioning circuitry is implemented by devices U5, U3, U2, U4, U1 on DAC BOARD 1. Like the ADC analogue conditioning circuitry, the various voltage ranges are selected by an eight channel analogue switch. It is the responsibility of the main system processor to set the gain of the output amplifier when initialising the DAC module prior to a frequency response test.

6.4 Timer Module Sample Rate Generation

It is an important requirement of the analogue I/O system that the rate at which the composite test signal is generated should be user selectable. The original specification stipulated a sample rate in the range from dc to 100KHz. A programmable timer is therefore used to generate a periodic pulse train which in turn synchronises the I/O operation.

6.4.1 6840 Programmable Timer Module

The 6840 Programmable Timer Module used in the design is part of the 6800 microprocessor family, and is fully bus compatible with 68000 systems. The Programmable Timer Module (PTM) itself consists of three 16 bit counter/timers which can operate independently, and in several distinct modes to fit a wide variety of measurement and synthesis applications.

In the design of the test waveform generation system, the PTM is operated in the wave synthesis mode, allowing a square wave to be generated with a variable period depending upon a preset value programmed into the counter latch. This data is determined by the user who selects fundamental frequency (fo) for the system. The relationship between the fundamental frequency and the sample rate of the system is governed by equation

$$\text{SAMPLE RATE} = 256.f_o \quad \text{Hz} \qquad 6.4$$

The PTM is basically set up as illustrated in Fig 6.6 the

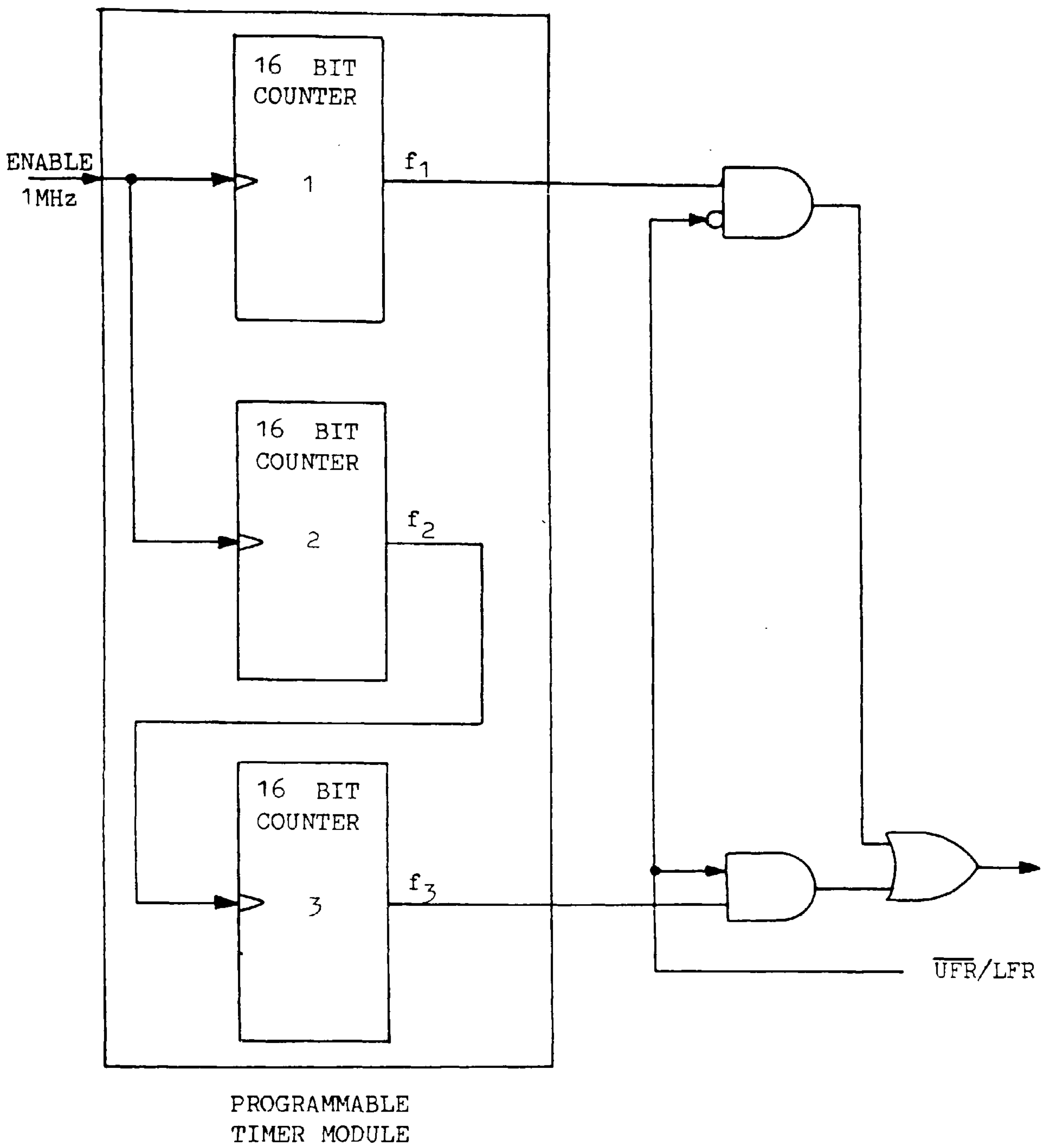


Figure 6.6 PRINCIPLE ADOPTED FOR VARYING THE SAMPLE RATE OF THE SYSTEM

first two 16 bit counters being clocked by the ENABLE (E) clock.

To set up the PTM, suitable integer count values have be to loaded into the PTM's three internal 16 bit latches. This data is transferred to the counters, when the control registers are initialised. In all the PTM has three control registers, each is associated with a particular counter/timer. Hexadecimal values stored in these control registers set the PTM into the correct mode. It is important to note that the individual timers will not operate, until a negative going START pulse is observed at the G inputs of the PTM. The addresses corresponding to the PTM registers are given in Figure 6.7.

6.5 Address Decoding

The memory map for the DAC Module is shown in Fig 6.7 and like the address decoding for the ADC, utilises the peripheral address field; ie VPA active and address lines A_0 to A_9. As with the address decoding used in the DAC Module system, the address lines A_8 and A_9 are used to divide the peripheral address field into four areas.

This chapter concludes the description of the hardware design of the new analyser. As stated in Chapter 3, the overall functional operation of the instrument is carried out under control of a 16 bit microprocessor. The following chapter gives the design of the software routines developed for this processor and in addition gives an overview of the software development enviroment.

3FF 300	
2FF 200	
1FF 100	
0FF 000	

3FF 3E0	
3DF 3C0	
3BF 3A0	
39F 380	
37F 360	
35F 340	INITIATE I/O OPERATION
33F 320	SET UP PTM
31F 300	SET UP O/P VOLTAGE RANGE

Figure 6.7 Memory Map of DAC Module

CHAPTER 7

SOFTWARE

7.1 Choice of Language

Due to the complex nature of the signal processing algorithms employed within the instrument, the bulk of the instrument software was written in a high level language (Pascal). The advantages associated with the use of a high level language as opposed to assembly language include:

Speed of software development
Reduction of Programming errors
Readability and Maintainability

However the inherent inefficiency introduced by using a compiler leads to larger machine code programs and subsequently a larger memory requirement. The availability of high density low cost memory devices however, effectively removes any real constraint on code size.

Use of a high level language compiler can also result in an increase in program execution times. The major tasks undertaken by the main system processor can be categorised as follows:

General house keeping activities
Communication with the User
Calculation of the spectral estimates of the captured signal
Calculations of the transfer function of the system

under test.

None of the above are fundamentally real time tasks and the time penalty incurred for using a high level language would, therefore, have no serious consequence. Furthermore, this reduction in execution time can be compensated for by choosing a 16 bit processor rather than an 8 bit processor as the main system processor.

High level languages, however, tend to be deficient in low level features and subsequently those routines which required direct access to hardware registers and bit manipulation functions were developed as separate assembly language subroutines.

7.2 The Development Environment

Throughout the research project use was made of a Philips Universal Microprocessor Development System - a PMDS-II. The system can be configured to provide development facilities for a number of 8 bit and 16 bit microprocessors. In its current configuration, however, support was only available for the following processors.

Motorola MC6800 (8 bit)
Motorola MC6802 (8 bit)
Motorola MC6809 (8 bit)
Motorola MC68000 (16 bit)

It was the availability of this development tool which dictated the choice of the 16 bit Motorola 68000 as the instrument's main system processor.

7.2.1 Software Tools

The PMDSII runs under a UNIX operating system and beside in-circuit emulation facilities the system offers the following software tools for 68000 based applications.

A Screen Editor: Provides facilities for keyboard entry of source code programs and stores the data entered on disk. It also provides an interactive command language for carrying out editing operations (insertions, deletions, substitutions etc.) on data being entered or previously stored on disk.

A 68000 Cross Assembler: Translates MC68000 assembly language source code programs which have been created by the editor into object code.

A Pascal Compiler: Translates Pascal source code programs created with the editor into the MC68000 object code.

A Linker: Combines object modules, resolves intermodule cross references and maps the resulting program into the address space of the target microcomputer. The resulting program module is stored on disk, from where it can be loaded into the prototype memory or into the Debuggers emulation memory for execution on the target microcomputer. Alternatively this program module may be downloaded into EPROM.

A Debugger: Provides full emulation and debug facilities. It allows emulation runs to be carried out in various combinations of prototype hardware and PMDS software. The mixture can vary from software only (simulation) to a virtually complete target circuit. Emulation can be performed in "real-time", at a variety of clock frequencies or in step mode.

7.2.2 The Development Cycle

Figure 7.1 shows a typical development cycle. It is a particularly powerful feature of the development system, in that it allows for independent assembly and/or compilation of source code modules.

7.3 Software Structure

The software developed during the study amounts to nearly two thousand lines of source code. Therefore, in order to avoid undue complexity, a modular approach to the software design was adopted. As stated in the previous section the PMDS II development system provides for the separate compilation/assembly of several program segment units or modules. Within this section a brief description of the various program modules is given. Appendices D and E give the high level design of the software and a full program listing. The design descriptions are presented in three sections.

User Interface and Signal Processing Modules :

These detail the design of the main instrument software which runs on the MC68000 M.P.U. board. It

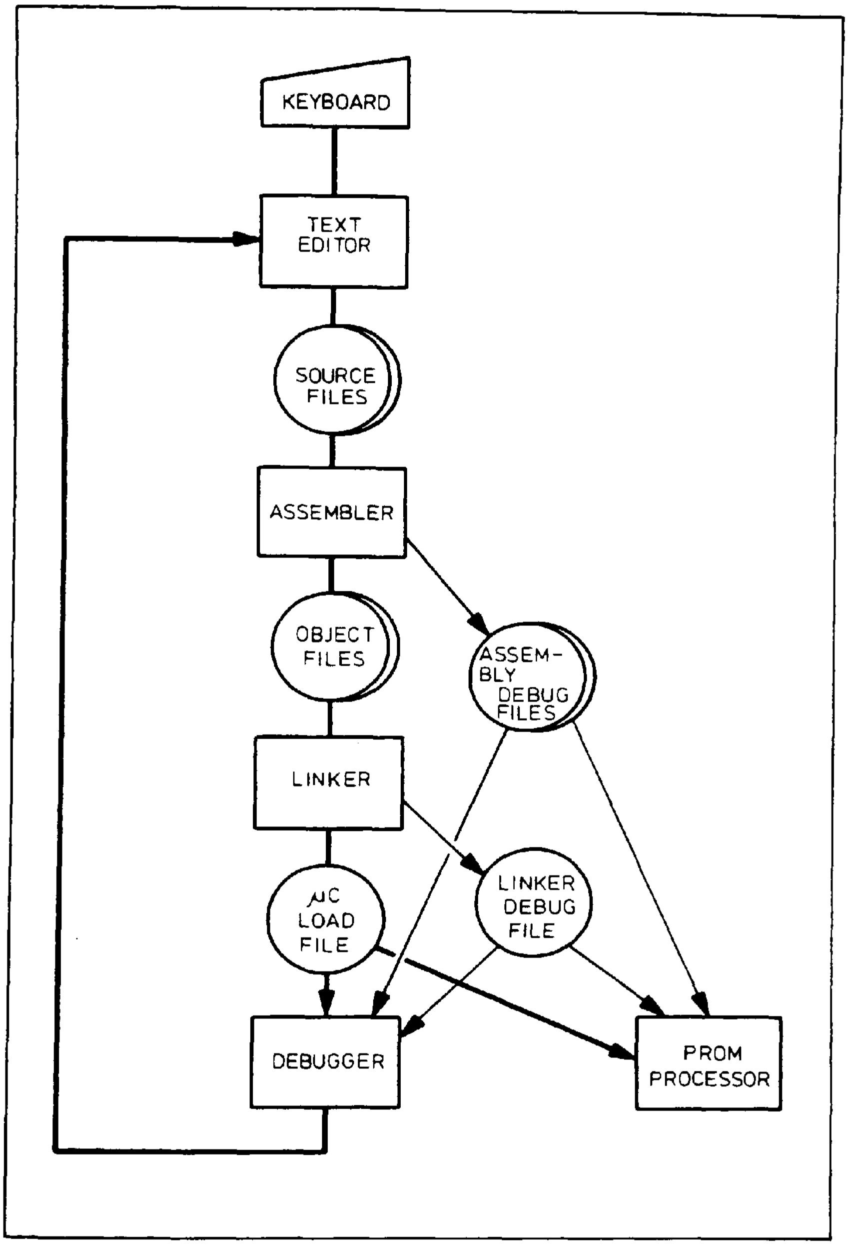


Figure 7. 1 <u>**Development Cycle**</u>

describes both the user interface and all signal processing routines.

Terminal Drive Modules :

The user exercises control of the instrument through a RS232 type alpha-numeric terminal. In order that this I/O stream may be accessed as a sequential file from within the Pascal routines, a number of driver routines had to be written to provide data transfer between the File Control Block (FCB) of the text files associated with the terminal and the USART device on the main M.P.U. Board. The description of these MC68000 assembly language routines are documented within this section.

Utility routines :

In order to facilitate in the development of the system software a number of routines have been developed which run on the microprocessor development system. These routines are used to generate the test waveform time samples and spectral estimates in the form required, in order that they may be incorporated into the frequency response analyser.

7.3.1 User Interface and Signal Processing Modules

7.3.1.1 The User Interface

In order to perform a frequency response analysis the user is required to perform the following operation.

a. Put the instrument into its frequency response analysis mode.

b. Select the test waveform to be applied to the system.

c. Select the first frequency at which the system response is to be measured.

d. Select the highest frequency at which the system response is to be measured.

e. Select the magnitude of the test waveform to be applied to the system.

These operations are carried out through a menu driven user interface. These menus are documented in figures 7.2a and 7.2b.

FREQUENCY RESPONSE INSTRUMENT COMMAND MENU

1) Obtain frequency response and transfer function of System Under Test [S.U.T.].

2) Display frequency response of last test results as Bode plots on screen.

3) Terminate instrument operations.

Which option do you require [1, 2 or 3]?

TEST WAVEFORM MENU

1) Prime M.F.T.S.

2) P.R.B.S.

PLEASE SELECT OPTION [1,2]

FUNDAMENTAL TEST FREQUENCY MENU

0) FundFreq = 0.01

1) FundFreq = 0.05

2) FundFreq = 0.10

3) FundFreq = 0.50

4) FundFreq = 1.00

5) FundFreq = 5.00

6) FundFreq = 10.00

7) FundFreq = 100.00

8) FundFreq = 500.00

9) FundFreq = 781.25

Which fundamental frequency do you require [1,2,3,4, or 5]?

Figure 7.2(a) User Menus

UPPER TEST FREQUENCY MENU

1) Freqmax = 3.6500000E+02

2) Freqmax = 2.6645000E+04

SELECT MAX. FREQUENCY [1,2]

OUTPUT MAGNITUDE MENU

FOR O/P RANGE +/_ 0.125 Volts.....PRESS KEY 1

FOR O/P RANGE +/_ 0.5 Volts.......PRESS KEY 2

FOR O/P RANGE +/_ 1.0 Volts.......PRESS KEY 3

FOR O/P RANGE +/_ 2.0 Volts.......PRESS KEY 4

FOR O/P RANGE +/_ 4.0 Volts.......PRESS KEY 5

FOR O/P RANGE +/_ 5.0 Volts.......PRESS KEY 6

FOR O/P RANGE +/_ 6.0 Volts.......PRESS KEY 7

FOR O/P RANGE +/_ 8.0 Volts.......PRESS KEY 8

PLEASE SELECT OUTPUT MAGNITUDE

Figure 7.2(b) User Menus

7.3.1.2 Module Descriptions

This section gives a brief description of the user interface and signal processing routines. High level designs and program listings are given in appendices C and D.

Module fridft:s Provides the user access to all system facilities through a main menu.

Module ftest:s Calls all the routine necessary in order to perform a frequency response analysis. In order to present an overview of this software Figure 7.3 gives a high-level pseudo code design of this module.

Module ffreq:s Called by the main ftest:s module. This module is used to determine the initial test waveform period and the number of test repetitions required, in order to provide the required band coverage. The modules prompts the user for the test parameters it requires through two menus.

Module Ginn:s Called by the main ftest module. This module uses a menu to prompt the user for the required test waveform magnitude. Also contained within this file is the routine which calculates the PTM (see Chapter 6) register values required to carry out a test at the chosen frequency.

Module range:s - Called by main ftest module. This module sets the type of waveform to be used, the magnitude of the waveform and the frequency of the waveform on the DAC board. The module then initialises a frequency response test, waits for completion and loads an integer array with the 256 bytes of captured data.

Module fft:s - Called by the main ftest:s module, this module calculates the Fourier transform of a complex real array of captured data and returns it in the same array. The module uses a 256 point F.F.T. algorithm (Chapter 3).

Module angmag: - Called by main ftest:s module this converts the passed array of spectral estimates from cartesian to polar notation.

Module specest:s - Called by main ftest:s module this module loads an array with the spectral estimates of the test signal used.

Module tranfn:s - Called by main ftest:s module this module calculates the transfer function of the system under test from the array of spectral estimates of the captured data and the array of spectral estimates of the test waveform used. The results are displayed on the user screen.

7.3.2 Terminal Driver Modules

Basic I/O in Pascal is performed using a number of file handling procedures. The Pascal compiler available on the PMDSII development system however does not implement these procedures. In order that information may be transfered to and from a terminal from within the Pascal program, it was necessary to create the data structures and write all routines to implement file I/O through the compiler File Control Block (FCB). See Figure 7.4. These routines which

MODULE - ftest:s

FUNCTION - obtain the transfer function of the system under test;

EXPORTS SystemTest;

FROM pause:s IMPORTS PAUSE;

FROM VDU2:s IMPORTS ClearScreen;

FROM tfreq:s IMPORTS TestFreq;

FROM Ginn:s IMPORTS RAN,OP;

FROM range:s IMPORTS INIT;

FROM ft:s IMPORTS FFT;

FROM angmag:s IMPORTS ANG;

FROM pow:s IMPORTS POWELL;

FROM prbs:s IMPORTS PRBS;

FROM tranfn:s IMPORTS TF;

BEGIN SystemTest.

REPEAT

Clear terminal screen (ClearScreen):

Display waveform menu;

Input waveform choice and determine number of component hamonics;

UNTIL valid choice is made;

Clear terminal screen (ClearScreen);

Input required frequency range from user and determine initial test waveform period and number of test repititions (TestFreq);

Input required output waveform magnitude (RAN);

Select waveform required;

Figure 7.3 High level design of main frequency response test module

```
FOR each test repetition DO

      Determine register value for timer on DAC board (OP);

      Set waveform output range, set timer on DAC board and
      initalise test (INIT);

      Read settings of autorange circuitry after test;

      Load relavent complex real array with captured integer
      values;

END;

FOR each test repetition DO

      Calculate fourier transform of captured data (FFT);

      Calculate polar rotation (ANG);

      Initialise array with spectral estimates of output waveform
      (SpecEst);

      Calculate the transfer function of the system under test (TF);

END SystemTest.
```

Figure 7.3 (Cont.) High level design of main frequency response test module

The layout of a file control block for the 68000 is:

```
type FCB = record
    BUFP       : pointer : {4 bytes}
    RECL       : integer : {2 bytes; record length in bits}
    LB         : integer : {2 bytes; lower bound of items in
                            file}
    UB         : integer : {2 bytes; upper bound of items in
                            file}
    STATUS     : integer : {2 bytes; the meaning of the bits
                            from left to right is:}
                            bit 0: LB UB EFFECTIVE
                            0=LB,UB not effective
                            1=LB,UB effective
                                 LB,UB are effective for files
                                 of type subrange.
                            bit 1: EOF; 0=not eof, 1=eof
                            bit 2: EOLN; 0=not eoln; 1=eoln
                            bit 3: INSPECTION;
                            bit 4: GENERATION;
                            bit 5: TEXTFILE;
                            bit 6-7: claimed for extension;
                            bit 8-15: free for user
FILENAME: packed array[1..6] of char;
                                 {6  character filename. Only
                                 for files mentioned in the
                                 program heading (first 6
                                 characters). For other files
                                 (internal files) FILENAME is
                                 set to 6 spaces}
SCRATCH: packed array [1..4] of char;
                                 {4 bytes area free for user}
end;
```

Before the first call of rewrite or reset, RECL, UB, LB are filled as applicable. For textfiles, RECL = 8, because for textfiles a record is 1 character (8 bits);STATUS bit 0 (LB UB EFFECTIVE) and bit 5 (TEXTFILE) are filled as applicable, other bits are 0; if LB UB EFFECTIVE =0 then also UB and LB are 0; FILENAME is filled as applicable.

Before any call of reset or rewrite and before the user's assembly routine is called, BUFP is set to the allocated file buffer address.

Figure 7.4 Pascal File Control Block

are specified in the PMDSII Pascal Reference Manual were written for the SN2661 Universal Synchronous, Asynchronous Receive and Transmit device (USART) which provides the interface between the 68000 main processor card and the instruments terminal. These modules are implemented in 68000 assembly language.

7.3.2.1 Module Definitions

Module rewrit:s This module sets up the file output buffer and initialises the SCN 2661 USART into the following configuration - asynchronous communication, eight bits, no parity, one stop bit, internal clock 9600 baud.

Module reset:s This module sets up the output file buffer, enters a space character into the input file buffer and sets EOLN in the File Control Block.

Module get:s This module loads the input file buffer with the character received from the terminal.

Module writln:s Outputs a 'carriage return' and a 'line feed' character to the terminal.

Module Put:s Transmits the character in the output file buffer to the terminal through the SN2661 device.

7.3.3 Utility Routines

A number of routines have been written to assist in the generation of the test waveform time samples and spectral estimates incorporated into the instrument. Figure 7.5

shows how these routines are used. The routines are listed in Appendix D.

Optsin routine This routine generates 256 time samples of the optimum prime composite waveform and scales the samples into the 0 to 4095 (12 bits range required by the instrument).

Fft routine This variable point Cooley Tukey F.F.T. routine is used to calculate the spectral estimates (testft:s) of the test waveforms which are incorporated into the instrument.

Aryfil routine This routine uses the spectral estimates calculated by the fft routine generate to the 'spcest' program module - part of the instrument software which calculates the transfer function of the system under test.

Romfil Routine Test waveform time samples generated by the opsin routine are used to generate a 68000 assembly language file. This file can be converted using the 68000 assembler and linker into a Motorola S record file which can be downloaded into the Test Waveform EPROM described in Chapter 6.

This chapter concludes the description of the developed analyser. The following chapter evaluates the accuracy of the new analyser and presents the results of a number of frequency response measurements obtained using both the Prime composite test waveform and the MPRBS signal.

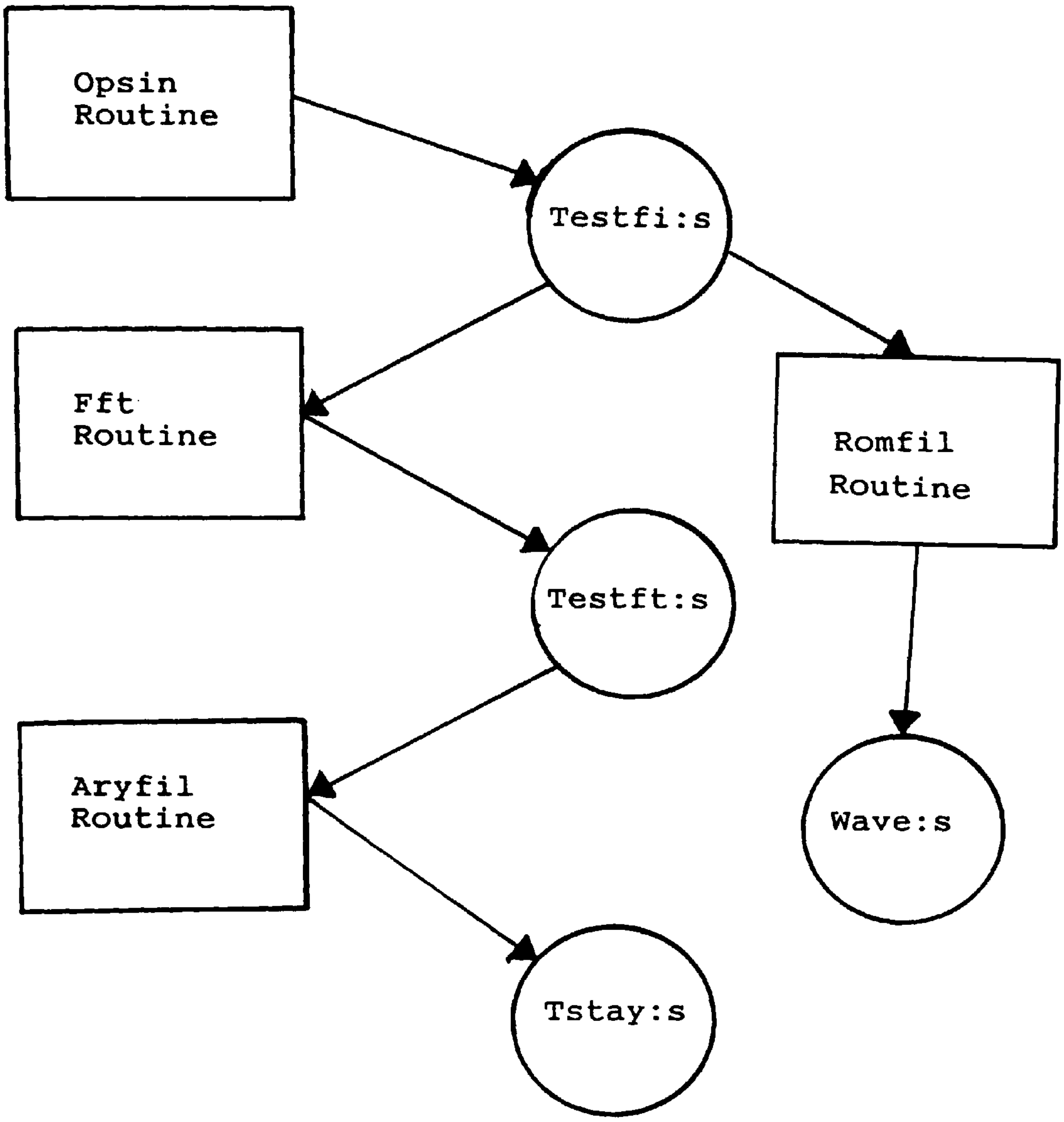


Figure 7.4 Utility Routines

CHAPTER 8

EVALUATION OF THE NEW INSTRUMENT

Two approaches were adopted in the evaluation of the developed instrument. The first approach was to use the instrument to test a system with a well defined transfer function. The second approach was to apply the instrument to a number of linear and non-linear systems and compare the results of the analysis with those obtained using a conventional frequency response analyser.

8.1 Performance Testing

In order to evaluate the accuracy of the new instrument, both its test signals were applied to a system with a known gain and phase response. The system chosen was a metre length of coaxial cable, as it possessed unity gain and zero phase shift over the entire frequency testing range of the instrument. Figure 8.1 shows the set up adopted for the performance tests. Using this configuration the accuracy of the Magnitude and Phase estimates were evaluated thereby enabling the highest frequency at which the instrument could be used to be determined.

8.1.1. Magnitude and Phase Accuracy

Tests were performed on the setup shown in Figure 8.1 using both the prime composite test waveform and the modified P.R.B.S. waveform. As described in Chapters 5 and 6 the developed instrument has eight different output voltage ranges and an eight level autoranged input stage. In order to obtain a figure for the magnitude and phase accuracy of the instrument, frequency response measurements were made using the maximum and minimum output voltage ranges. Since the system under test exhibits unity gain those tests also use the maximum and minimum voltage gains of the autoranging instrument input stage.

Frequency response tests were repeated several times for each of the test waveforms incorporated within the analyser. Multiple tests were used to establish the measurement consistency of the instrument and to identify any deterministic noise which may be present in the measurement channel. Figures 8.2 and 8.3 show a typical result of a system measurement, at an output voltage of ±8 Volts, using both the prime composite test waveform and the Modified PRBS waveform respectively. A fundamental test frequency of 1Hz was adopted during these tests.

As can be seen from both Figure 8.2 and Figure 8.3 the results obtained using the new analyser correspond very closely with those which can be theoretically expected. The magnitude response measured indicates an average gain which differs from unity by only 0.4%. This offset however can be directly compensated for by trimming the gain in the measurement channel. In the current configuration the gain is adjusted as accurately as possible within the capability of the gain adjusting potentiometers used (see Appendix A). If this gain

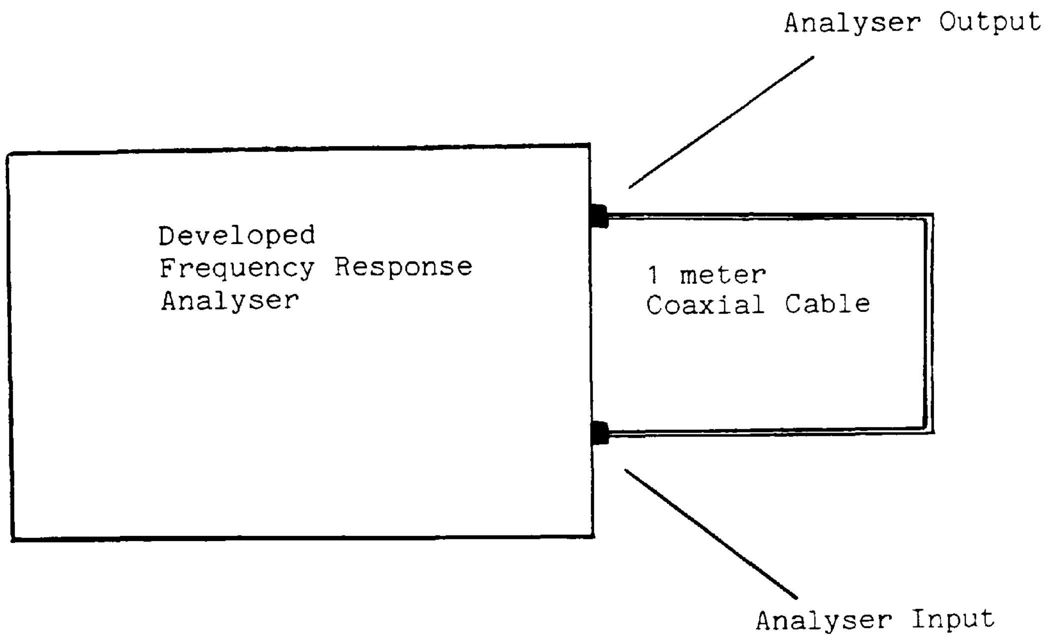


Figure 8.1 Performance Testing Configuration

'offset' proves to be significant then either higher performance potentiometers must be utilised or the instrument software adjusted to compensate for the gain shift in the measured response. Taking this into account then the observed magnitude accuracy measured varies from 0.2% using the Prime Composite Test Waveform to 0.9% using the Modified PRBS waveform. Phase accuracy of the instrument on the other hand was observed to vary from 0.117 degrees for the Prime Composite Waveform to 0.79 degrees for the Modified PRBS Waveform.

Figures 8.4 and 8.5 show typical results of the Magnitude and Phase accuracy tests using an output voltage range of ±125mV. These results demonstrate the decreased accuracy of the instrument at low test waveform voltage levels. In the case of the Prime Composite Waveform the observed magnitude errors have increased substantially to give a maximum error of approximately 11% while the phase error has increased to a maximum of 7.06 degrees. The low voltage levels have a similar effect on the instrument accuracy when using the Modified PRBS waveform. The measured maximum magnitude error was 13% while the maximum phase error rose to 8.687 degrees.

Although the results obtained using the new Frequency Response Analyser show a high degree of agreement with those which can be theoretically expected, the potential accuracy of the instrument is not fully realised. The size and random nature of the deviation from the ideal result suggests that any measurement error which is experienced is primarily due to electrical noise within the instrument as opposed to quantisation 'noise' introduced by the ADC. (The input of the ADC is autoranged)

Furthermore, it may be seen from the tests performed that the magnitude and phase errors are slightly higher in the case of the Modified PRBS signal than for that of the Prime signal. This can be expected since, despite the fact that the Modified PRBS signal possesses a smaller Peak factor, the signal has more component test harmonics which have the effect of reducing the signal to noise ratio of the individual test spectra. In addition, the frequency domain envelope of the signal is not rectangular as is the case with the Prime waveform (See Appendix F). As a result, the magnitudes of the component harmonic test frequencies are correspondingly smaller in the case of the Modified PRBS waveform than in the prime waveform and consequently they are more susceptible to corruption by electrical noise.

It is also important to note from the results obtained that the accuracy of the instrument does not deteriorate as the frequency of the test harmonics increase. The results obtained for both the Modified PRBS and Prime signals show no errors which can be directly attributable to the effects of aliasing and as such show the effectiveness of the system response formula derived in Chapter4.

Whereas great effort has been made to reduce the effect of noise in the instrument design it is obvious that the full accuracy of the instrument will not be realised until the ADC and DAC modules are implemented using Printed Circuit Board technology.

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
0.000	1.134	0.000
3.000	1.006	359.938
5.000	1.004	359.904
7.000	1.006	0.023
11.000	1.003	359.859
13.000	1.005	359.997
17.000	1.004	359.966
19.000	1.005	0.031
23.000	1.005	359.984
29.000	1.006	359.937
31.000	1.003	359.907
37.000	1.003	0.058
41.000	1.003	0.048
43.000	1.003	0.020
47.000	1.003	0.177
53.000	1.002	359.900
59.000	1.003	359.939
61.000	1.002	0.134
67.000	1.002	0.017
71.000	1.006	0.002
73.000	1.003	359.965

Press Return to Continue

Test Parameters

Fundamental Test Frequency - 1Hz
Output Voltage Range - $\pm$ 8V

Figure 8.2 Measured Frequency Response of Coaxial Cable
Test Waveform - Prime Composite

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
0.000	1.140	0.000
1.000	1.003	359.951
2.000	1.003	0.047
3.000	1.002	359.857
4.000	1.005	0.041
5.000	1.003	0.098
6.000	1.003	0.043
7.000	1.002	359.906
8.000	1.002	359.950
9.000	1.002	0.046
10.000	1.005	359.960
11.000	1.001	0.442
12.000	1.004	0.021
13.000	1.001	0.271
14.000	1.004	0.045
15.000	1.001	359.908
16.000	0.997	359.776
17.000	1.002	359.983
18.000	1.002	359.911
19.000	1.002	0.102
20.000	1.004	359.983
21.000	1.004	0.134
22.000	1.001	0.048
23.000	0.997	0.093
24.000	0.999	0.173
25.000	1.006	359.973
26.000	1.004	359.910
27.000	1.004	0.395
28.000	1.002	359.892
29.000	1.005	0.008
30.000	1.003	0.035
31.000	1.000	0.054
32.000	1.003	0.032
33.000	1.004	359.840
34.000	1.001	359.849
35.000	1.001	359.593
36.000	1.005	359.846
37.000	1.013	359.742
38.000	1.001	359.934
39.000	1.002	359.817
40.000	1.007	0.117
41.000	1.005	0.471
42.000	1.001	359.988

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
43.000	1.003	0.144
44.000	1.000	359.757
45.000	1.002	0.124
46.000	1.002	0.310
47.000	1.003	359.946
48.000	1.006	359.890
49.000	1.007	359.780
50.000	1.007	0.132
51.000	0.997	0.014
52.000	0.999	359.796
53.000	1.003	0.011
54.000	1.003	0.190
55.000	1.001	0.062
56.000	1.000	0.095
57.000	1.002	359.950
58.000	1.006	0.032
59.000	1.001	359.966
60.000	1.000	359.866
61.000	1.004	359.845
62.000	1.004	359.798
63.000	1.004	0.047
64.000	1.002	0.164
65.000	1.002	359.892
66.000	1.001	0.124
67.000	0.999	0.005
68.000	1.004	359.947
69.000	1.000	0.298
70.000	1.002	0.146
71.000	1.004	359.913
72.000	1.002	0.030
73.000	0.995	0.140
74.000	1.006	0.719
75.000	1.003	359.958
76.000	1.005	0.020
77.000	1.003	359.989
78.000	0.998	0.065
79.000	1.001	359.899
80.000	1.001	0.103
81.000	1.004	359.915
82.000	0.999	0.037
83.000	1.002	0.005
84.000	1.003	0.024
85.000	1.000	359.846
86.000	1.000	359.997

Figure 8.3 Measured Frequency Response of Coaxial Cable
Test Waveform - Modifed PRBS

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
87.000	1.004	0.014
88.000	1.003	0.126
89.000	0.999	359.611
90.000	0.998	359.623
91.000	1.001	359.883
92.000	1.000	359.901
93.000	1.001	359.846
94.000	1.001	359.978
95.000	1.001	0.039
96.000	1.002	359.908
97.000	1.002	359.897
98.000	1.002	359.892
99.000	1.002	359.920
100.000	1.001	0.121
101.000	1.003	0.093
102.000	1.003	359.985
103.000	0.999	359.838
104.000	0.997	0.264
105.000	1.002	359.939
106.000	1.003	359.886
107.000	1.001	359.926
108.000	1.008	359.554

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
109.000	1.000	359.826
110.000	1.003	0.009
111.000	0.997	359.886
112.000	0.999	359.862
113.000	1.001	0.094
114.000	0.995	0.064
115.000	1.002	359.864
116.000	1.000	0.018
117.000	1.003	359.959
118.000	1.002	0.328
119.000	1.001	359.904
120.000	0.997	0.253
121.000	1.003	359.948
122.000	1.003	359.833
123.000	1.002	359.976
124.000	0.999	0.195
125.000	1.004	0.632
126.000	1.002	359.964

Test Parameters

Fundamental Test Frequency - 1Hz
Output Voltage Range - $\pm$8V

Figure 8.3 (Cont'd) Measured Frequency Response of Coaxial Cable
Test Waveform - Modified PRBS

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
0.000	14.170	0.000
3.000	1.050	359.691
5.000	1.049	0.313
7.000	1.032	0.343
11.000	1.087	0.082
13.000	1.014	0.665
17.000	1.033	1.162
19.000	1.041	1.983
23.000	1.092	0.529
29.000	1.032	1.237
31.000	1.059	358.798
37.000	1.005	357.235
41.000	1.040	0.245
43.000	1.022	352.940
47.000	1.027	359.478
53.000	1.036	1.576
59.000	1.016	359.048
61.000	1.080	357.207
67.000	1.026	359.811
71.000	1.045	0.289
73.000	0.982	357.141

Test Parameters

Fundamental Test Frequency - 1Hz
Output Voltage Range - $\pm$125mV

Figure 8.4 Measured Response of Coaxial Cable
Test Waveform - Prime Composite

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
0.000	14.057	0.000
1.000	0.990	359.396
2.000	1.005	359.564
3.000	1.017	359.848
4.000	1.037	1.432
5.000	1.023	4.142
6.000	1.040	352.989
7.000	1.018	359.478
8.000	1.049	358.367
9.000	1.070	359.472
10.000	1.034	358.356
11.000	0.981	3.568
12.000	1.037	3.641
13.000	1.040	358.881
14.000	0.975	357.547
15.000	1.035	1.059
16.000	1.078	357.789
17.000	1.016	0.876
18.000	1.028	3.200
19.000	0.958	358.370
20.000	1.024	359.084
21.000	1.068	2.097
22.000	1.086	2.643
23.000	0.996	357.159
24.000	1.072	5.956
25.000	0.964	357.309
26.000	1.004	0.649
27.000	1.025	356.675
28.000	1.033	356.204
29.000	0.998	1.239
30.000	0.974	6.637
31.000	1.004	0.059
32.000	0.994	3.523
33.000	1.025	1.683
34.000	1.119	0.582
35.000	0.989	0.403
36.000	1.066	359.238
37.000	1.024	8.687
38.000	1.004	358.207
39.000	1.002	2.415
40.000	1.053	2.025
41.000	1.038	357.289
42.000	1.046	0.074

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
43.000	1.097	359.682
44.000	1.020	357.528
45.000	1.137	358.398
46.000	0.993	358.354
47.000	1.009	0.636
48.000	1.010	358.282
49.000	1.029	0.994
50.000	1.086	2.435
51.000	1.037	359.291
52.000	1.077	0.439
53.000	0.982	0.090
54.000	1.071	3.759
55.000	1.033	357.878
56.000	1.026	1.265
57.000	0.984	357.107
58.000	1.021	0.258
59.000	1.046	1.684
60.000	1.014	2.651
61.000	0.983	359.798
62.000	1.009	1.597
63.000	1.041	1.127
64.000	1.010	9.467
65.000	1.050	0.603
66.000	0.972	354.631
67.000	1.131	359.478
68.000	1.022	356.950
69.000	1.167	351.552
70.000	1.034	356.178
71.000	1.046	2.692
72.000	1.057	1.178
73.000	1.125	359.803
74.000	1.057	3.600
75.000	1.079	0.866
76.000	1.033	1.638
77.000	1.072	0.669
78.000	1.001	2.530
79.000	1.164	354.655
80.000	1.045	2.747
81.000	1.018	359.988
82.000	1.055	0.950
83.000	1.111	359.070
84.000	0.997	0.001
85.000	0.992	355.353
86.000	1.013	0.412

Figure 8.5 Measured Frequency Response of Coaxial Cable Test Waveform - Modified PRBS

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
87.000	1.037	354.898
88.000	1.017	1.304
89.000	1.099	352.783
90.000	0.960	354.598
91.000	1.021	359.654
92.000	1.179	5.666
93.000	1.046	358.645
94.000	0.998	359.364
95.000	0.957	359.541
96.000	1.026	0.898
97.000	1.046	2.772
98.000	1.000	1.182
99.000	1.094	359.955
100.000	0.997	4.306
101.000	1.013	0.275
102.000	1.019	0.744
103.000	1.066	0.124
104.000	0.951	2.273
105.000	1.044	2.165
106.000	1.111	1.201
107.000	1.060	358.247
108.000	0.971	5.782

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
109.000	1.054	4.064
110.000	1.031	2.564
111.000	1.022	351.698
112.000	0.938	6.986
113.000	1.046	358.978
114.000	1.217	7.239
115.000	1.045	2.570
116.000	0.982	358.483
117.000	1.050	0.560
118.000	1.037	353.293
119.000	1.076	4.495
120.000	1.073	0.798
121.000	1.066	358.619
122.000	0.998	358.190
123.000	1.025	3.505
124.000	0.993	357.377
125.000	0.996	3.228
126.000	0.971	0.289

Test Parameters

Fundamental Test Frequency - 1Hz
Output Voltage Range - $\pm$125mV

Figure 8.5 (Cont'd) Measured Frequency Response of Coaxial Cable
Test Waveform - Modified PRBS

8.1.2 Test Bandwidth

The range of frequencies over which a system response may be measured was determined using the test arrangement depicted in Figure 8.1. By applying the composite test signals to a system which to all practical purposes has an infinite bandwidth, then the upper frequency limit at which the instrument may be used to obtain a system response may be determined. The upper limit will in general be reached when the observed system response falls by 3dB or the instrument fails in some manner. In the tests carried out, the fundamental test frequency was increased until one of these conditions was met. It was found that the instrument eventually failed when the fundamental test frequency used was raised to around 780Hz. This gave a maximum test frequency of around 57KHz for the Prime Composite Waveform and around 100KHz for the Modified PRBS waveform.

Typical results of the system tests carried out using the two waveforms are shown in Figs. 8.6 and 8.7.

As the test frequency was raised above these upper limits the instrument failed to complete a system test. It was observed that the ADC module failed to complete its full test sequence and so caused the instrument to 'tie up'. This ADC module failure at very high sample rates results directly from the inability of the ADC to convert and store the resulting sample before the next sample cycle is initiated.

If a substantially higher sample rate is required then both the conversion time of the ADC (currently 6 microseconds) and the rate at which a converted sample can be stored in the RAM need to be increased. The current lack of availability of low cost, 12 bit, very high speed Analogue to Digital Converters curtails the first course of action. However it is envisaged that the highest frequency at which the instrument can measure a system response may be doubled to 100KHz, by using high performance devices in the design of the ADC control sequencers and by decoupling the ADC clock and the sequencer clock.

Figures 8.6 and 8.7 also show that above 50KHz the gain measured by the instrument gradually increases. This may be attributed to a rise in gain in the instruments measurement channel, so that any attempt to increase the bandwidth above 50KHz will mean that this gain increase problem would have to be addressed.

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
0.000	1.115	0.000
2343.750	0.991	359.912
3906.250	0.999	0.336
5468.750	1.014	1.725
8593.750	0.983	1.519
10156.250	1.009	0.085
13281.250	1.003	2.250
14843.750	0.994	362.872
17968.750	1.015	2.362
22656.250	1.004	0.170
24218.750	0.977	4.613
28906.250	1.019	3.886
32031.250	1.075	4.265
33593.750	1.061	359.841
36718.750	1.057	4.708
41406.250	1.047	363.781
46093.750	1.071	366.480
47656.250	1.076	6.088
52343.750	1.070	4.066
55468.750	1.115	4.971
57031.250	1.112	3.440

Test Parameters

Fundamental Test Frequency - 781.25Hz
Output Voltage Range - $\pm$8V

Figure 8.6 Measured Response of Coaxial Cable Test Waveform - Prime Composite

FREQ(Hz).	MAGN(T.F.)	PH(T.F.)
0.000	1.133	0.000
781.250	0.992	359.115
1562.500	1.000	1.152
2343.750	0.996	357.878
3125.000	0.983	0.499
3906.250	1.008	358.752
4687.500	0.978	356.416
5468.750	1.003	6.801
6250.000	1.003	359.348
7031.250	0.992	1.774
7812.500	1.003	357.764
8593.750	0.981	2.703
9375.000	0.983	5.111
10156.250	0.993	0.510
10937.500	0.990	356.154
11718.750	0.992	361.434
12500.000	0.986	0.335
13281.250	0.994	359.545
14062.500	0.999	0.504
14843.750	0.997	1.320
15625.000	0.998	3.320
16406.250	1.003	1.250
17187.500	0.994	1.961
17968.750	1.017	1.139
18750.000	1.004	5.807
19531.250	0.989	2.524
20312.500	1.006	359.716
21093.750	1.001	359.960
21875.000	1.011	2.839
22656.250	1.014	361.898
23437.500	0.977	362.586
24218.750	1.013	1.977
25000.000	0.997	358.369
25781.250	1.007	0.045
26562.500	1.017	0.459
27343.750	1.003	2.066
28125.000	1.037	359.322
28906.250	0.989	0.737
29687.500	1.013	2.626
30468.750	1.003	1.832
31250.000	0.980	3.260
32031.250	1.033	357.794
32812.500	1.001	1.611

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
33593.750	1.018	363.181
34375.000	1.003	2.207
35156.250	1.002	361.993
35937.500	1.064	364.318
36718.750	1.014	4.815
37500.000	1.034	3.255
38281.250	1.033	2.110
39062.500	1.012	2.398
39843.750	1.025	0.760
40625.000	0.989	2.189
41406.250	1.016	363.220
42187.500	1.031	363.333
42968.750	1.037	3.296
43750.000	1.069	2.784
44531.250	1.031	362.594
45312.500	1.042	362.968
46093.750	1.050	3.280
46875.000	1.026	367.178
47656.250	1.028	4.121
48437.500	1.045	2.469
49218.750	1.038	3.350
50000.000	1.037	4.880
50781.250	1.043	363.205
51562.500	1.046	2.165
52343.750	1.064	4.024
53125.000	1.034	363.347
53906.250	1.030	3.767
54687.500	1.082	3.375
55468.750	1.036	368.100
56250.000	1.061	7.558
57031.250	0.999	5.706
57812.500	0.973	364.479
58593.750	1.067	363.542
59375.000	1.048	0.815
60156.250	1.059	362.474
60937.500	1.044	361.981
61718.750	1.054	364.939
62500.000	1.092	2.340
63281.250	1.041	2.337
64062.500	1.083	0.172
64843.750	1.090	359.843
65625.000	1.056	3.543
66406.250	1.101	2.033
67187.500	1.083	362.160

Figure 8.7 Measured Frequency Response of Coaxial Cable Test Waveform - Modified PRBS

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
67968.750	1.059	3.742
68750.000	1.079	362.872
69531.250	1.112	363.676
70312.500	1.100	365.793
71093.750	1.089	3.650
71875.000	1.078	362.179
72656.250	1.091	3.954
73437.500	1.119	3.312
74218.750	1.082	363.133
75000.000	1.089	2.412
75781.250	1.094	363.173
76562.500	1.065	3.289
77343.750	1.091	360.950
78125.000	1.097	4.272
78906.250	1.102	363.381
79687.500	1.103	2.337
80468.750	1.116	3.074
81250.000	1.066	1.465
82031.250	1.107	363.603
82812.500	1.101	361.779
83593.750	1.114	2.354
84375.000	1.126	366.250

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
85156.250	1.134	4.719
85937.500	1.110	2.345
86718.750	1.107	358.007
87500.000	1.120	357.235
88281.250	1.112	366.052
89062.500	1.153	357.029
89843.750	1.105	2.370
90625.000	1.116	359.550
91406.250	1.121	2.510
92187.500	1.185	3.192
92968.750	1.157	360.239
93750.000	1.048	2.775
94531.250	1.145	0.923
95312.500	1.076	361.377
96093.750	1.142	1.394
96875.000	1.125	1.753
97656.250	1.189	361.243
98437.500	1.139	1.005
99218.750	1.146	359.319

Test Parameters

Fundamental Test Frequency - 781.25Hz
Output Voltage Range - $\pm 8V$

Figure **8.7** Measured Frequency Response of Coaxial Cable
Test Waveform - Modified PRBS

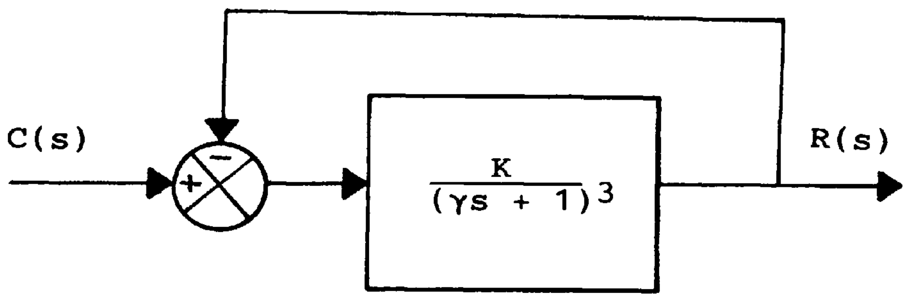
8.2 Comparative Testing

The relative performance of the two waveforms was then evaluated by using them to measure the response of a number of systems. Results from these tests were compared with those obtained when using a conventional monotonic frequency response analyser.

A control system simulator (41) was used to provide test systems. This allowed easy simulation of a number of systems and also the facility to introduce a controlled amount of non-linear behaviour.

8.2.1 Linear Systems

The linear system shown in Fig 8.8 was simulated and its response measured. Figs 8.9 and 8.10 show a graphical representation of the results obtained, while a numerical printout of the results from both analysers is contained in Appendix E.



$\gamma = 0.01s$
$K = 4$

Figure 8.8 - Linear Third Order System

Figures 8.9 and 8.10 serve to demonstrate the high degree of correlation found between those results obtained with the new instrument as compared with those obtained with the monotonic analyser - The time associated with each test over the frequency range shown is outlined in the following table.

Technique	Test Time	No of Estimates	Time/estimate
Monotonic	224s	100	2.24
PRBS	4s	128	0.031
Prime	4s	20	0.2

Note a 2s settling time was applied to all tests.

It can therefore be clearly seen that the composite test frequency technique provides a greatly improved measurement time over the monotonic approach.

It is important to note that the deviation (with respect to the monotonic analyser) in the results obtained over the lower harmonic test frequencies using the composite waveforms may be directly attributed to the effects of electrical noise identified in Section 8.1. As the frequency of the test harmonics increases however a slight deterioration in the accuracy of the results obtained using the new analyser may be observed. In particular the difference in the system magnitude response results obtained by the two analysers at these frequencies is too large to be attributed to electrical noise alone (up to 17% at the 73 test harmonic).

The dynamics of the system under test act as a low pass filter and effectively band-limits the composite test signals removing any high frequency harmonics introduced into the regenerated signal by the original sampling process. This in turn introduces errors into the calculated spectrum of the captured signal since the formula derived in Chapter4 for the system response calculation assumes no attenuation at the aliasing frequencies. The result of this is the observed decrease in calculated gain at the higher end of the test frequency band and an associated deterioration in the phase accuracy.

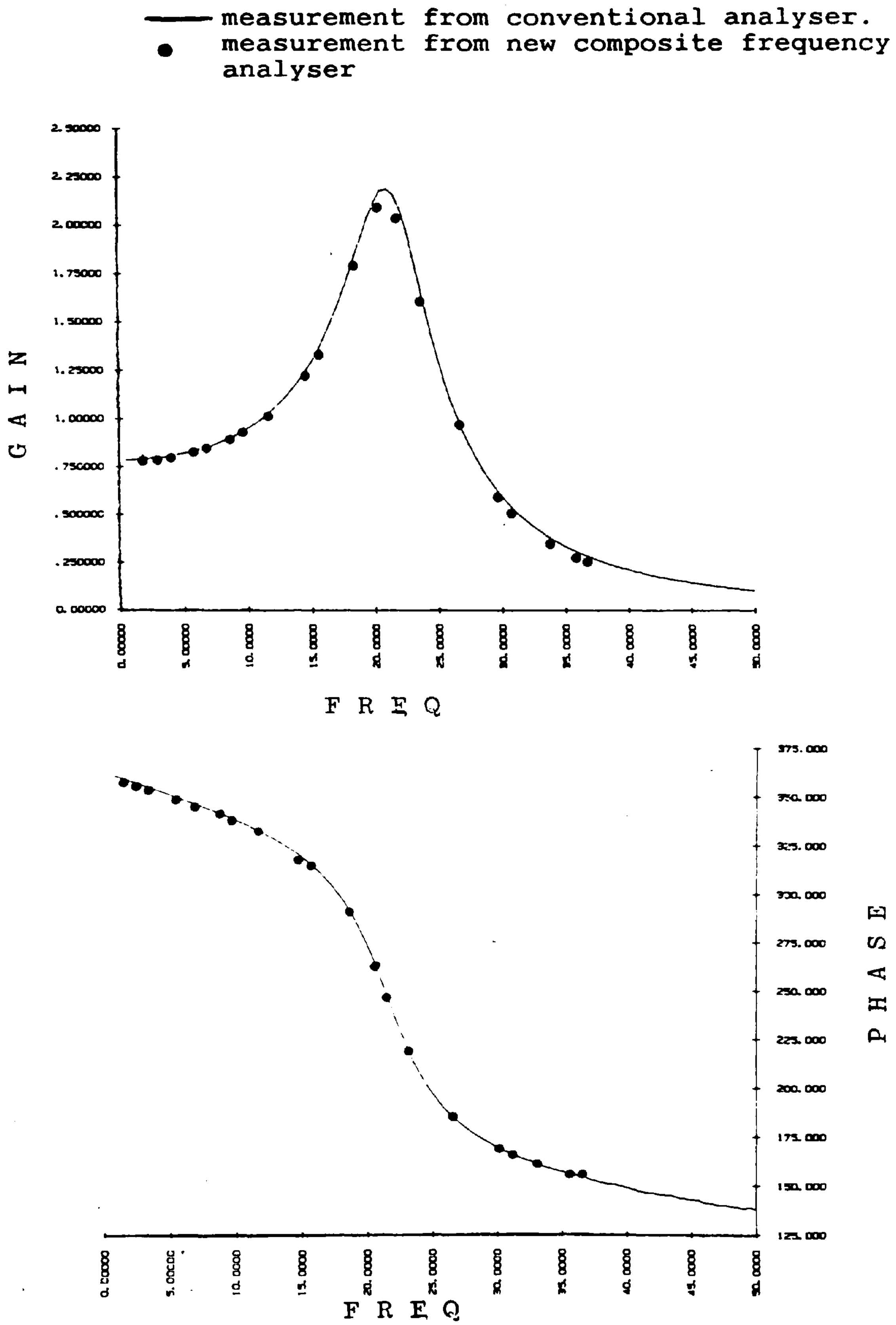


Figure 8.9 Frequency Response of Linear Third Order System measured using the Prime Composite Test Waveform

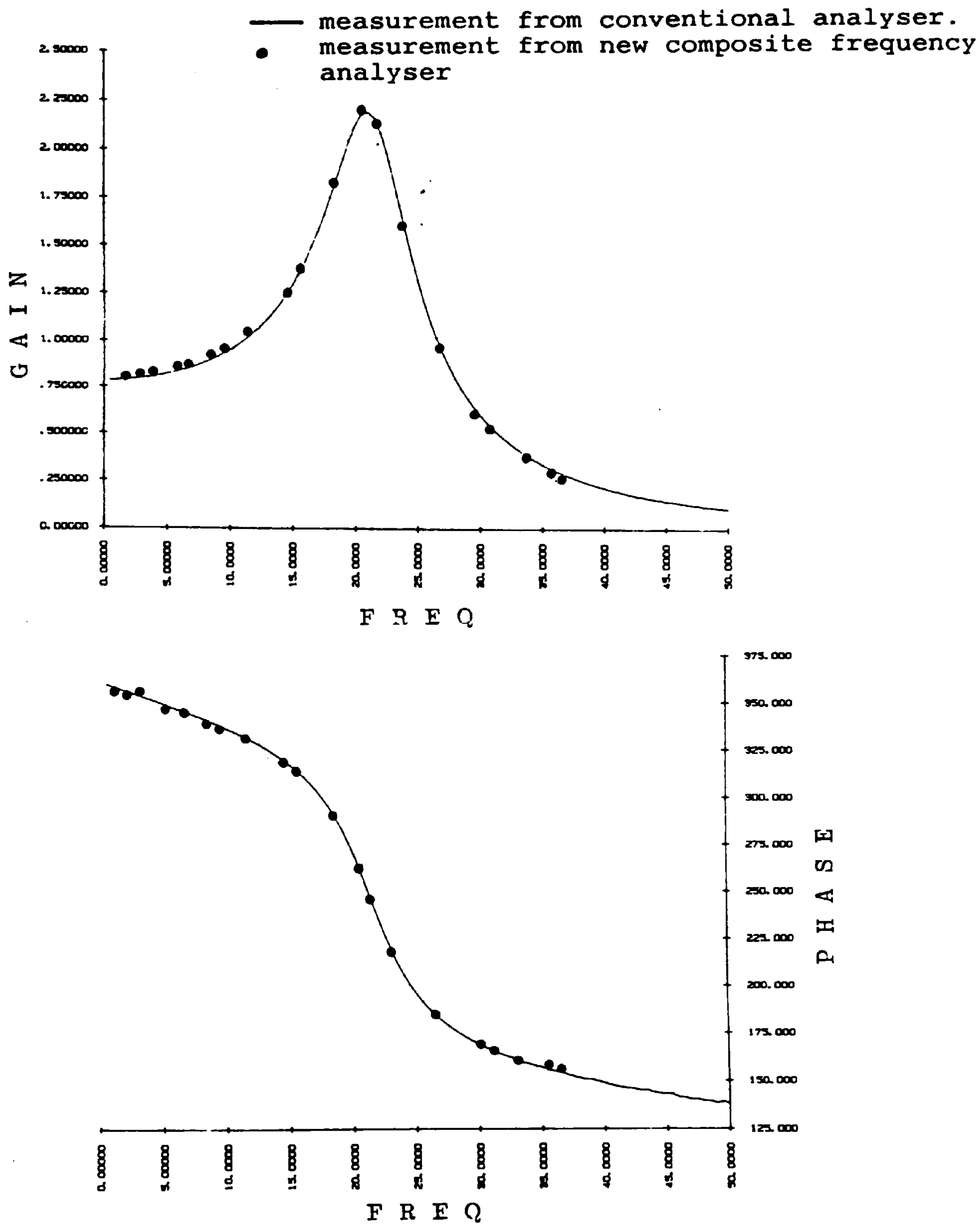


Figure 8.10 Frequency response of linear third order system measured using the PRBS signal

8.2.2 Linear System with Saturation

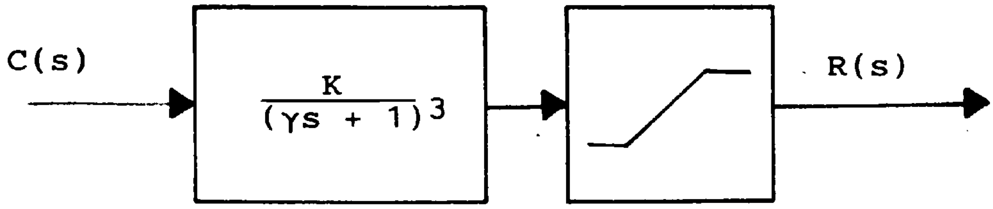
In order to compare the signals considered in this research programme and also further assess the suitability of the instrument for identifying systems with various degrees of non-linearity, a third order system which allowed degrees of non-linearities to be introduced was tested. The system was arranged both in open-loop and closed-loop configuration as indicated in figures 8.11(a) and 8.11(b). As can be seen the non-linearity is of the saturation kind, which will generate harmonics at odd multiples of the component test frequencies. The amplitude of these harmonics will be determined by the degree of non-linearity introduced. In order to quantify the degree of system non-linearity the following saturation figure is defined.

$$\text{Saturation Figure } (\beta) = (A_m - A_s)/A_m \times 100\%$$

Where
A_m is the peak amplitude of the input test signal
and
A_s is the voltage amplitude at which saturation occurs

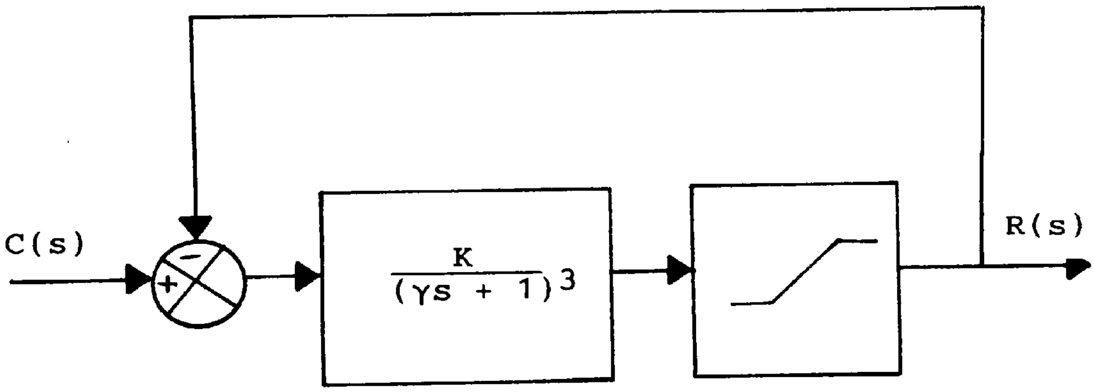
From the above definition it can be seen that there will be an increase in the degree of saturation as β increases.

Clearly , in the case of the closed-loop system, not only will there be harmonics generated at odd component frequencies due to saturation, but there will also be sum and difference frequencies produced as the result of negative feedback. One would expect therefore, the spectral estimates to have a greater harmonic content over



$K = 1,\ \gamma = 0.01s$

Figure 8.11(a) Open-loop Third Order System with Saturation



$K = 1,\ \gamma = 0.01s$

Figure 8.11(b) Closed-loop Third Order System with Saturation the frequencies of test. However, these effects may be partially moderated as negative feedback will tend to linearise the effect of the non-linearity. A further expectation would be that the effective forward path gain will decrease with an increase in saturation and this would reflect itself in reducing the magnitude plots for increasing values of β.

In the case of monotonic testing, the prediction of reduced gain can be accurately estimated using the describing function. However, the phenomonon is very much dependant on the amplitude probability distribution of the test signal, and clearly in the case of the signals considered in this investigation the reduction in gain for different β values will be different from those obtained using a single sinusoid test signal.

The results obtained from the instrument in the case of the system configurations shown in Figure 8.11, which includes a saturation non-linearity are given in Figures 8.12, 8.13, 8.14 and 8.15.

Figures 8.12, and 8.13 give the results for the open-loop system, with different levels of saturation. It can be seen from these results, that the trends already discussed in this section are readily observable :

(i) With the introduction of saturation, there is a reduction in gain.

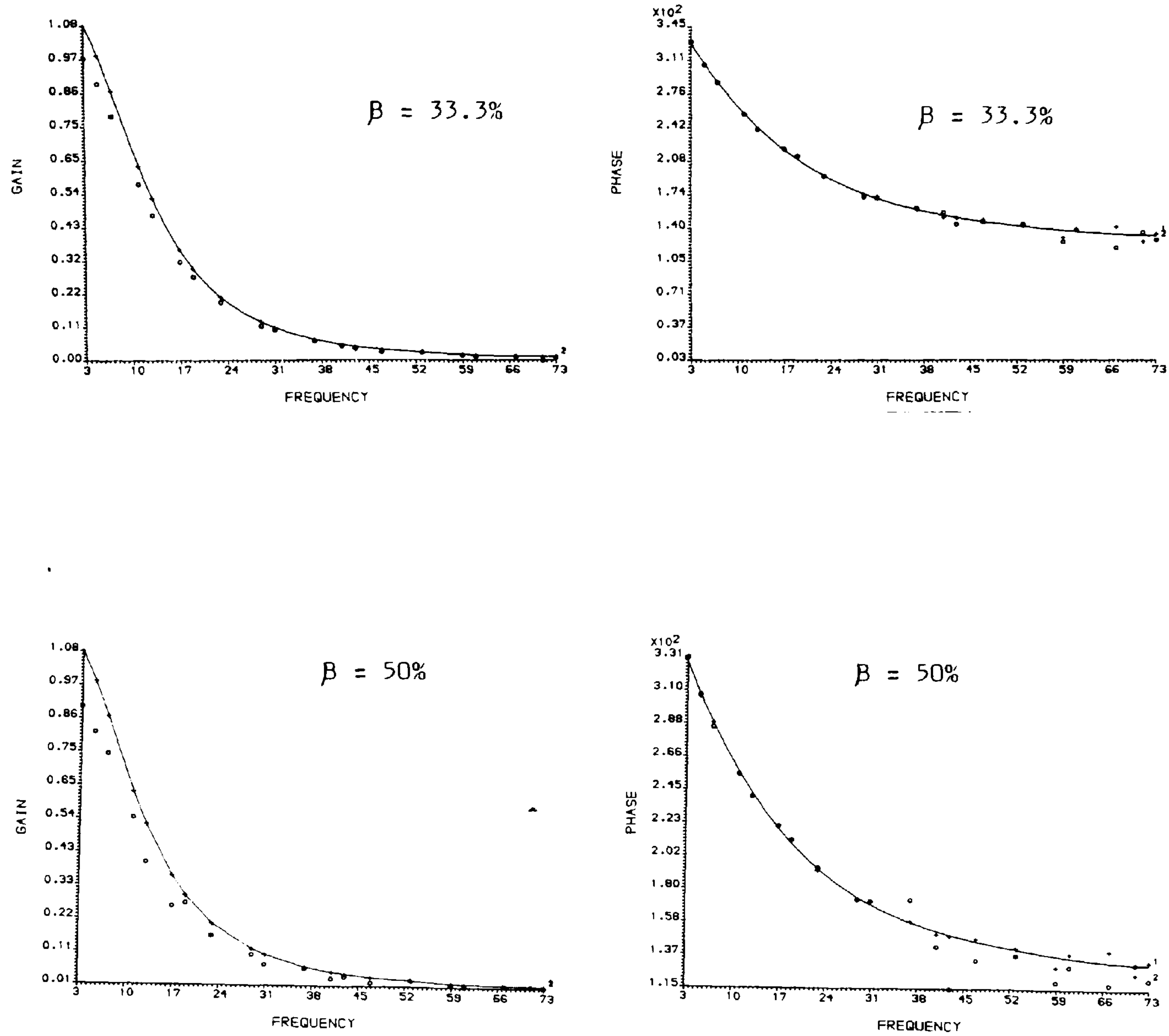


Figure 8.12 Frequency Response of Open-loop System with Saturation Non-linearity using Prime Test Signal

—+—+—+—+— Linear Response

o Measured using Prime Signal

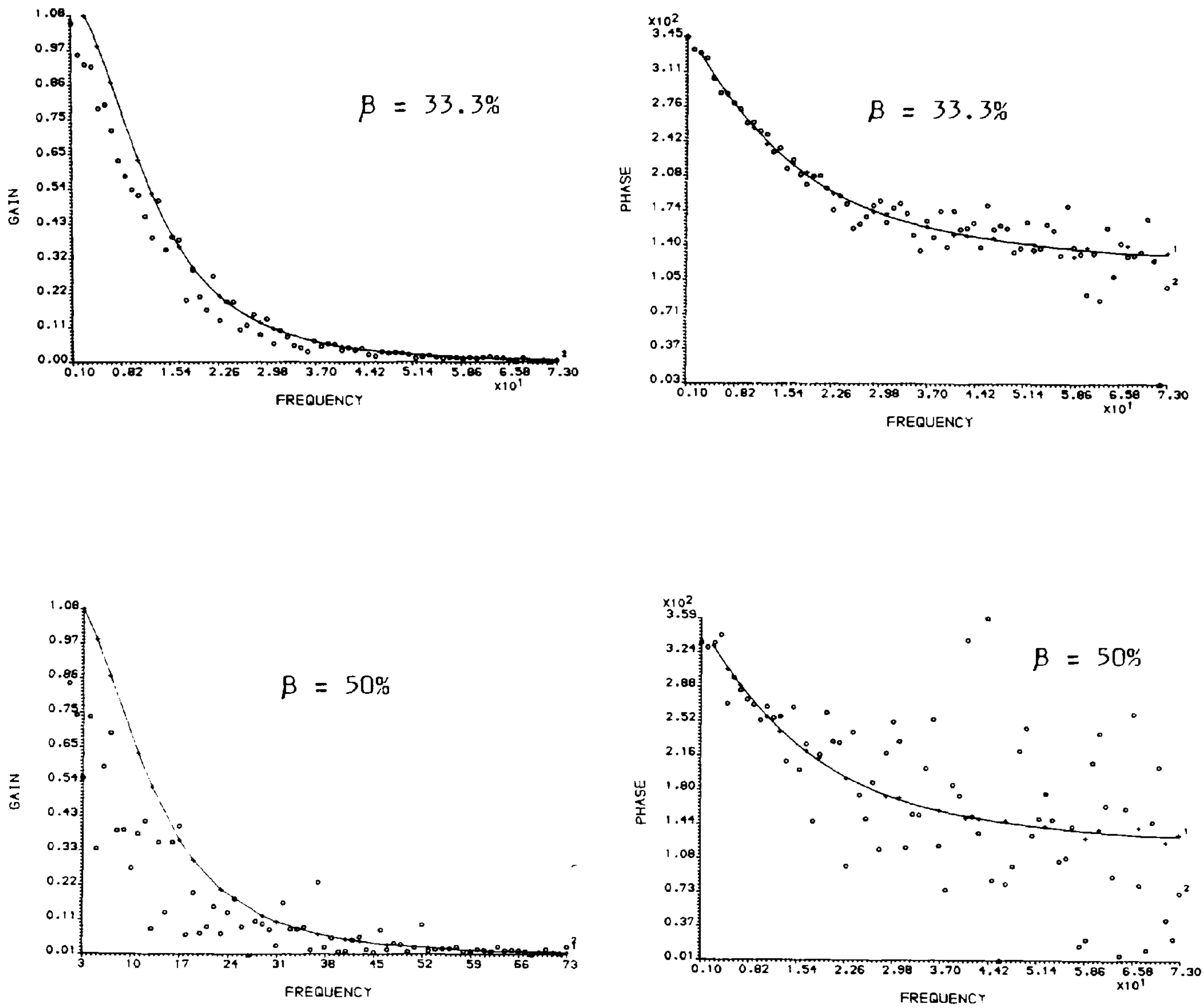


Figure 8.13 Frequency Response of Open-loop System with Saturation Non-linearity using Modified PRBS Signal

—+—+—+—+—+— Linear Response

o Measured using Modified PRBS Signal

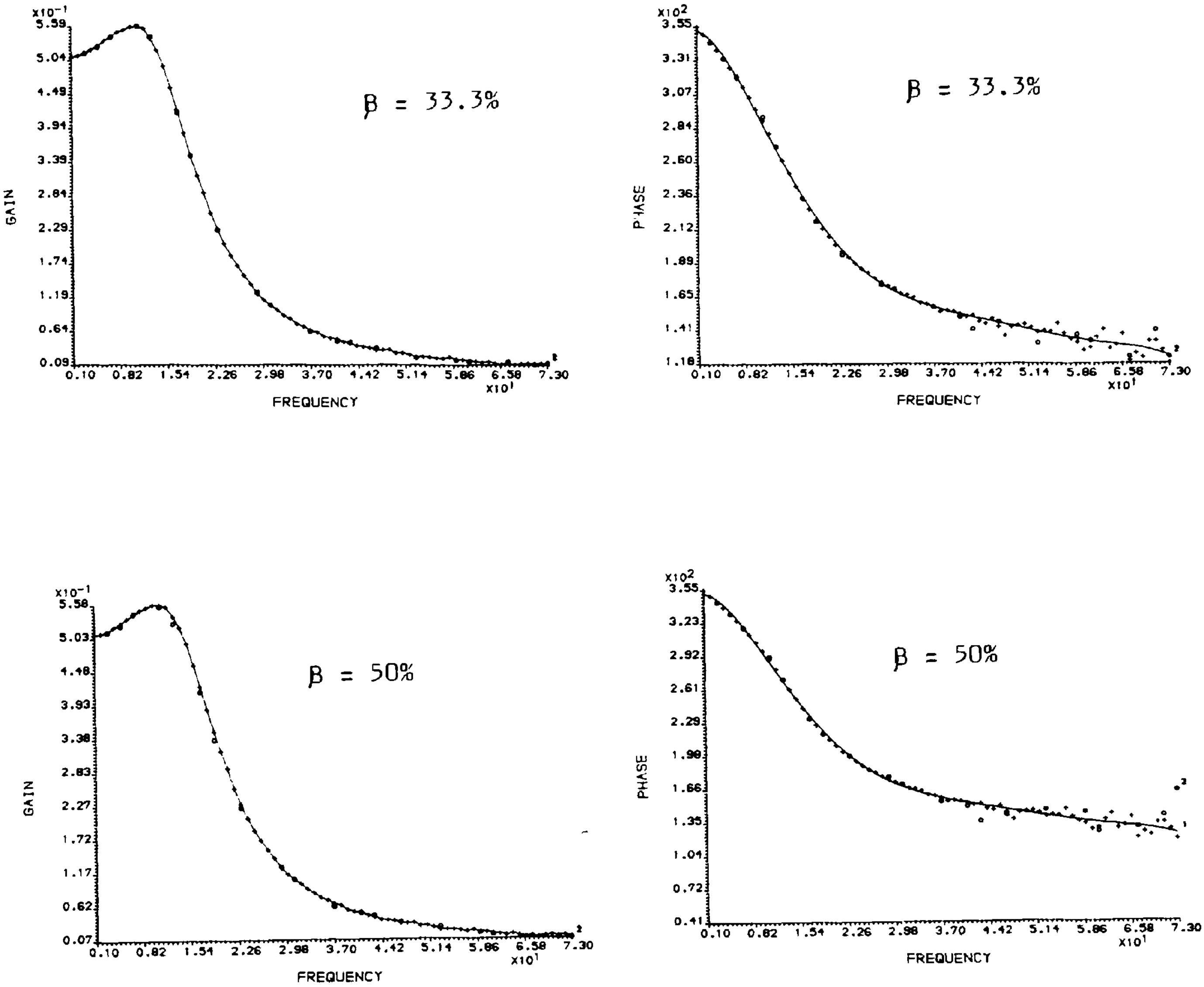


Figure 8.14 Frequency Response of Closed-loop System with Saturation Non-linearity using Prime Signal

—+—+—+—+— Linear Response

o Measured using Prime Signal

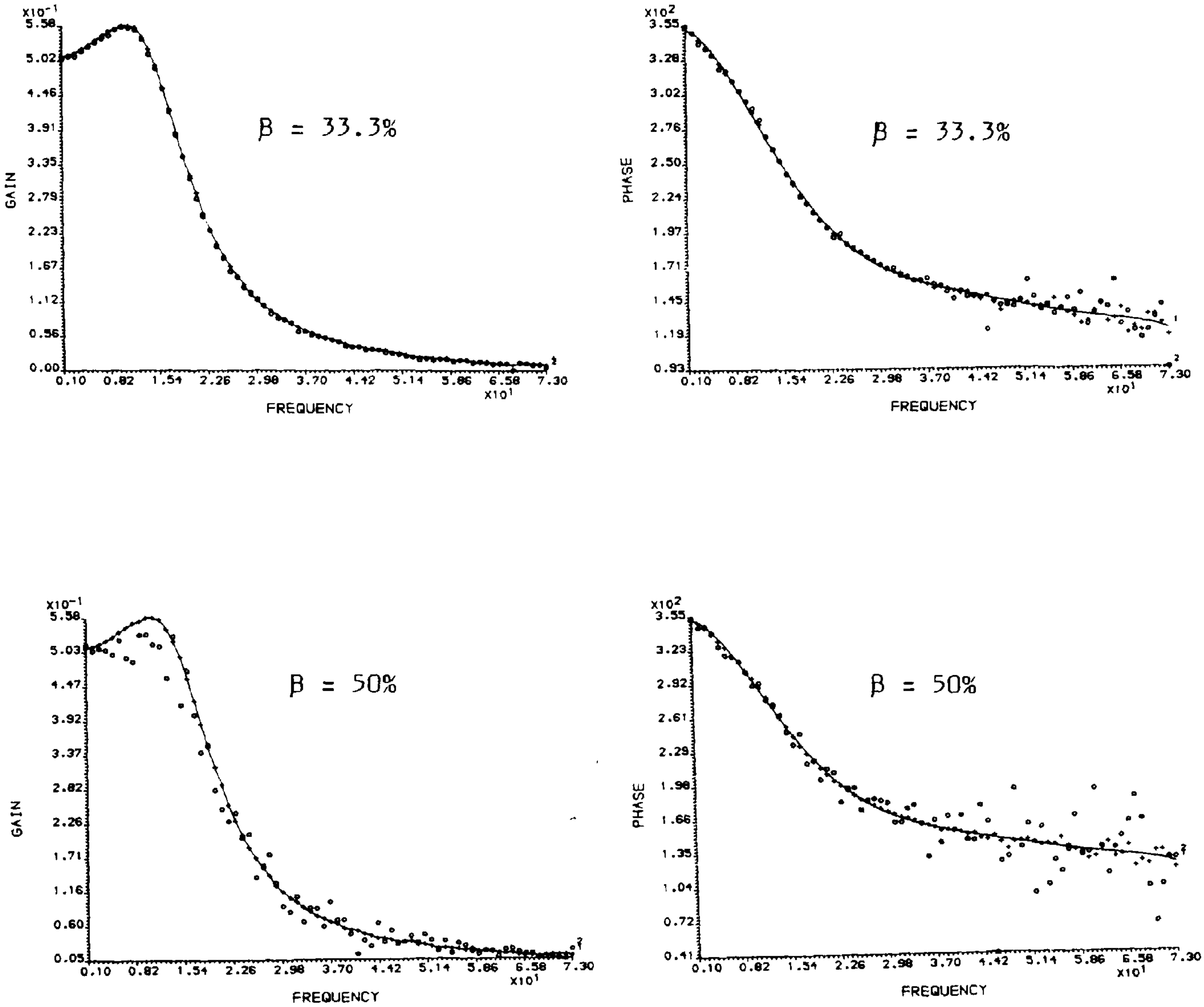


Figure 8.15 Frequency Response of Closed-loop System with Saturation Non-linearity using Modified PRBS Signal

—+—+—+—+— Linear Response

o Measured using Modified PRBS Signal

(ii) The prime sinusoid gives results that have a substantially reduced level of scatter when compared with those obtained using the Modified PRBS waveform.

(iii) The degree of degradation of the spectral estimates is directly related to the amount of saturation introduced, i.e. as β increases, so the degree of scatter increases. However, in the case of the prime sinusoid this degradation is minimal compared to the Modified PRBS, for the reasons that have been discussed previously.

(iv) The phase estimates have a great susceptibility to scatter, however, this can be partly attributed to a lower signal/noise ratio at the higher test frequencies.

The closed loop results are shown in figures 8.14 and 8.15. Again the trends are consistent with those measured in the open loop case, with the additional observations.

(i) The result of negative feedback serves to linearise the behaviour of the system, so that the results obtained using prime when the saturation level is 33.3% (β = 33.3%) gives estimates that are comparable with a linear system (β = 0).

(ii) As β increases, there is an observable reduction in the gain of the system. However, this reduction in relative terms is less than that observed in the open-loop case.

(iii) In all cases the prime sinusoid gives estimates with less scatter for both magnitude and phase measurement.

(iv) The estimates obtained for the closed loop system for both signals show less scatter than the corresponding open loop case.

This chapter has quantified the errors obtained using the new instrument, and has demonstrated that it can provide reliable and accurate estimates when measuring the frequency responses of dynamic systems. It has also been shown, that the prime sinusoid signal has properties which make it ideally suitable for performing composite frequency tests on non-linear systems.

One of the novel features of the analyser has been the ommission of an anti-aliasing filter. It has been shown that in the main, the results measured show strong agreement with those obtained from a monotonic analyser. When testing systems which exhibit low pass filter characteristics however, a degree of discrepancy in the measured results has been observed which may be attributed to a limitation of the frequency response calculation derived in Chapter 4. In response to this a new anti-aliasing compensation algorithm has been developed which overcomes this limitation. This work is presented in the next chapter.

CHAPTER 9

A NEW ANTI-ALIASING COMPENSATION ALGORITHM

The theoretical basis for the instrument's frequency response measurement strategy has been presented in detail in Chapter 4. A formula was developed for calculating the test system response which compensated for the effects of aliasing on measurements taken. This formula was based on the assumption that all high frequency harmonics generated by the waveform regeneration procedure were unaffected by the dynamics of the system under test.

It has subsequently been shown in Chapter 8, that the use of the formula results in system measurements which closely correspond to those obtained using a conventional monotonic analyser. In the case where the system possessed low pass filter characteristics, however, the calculated magnitude response of the system at the higher test frequencies deviated from the actual due to the effects of aliasing.

The aim of this chapter is to examine the aliasing problem further and to develop a new anti-aliasing compensation algorithm which may be used within the instrument to provide aliasing compensation for all system measurement regardless of the system dynamics, thereby eliminating the need for an elaborate, variable cut-off frequency, anti-aliasing filter.

Due to the differing spectral nature of the two composite waveforms employed within the instrument the new algorithm will be developed independently for each waveform.

9.1 The Prime Composite Waveform Algorithm

As discussed in Chapter 4 the system response at any harmonic test frequency (m/NT) is given by equation 4.34

$$G\left(\frac{m}{NT}\right) = \frac{U\left(\frac{m}{NT}\right) - \frac{1}{T^2} \sum_{\substack{k=-\infty \\ k<>0}}^{\infty} S\left(\frac{m}{NT}\right) H\left(\frac{kN+m}{NT}\right) G\left(\frac{kN+m}{NT}\right)}{\frac{1}{T^2} S\left(\frac{m}{NT}\right) H\left(\frac{m}{NT}\right)} \qquad 9.1$$

In order to accurately calculate the system response the right hand side of the above equation must be evaluated.

U(m/NT) is the spectral estimate of the sampled system output as calculated by the DFT and so is therefore known S(m/NT) is the discrete spectrum of the ROMed waveform and is known for all values of m

$H\left(\frac{m}{NT}\right), H\left(\frac{kN+m}{NT}\right)$ is the transfer function of the zero order hold DAC and is given by equation 4.10

$$H\left(\frac{kN+m}{NT}\right) = \frac{T \sin\left(2\pi(T/2)\ \frac{kN+m}{NT}\right)}{2\pi(T/2)\ \frac{kN+m}{NT}}$$

ie

$$H\left(\frac{kN+m}{NT}\right) = \frac{T \sin\left(\frac{kN+m}{NT}\right)\pi}{\left(\frac{kN+m}{NT}\right)\pi} \qquad 9.2$$

and thus $H\left(\frac{m}{NT}\right)$, $H\left(\frac{kN+m}{NT}\right)$ is known for all values of m and k.

Therefore in order to accurately evaluate the system response at any frequency (m/NT) only values for G(kN+m)/NT $k = \pm 1, \pm 2 \ldots$ must be obtained.

That is the transfer function of the system at the higher "aliasing" frequencies G(kN+m)/NT $k = \pm 1, \pm 2$ .. must be measured.

In theory this represents the calculation of an infinite number of terms in order to obtain one spectral estimate. However, in practice only a small number of terms need be evaluated in order to obtain a high degree of accuracy, as is shown in the following section.

9.1.1 Limiting the number of calculated terms

To consider the error introduced in calculating G(m/NT) from equation 9.1 when only selecting values of k in the range $\pm$kmax, the difference (Diff) between the actual and the calculated value must be evaluated

$$\mathrm{Diff} = \frac{U\left(\frac{m}{NT}\right) - \frac{1}{T^2} \sum\limits_{\substack{k=-\infty \\ k<>0}}^{\infty} S\left(\frac{m}{NT}\right) H\left(\frac{kN+m}{NT}\right) G\left(\frac{kN+m}{NT}\right)}{\frac{1}{T^2} S\left(\frac{m}{NT}\right) H\left(\frac{m}{NT}\right)}$$

$$- \frac{U\left(\frac{m}{NT}\right) - \frac{1}{T^2} \sum\limits_{\substack{k=-kmax \\ k<>0}}^{kmax} S\left(\frac{m}{NT}\right) H\left(\frac{kN+m}{NT}\right) G\left(\frac{kN+m}{NT}\right)}{\frac{1}{T^2} S\left(\frac{m}{NT}\right) H\left(\frac{m}{NT}\right)} \qquad 9.3$$

In an attempt to quantify this error for a worse case situation this difference may be evaluated for an all pass filter exhibiting no dominant lag. (The existence of such a lag would have the effect of making high value terms of k negligible) That is consider $G(f) = 1\angle\Theta$ for all f.

It is also important to note that the magnitude of the aliasing terms are affected by the chosen value of m, i.e. the frequency at which the response is to be measured.

From equation 4.10 it can be seen that the effect of aliasing can increase with m since $H((kN+m)/NT)$ increases with m in the range m = 3.....73. The largest value of m in the composite test signal is 73 so this must be chosen for a worse case evaluation.

Noting from equation 4.2 that $S(m/NT) = 1\angle\Theta$ then the difference equation 9.3 in this worst case scenario may be rewritten.

$$\text{Diff} = \frac{\left[U\left(\frac{73}{NT}\right) - \frac{1}{T^2}\sum_{\substack{k=-\infty \\ k<>0}}^{\infty} H\left(\frac{kN+73}{NT}\right)\right] - \left[U\left(\frac{73}{NT}\right) - \frac{1}{T^2}\sum_{\substack{k=-kmax \\ k<>0}}^{kmax} H\left(\frac{kN+73}{NT}\right)\right]}{\frac{1}{T^2}\, H\left(\frac{73}{NT}\right)} \qquad 9.4$$

That is the difference in considering all aliasing terms and only those for -kmax < k >+kmax is given by

$$\text{Diff} = \frac{\sum_{\substack{k=-\infty \\ k<>0}}^{\infty} H\left(\frac{kN+73}{NT}\right)}{H\left(\frac{73}{NT}\right)} - \frac{\sum_{\substack{k=-kmax \\ k<>0}}^{kmax} H\left(\frac{kN+73}{NT}\right)}{H\left(\frac{73}{NT}\right)} \qquad 9.5$$

In order to evaluate the above equation it should be noted from equation 4.10

$$\sum_{\substack{k=-\infty \\ k<>0}}^{\infty} H\left(\frac{kN+73}{NT}\right) = \sum_{\substack{k=-\infty \\ k<>0}}^{\infty} T \frac{\sin\left(\frac{kN+73}{N}\right)\pi}{\left(\frac{kN+73}{N}\right)\pi} \qquad 9.6$$

The above term is a convergent series where

$$\sum_{\substack{k=-\infty \\ k<>0}}^{k=\infty} T \frac{\sin \left(\frac{kN+73}{N}\right)\pi}{\left(\frac{kN+73}{N}\right)\pi} = 0.1284905\ T \qquad 9.7$$

so that

$$\text{Diff} = \frac{0.01284905T - \sum_{\substack{k=-\max \\ k<>0}}^{k\max} T \frac{\sin \left(\frac{kN+73}{N}\right) \pi}{\left(\frac{kN+73}{N}\right) \pi}}{T \frac{\sin \left(\frac{73\pi}{N}\right)}{\frac{73\pi}{N}}} \qquad 9.8$$

therefore

$$\text{Diff} = 0.83 \left[0.1284905 - \sum_{\substack{k=-k\max \\ k<>0}}^{k\max} \frac{\sin \left(\frac{kN+73}{N}\right) \pi}{\left(\frac{kN+73}{N}\right) \pi} \right] \qquad 9.9$$

A modula 2 program has been written to evaluate this difference term. A listing of the program is given in figure 9.1 and a printout of the results obtained is given in figure 9.2.

From the results presented it can be seen that in this "worst case" situation by considering only terms where $|k|<3$ then the error introduced into the antialiasing algorithm is less than -42dB.

Furthermore by increasing the number of terms such that $|k|<20$ then the error falls to less than - 72dB.

9.1.2 An algorithm for calculating the system response from measured data

As discussed in the previous section when calculating the transfer function of the system for m = 3, 5 ....73, the response of the system at the higher frequencies (kN+m)/NT, k = ± 1, ± 2 ..... must be known. It has been shown, however, that values of G(kN+m/NT) need only be known for $|k|<20$ in order to accurately calculate the system response at G(m/NT). The response of the system at these higher frequencies can be obtained by a single further frequency response measurement where the fundamental frequency of the test signal is set to 73/NT since the response of the system is then measured at frequencies up to

$$\frac{73 \times 73}{NT} = \frac{5329}{NT}$$

whereas the system response is only required up to

$$\frac{20 \times 256 + 73}{NT} = \frac{5193}{NT}$$

However, this additional frequency response measurement may in turn also be corrupted by the effect of aliasing and consequently yet another frequency response measurement may be required in order to correct for aliasing effects. This problem continues until a set of measurements can be made

```
MODULE Terms2;
IMPORT IO,MATHLIB;
CONST n = 128.0 ; N=256.0; pi=3.141592654;
VAR r :INTEGER;
    Term1,Term2,Ang1,Ang2,
    Result,ConVal,HarmAng,HarmVal,InvHarmVal : LONGREAL;
BEGIN
    Result := 0.0;
    HarmAng := (LONGREAL(n)*pi/N);
    HarmVal := MATHLIB.Sin(HarmAng)/HarmAng;
    InvHarmVal := 1.0 / HarmVal;
    ConVal := 1.0 - HarmVal;
    IO.WrLngReal(ConVal,8,15);
    IO.WrLn;
    FOR r := 1 TO 40 DO
        Ang1 := (LONGREAL(r)*N+n)*pi/N;
        Ang2 := (LONGREAL(r)*N-n)*pi/N;
        Term1 := MATHLIB.Sin(Ang1)/Ang1;
        Term2 := MATHLIB.Sin(Ang2)/Ang2;
        Result := Result+Term1+Term2;
        IO.WrLngReal(Result,8,15);
        IO.WrLngReal(InvHarmVal*(ConVal-Result),8,15);
        IO.WrLngReal(20.0*MATHLIB.Log10(ABS(InvHarmVal*(ConVal
        -Result))),8,15);
        IO.WrInt(r,15);
        IO.WrLn;
    END;
END Terms2.
```

Figure 9.1

Modula-2 Program Terms to calculate the error introduced by limiting the number of G(kn+m/N) terms in a worst case situation

$$\sum_{\substack{k=-kmax \\ k<>0}}^{kmax} \frac{\sin \frac{(kN+73)\pi}{N}}{\frac{(kN+73)\pi}{N}}$$

$\sum$ (as above)	Diff	$20\log_{10}(\mathrm{Diff})$	
1.5427688E-1	-2.9588171E-2	-3.0577638E+1	1
1.1810863E-1	1.1912520E-2	-3.8479927E+1	2
1.3400021E-1	-6.3220241E-3	-4.3982877E+1	3
1.2509671E-1	3.8941565E-3	-4.8191732E+1	4
1.3078449E-1	-2.6321976E-3	-5.1593630E+1	5
1.2683858E-1	1.8954783E-3	-5.4445624E+1	6
1.2973588E-1	-1.4289807E-3	-5.6899473E+1	7
1.2751850E-1	1.1153169E-3	-5.9052034E+1	8
1.2927004E-1	-8.9445569E-4	-6.0968823E+1	9
1.2785156E-1	7.3314933E-4	-6.2696151E+1	10
1.2902369E-1	-6.1178879E-4	-6.4267970E+1	11
1.2803888E-1	5.1821143E-4	-6.5709860E+1	12
1.2887794E-1	-4.4454850E-4	-6.7041617E+1	13
1.2815451E-1	3.8553126E-4	-6.8278808E+1	14
1.2878466E-1	-3.3752174E-4	-6.9433965E+1	15
1.2823085E-1	2.9794622E-4	-7.0517242E+1	16
1.2872141E-1	-2.6493921E-4	-7.1537075E+1	17
1.2828385E-1	2.3712541E-4	-7.2500438E+1	18
1.2867655E-1	-2.1346945E-4	-7.3413285E+1	19
1.2832215E-1	1.9318346E-4	-7.4280601E+1	20
1.2864359E-1	-1.7565573E-4	-7.5106754E+1	21
1.2835071E-1	1.6040917E-4	-7.5895416E+1	22
1.2861868E-1	-1.4706357E-4	-7.6649898E+1	23
1.2837258E-1	1.3531669E-4	-7.7372973E+1	24
1.2859938E-1	-1.2492206E-4	-7.8067217E+1	25
1.2838969E-1	1.1568091E-4	-7.8734766E+1	26
1.2858413E-1	-1.0742770E-4	-7.9377675E+1	27
1.2840333E-1	1.0002753E-4	-7.9997609E+1	28
1.2857188E-1	-9.3365746E-5	-8.0596249E+1	29
1.2841438E-1	8.7348365E-5	-8.1174904E+1	30
1.2856188E-1	-8.1893837E-5	-8.1734976E+1	31
1.2842346E-1	7.6935164E-5	-8.2277502E+1	32
1.2855362E-1	-7.2412966E-5	-8.2803673E+1	33
1.2843100E-1	6.8278554E-5	-8.3314314E+1	34
1.2854671E-1	-6.4487764E-5	-8.3810454E+1	35
1.2843734E-1	6.1004624E-5	-8.4292745E+1	36
1.2854088E-1	-5.7795651E-5	-8.4762097E+1	37
1.2844272E-1	5.4833893E-5	-8.5219018E+1	38
1.2853591E-1	-5.2093513E-5	-8.5664327E+1	39
1.2844732E-1	4.9554074E-5	-8.6098413E+1	40

Figure 9.2 Table showing the errors introduced by varying the number of $G\left(\frac{kN + m}{N}\right)$ terms a worse case scenario – Prime Composite Waveform

which can be guaranteed to be negligibly affected by aliasing - for example a measurement just below the maximum frequency range of the instrument. [Here it is assumed that any system tested will have no significant response at any frequency above the maximum range of the instrument. If this cannot be guaranteed a simple fixed cut-off frequency filter may be employed.]

The maximum number of repeat tests required is in practice very small since using the Prime Composite Waveform the response of the system is known up to the 73 harmonic of the fundamental test frequency. Therefore two repeat test would guarantee an accurate frequency response analysis from 50kHz down to 0.12Hz. (Three repeat tests gives 50kHz down to 1.7 x 10^{-3}Hz)

The above procedure therefore represents a algorithm by which the response of a system may be measured without the need for elaborate anti-aliasing filters.

As discussed above a repeat test is necessary in order to evaluate the system response at the significant aliasing frequencies.

This repeat measurement gives the system response at 20 frequencies in the range 73/NT to 5329/NT. As explained in Section 9.1.1 however the response of the system needs to be known at 20 additional spectral points for each calculated frequency in order to obtain an accuracy of 72dB. That is the system response ideally needs to be known at 20 x 20 =

400 points in the range (256-73)/NT to 5120/NT i.e. in order to obtain values for all the terms in the compensation algorithm the response of the system must be interpolated between measurement points.

The interpolation algorithm adopted in the instrument is a very simple one. When the system response at a given frequency is required for the antialiasing algorithm, and this frequency lies between two known measured points, the value adopted is that of the system response at the nearest measured spectral point.

9.1.3 Results obtained using the new algorithm

In order to evaluate the algorithm it has been incorporated within the FFT based analyser. Since the algorithm requires no additional hardware and also puts no real time constraint on the analyser, the algorithm has been implemented in software on the general purpose CPU of the analyser.
The implementation in the analyser only incorporates the first eight aliasing terms giving a maximum possible error of <-48dB. See figure 9.2

In order to demonstrate the effectiveness of the anti-aliasing algorithm the analyser was then used to measure the response of two different 'real' systems.

System 1 This was an all pass filter implemented by a one metre length of co-axial cable. This was chosen since no attenuation of the aliasing frequencies will occur due to any filtering effect of the system dynamics. This system tested both with and without the algorithm.

System 2 A second order system with two poles at 300 Hz. This was chosen to given a significant attenuation at the aliasing frequencies. The response of the system was measured in the range 28.146Hz to 684.886Hz.

The frequency response obtained for the first system was compared with that theoretically expected (gain = 1 phase $\angle$ 0 for all f) while that obtained for the second system was compared with those obtained when using a commercial, monotonic frequency response analyser applied to the same system.

The results of these tests are presented in Figures 9.3 and 9.4.

Figure 9.3a shows the calculated frequency response estimates of the all pass filter using the prime composite test waveform without applying the antialiasing algorithm. As expected the higher test frequencies are significantly affected by aliasing effects e.g. the spectral estimate at the 73rd harmonic test frequency has a 14% error in the magnitude of the calculated system transfer function. [This figure may be theoretically predicted from equation 4.34 where m = 73 and G(kn+m/NT) = 1 $\angle$ 0]

Figure 9.3b on the other hand show the result obtained from the same test but with the antialiasing correction algorithm

Figure 9.3a **All pass filter - no aliasing correction algorithm applied**

FREQ(Hz)	MAGN(T.F.)	FH(T.F.)
0.000	1.108	0.000
28.146	0.994	0.121
46.910	0.992	0.158
65.674	0.994	0.059
103.202	0.995	359.996
121.966	0.997	359.947
159.494	0.998	0.025
178.258	1.003	0.041
215.786	1.006	359.998
272.078	1.018	359.980
290.842	1.020	0.145
347.134	1.029	0.039
384.662	1.039	359.979
403.426	1.044	359.952
440.954	1.053	0.053
497.246	1.069	359.890
553.538	1.087	359.839
572.302	1.096	0.016
628.594	1.119	359.971
666.122	1.132	0.018
684.886	1.141	359.892

Press Return to Continue

Figure 9.3b **All pass filter - with aliasing correction algorithm applied**

FREQ(Hz)	MAGN(T.F.)	FH(T.F.)
0.000	1.113	0.000
28.146	0.994	359.906
46.910	0.991	0.084
65.674	0.991	359.863
103.202	0.995	0.084
121.966	0.992	359.809
159.494	0.990	0.087
178.258	0.996	359.874
215.786	0.992	359.978
272.078	0.994	359.546
290.842	0.991	0.119
347.134	0.991	0.313
384.662	0.994	0.000
403.426	0.996	0.282
440.954	0.993	0.264
497.246	0.990	0.286
553.538	0.993	0.458
572.302	0.992	0.674
628.594	0.993	0.212
666.122	0.991	0.637
684.886	0.987	0.764

Press Return to Continue

Figure 9.4a Two cascaded 300Hz low pass filters

- Measurement Instrument Thorn EMI Monotonic F.R.A.

28.1460	1.00420	345.1
46.910	1.00412	335.2
65.674	1.00216	325.0
103.202	0.99110	303.5
121.966	0.98150	292.8
159.494	0.93455	270.3
178.258	0.90380	258.5
215.786	0.80886	234.6
272.078	0.62132	200.3
290.842	0.55822	189.7
347.134	0.38053	161.5
384.662	0.29024	145.8
403.426	0.25241	136.7
440.954	0.19076	128.0
497.246	0.13252	110.3
553.538	0.08821	98.67
572.302	0.07670	95.28
628.594	0.05368	85.84
666.122	0.04294	80.85
684.886	0.03861	78.59

Figure 9.4b Two cascaded 300Hz low pass filters

- Measurement Instrument Prime Composite F.R.A.
anti aliasing correction algorithm applied

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
0.000	1.109	0.000
28.146	0.991	344.972
46.910	0.989	335.406
65.674	0.987	327.004
103.202	0.979	304.084
121.966	0.967	291.590
159.494	0.923	271.014
178.258	0.892	259.273
215.786	0.806	234.229
272.078	0.620	198.128
290.842	0.558	190.910
347.134	0.384	162.420
384.662	0.295	147.785
403.426	0.254	135.912
440.954	0.191	126.972
497.246	0.130	110.817
553.538	0.088	98.342
572.302	0.079	94.770
628.594	0.057	86.626
666.122	0.044	83.746
684.886	0.039	79.457

Press Return to Continue

applied. Here the obvious effects of aliasing have been compensated for - the measurement errors at the higher frequencies have been significantly reduced. The remaining small variation in the magnitude of the calculated system frequency response can be primarily attributed to electrical noise in the measurement channel as was identified in Chapter 8.

Figure 9.4 shows a comparative test between the Prime Composite Frequency Response analyser and a commercial monotonic frequency response analyser as applied to a linear second order system.

The transfer function obtained by the monotonic analyser can be viewed to all intent and purpose as free from any aliasing effects due to the harmonic rejection characteristic of the detection process.

As can be seen the responses obtained using the two instruments are comparable, with a slight disparity arising from the different noise characteristics of the two instruments.

The results presented demonstrate how the frequency response of a system can be accurately measured by employing a Prime Composite test waveform in conjunction with the new anti-aliasing compensation algorithm.

9.2 The Modified PRBS Waveform Algorithm

In applying the strategy outlined in the previous section to the Modified PRBS waveform the same technique as used in Section 4.2 will be employed. The algorithm will be applied to a single periodic pulse waveform and superposition will be used to develop the argument for the whole Modified PRBS signal.

Equation 4.83 shows that in order to evaluate the system response at a single test frequency an infinite number of terms must be taken into account. Since this cannot be realised in practice, the number of terms must then be limited and it is important to assess the effect of this on the accuracy of the calculated system response.

In order to evaluate the error introduced in calculating G(m/NT) from equation 4.83 when only considering values of k in the range $\pm$ kmax, the difference (Diff) between the actual and calculated values must be determined.

$$\mathrm{Diff} = \frac{U\left(\frac{m}{NT}\right) - \frac{1}{T}\sum_{\substack{k=-\infty \\ k<>0}}^{\infty} S_i\left(\frac{kN+m}{NT}\right)G\left(\frac{kN+m}{NT}\right)}{\frac{1}{T}\,S_i\left(\frac{m}{NT}\right)} - \frac{U\left(\frac{m}{NT}\right) - \frac{1}{T}\sum_{\substack{k=-kmax \\ k<>0}}^{kmax} S_i\left(\frac{kN+m}{NT}\right)G\left(\frac{kN+m}{NT}\right)}{\frac{1}{T}\,S_i\left(\frac{m}{NT}\right)} \qquad 9.10$$

Therefore

$$\text{Diff} = \frac{\sum\limits_{\substack{k=-\infty \\ k<>0}}^{\infty} S_i\left(\frac{kN+m}{NT}\right) G\left(\frac{kN+m}{NT}\right)}{S_i\left(\frac{m}{NT}\right)}$$

$$- \frac{\sum\limits_{\substack{k=-kmax \\ k<>0}}^{kmax} S_i\left(\frac{kN+m}{NT}\right) G\left(\frac{kN+m}{NT}\right)}{S_i\left(\frac{m}{NT}\right)} \qquad 9.11$$

To quantify this difference in a worse case situation, this expression may be evaluated for an all pass filter exhibiting no dominant lag.

i.e. $G(f) = 1\angle\,\Theta$ for all f

(Again the existence of such a lag would have the effect of making high value terms of k negligible.)

The value of m chosen will also effect the error introduced in the transfer function calculation. Equation 4.83 shows that the effect of aliasing can increase with m since from equation 4.53 it can be shown that the value of Si(kn+m/NT) increases with m and Si(m/NT) decreases with m in the range m = 1.... 128 (The harmonic range over which measurements are taken) This is also readily observable from the comparison of the true spectrum of the Modified PRBS signal, and the heavily aliased spectrum given in Appendix F.

In order therefore to obtain a worst case value equation 9.11 may be rewritten

$$\mathrm{DIFF} = \frac{\sum_{\substack{k=-\infty \\ k<>0}}^{\infty} S_i\left(\frac{kN+128}{NT}\right)}{S_i\left(\frac{128}{NT}\right)} - \frac{\sum_{\substack{k=-kmax \\ k<>0}}^{kmax} S_i\left(\frac{kN+128}{NT}\right)}{S_i\left(\frac{128}{NT}\right)} \qquad 9.12$$

Noting from equation 4.53

$$S_i\left(\frac{m}{NT}\right) = \frac{\frac{1}{NT} A_i \sin\left(\frac{128\pi}{N}\right) e^{-j(2\pi i128/N)}}{\left(\frac{128\pi}{N}\right)} \qquad 9.13$$

$$S_i\left(\frac{kN+128}{NT}\right) = \frac{\frac{1}{NT} A_i \sin\left(\frac{kN+128}{N}\right)\pi \; e^{-j(2\pi i128/N)}}{\left(\frac{kN+128}{N}\right)\pi} \qquad 9.14$$

$$\mathrm{Diff} = \frac{\sum_{\substack{k=-\infty \\ k<>0}}^{\infty} \frac{\sin\left(\frac{kN+128}{N}\right)\pi}{\left(\frac{kN+128}{N}\right)\pi} - \sum_{\substack{k=-kmax \\ k<>0}}^{kmax} \frac{\sin\left(\frac{kN+128}{N}\right)\pi}{\left(\frac{kN+128}{N}\right)\pi}}{\frac{\sin\left(\frac{128\pi}{N}\right)}{\left(\frac{128\pi}{N}\right)}} \qquad 9.15$$

However $\frac{\sin(128/N)\pi}{(128/N)\pi} = 0.63662$ 9.16

and the series $\sum_{\substack{k=-\infty \\ k=<>0}}^{\infty} \frac{\sin\left(\frac{kN+128}{N}\right)\pi}{\left(\frac{kN+128}{N}\right)\pi}$ converges to

$$\left(1 - \frac{\sin(128/N)\pi}{(128/N)\pi}\right) \text{ ie } 0.36338 \qquad 9.17$$

$$\mathrm{Diff} = \frac{1}{0.63662}\left[0.36338 - \sum_{\substack{k=-kmax \\ k=<>0}}^{kmax} \frac{\sin\left(\frac{kN+128}{N}\right)\pi}{\left(\frac{kN+128}{N}\right)\pi}\right] \qquad 9.18$$

This equation was evaluated using the programme Terms presented in the previous section and the results obtained are given in figure 9.5.

From this table it can be seen that for this worst case scenario by only considering terms where $|k| < 6$ then the error introduced into the transfer function calculation is less than -42dB.

Furthermore by increasing the number of terms such that $|k| < 32$ then the error in the calculation falls to <72dB i.e. below the theoretical resolution of the instrument.

By a similar argument to that given in the previous section for the Prime Composite waveform, it can be demonstrated that this number of terms of G(kN+m/NT) can be evaluated by a second frequency response test at the aliasing frequencies since this would cover a frequency range up to 128x128/NT or 16384/NT and the system response is only required up to 8320/NT.

The above arguments have been applied to a single pulse periodic waveform $S_i(f)$. However from equation 4.4 it is shown that the Modified PRBS signal incorporated within the frequency response analyser is a composite signal comprised of a series of such waveforms. By superposition therefore it is evident that the new antialiasing algorithm may be readily applied to the MPRBS waveform to compensate for any aliasing effects introduced by the measurement procedure.

$\sum_{k=-kmax,\ k<>0}^{kmax} \frac{\sin\left(\frac{kN+128}{N}\right)\pi}{\left(\frac{kN+128}{N}\right)\pi}$	Diff	$20\log_{10}(\text{Diff})$	
4.2441318E-1	-9.5870340E-2	-2.0366315E+1	1
3.3953054E-1	3.7462994E-2	-2.8527950E+1	2
3.7590882E-1	-1.9679863E-2	-3.4119558E+1	3
3.5569867E-1	1.2066168E-2	-3.8368612E+1	4
3.6855967E-1	-8.1358519E-3	-4.1791939E+1	5
3.5965590E-1	5.8501621E-3	-4.4656642E+1	6
3.6618533E-1	-4.4062482E-3	-4.7118621E+1	7
3.6119224E-1	3.4368890E-3	-4.9276690E+1	8
3.6513415E-1	-2.7550614E-3	-5.1197374E+1	9
3.6194308E-1	2.2574699E-3	-5.2927561E+1	10
3.6457918E-1	-1.8833168E-3	-5.4501532E+1	11
3.6236485E-1	1.5949440E-3	-5.5945091E+1	12
3.6425114E-1	-1.3680189E-3	-5.7278158E+1	13
3.6262503E-1	1.1862595E-3	-5.8516406E+1	14
3.6404132E-1	-1.0384346E-3	-5.9672417E+1	15
3.6279670E-1	9.1659960E-4	-6.0756407E+1	16
3.6389907E-1	-8.1500213E-4	-6.1776825E+1	17
3.6291588E-1	7.2939941E-4	-6.2740692E+1	18
3.6379823E-1	-6.5660197E-4	-6.3653956E+1	19
3.6300196E-1	5.9417976E-4	-6.4521643E+1	20
3.6372416E-1	-5.4025018E-4	-6.5348101E+1	21
3.6306616E-1	4.9334155E-4	-6.6137046E+1	22
3.6366816E-1	-4.5228493E-4	-6.6891758E+1	23
3.6311530E-1	4.1614755E-4	-6.7615053E+1	24
3.6362480E-1	-3.8417258E-4	-6.8309473E+1	25
3.6315375E-1	3.5574603E-4	-6.8977199E+1	26
3.6359054E-1	-3.3036032E-4	-6.9620242E+1	27
3.6318440E-1	3.0759821E-4	-7.0240324E+1	28
3.6356301E-1	-2.8710889E-4	-7.0839067E+1	29
3.6320923E-1	2.6860103E-4	-7.1417847E+1	30
3.6354054E-1	-2.5182572E-4	-7.1977998E+1	31
3.6322962E-1	2.3657477E-4	-7.2520632E+1	32
3.6352198E-1	-2.2266748E-4	-7.3046864E+1	33
3.6324657E-1	2.0995203E-4	-7.3557599E+1	34
3.6350647E-1	-1.9829455E-4	-7.4053784E+1	35
3.6326081E-1	1.8758235E-4	-7.4536160E+1	36
3.6349336E-1	-1.7771445E-4	-7.5005545E+1	37
3.6327289E-1	1.6860589E-4	-7.5462545E+1	38
3.6348220E-1	-1.6017924E-4	-7.5907875E+1	39
3.6328323E-1	1.5236959E-4	-7.6342034E+1	40

Figure 9.5 Table showing the errors introduced by varying the number of $G\left(\frac{KN + m}{N}\right)$ terms for a worse case scenario – Binary Pulse Waveform

CHAPTER 10

CONCLUSIONS AND FURTHER WORK

This thesis describes an investigation into the use of composite frequency signals for the frequency response testing of system dynamics. The design and development of a new analyser based upon composite frequency test procedures is presented.

It has been established that the use of periodic composite test signals offers significant advantages over alternative measurement waveforms in terms of reduced measurement times and ease of implementation (mechanisation).

The desirable attributes of a composite frequency waveform are:

a) A Low peak factor.

b) A relatively flat amplitude envelope in the frequency domain.

Two composite waveforms which possess both of the above attributes have been reviewed. These are:

1) A binary waveform developed from a 255 sequence length Pseudo Random Binary Signal.

2) A new optimum phase composite frequency waveform comprised of 20 primely related harmonics.

The theoretical basis for the design of a frequency response analyser incorporating the waveforms has been put forward. The test procedure identified makes use of digital techniques for the storage and generation of the composite waveforms. It has been shown, however, that the application of such techniques results in reconstituted waveforms which are non band-limited in the frequency domain. If these waveforms are applied directly to the system under test then the system output cannot be guaranteed to be digitally sampled without introducing aliasing effects into the measurement process.

The accepted solution to this problem is to use anti-aliasing filters to band-limit the reconstituted waveform before application to the system under test. The nature of composite frequency response analysers however can put severe performance requirements on the design of such filters. As a result anti-aliasing filters generally incorporate both analogue and digital filtering procedures which in turn can make these filters complex and expensive to produce.

An anti-aliasing compensation technique which avoids the drawbacks associated with the use of such filters has been developed. This new technique is largely based upon a software algorithm and therefore does not require the use of hardware filters. The theoretical basis of the new algorithm has been presented along with a method by which the algorithm can easily be incorporated within a composite frequency response analyser.

A novel, high performance frequency response analyser based upon the two composite frequency waveforms has been developed, which incorporates this anti-aliasing algorithm.

The effectiveness of the new anti-aliasing algorithm has been demonstrated by applying the developed instrument to a number of real systems. It is evident from the results obtained that the algorithm may be applied in place of an anti-aliasing filter, thereby offering significant advantages by reducing system complexity and cost.

In addition, the newly developed analyser has been used to evaluate the relative performance of the two complex frequency test waveforms. This has been done by applying the two waveforms to both a linear system and systems which exhibit slight non-linear behaviour.

Results of the tests show the significant advantages offered by composite frequency test procedures, in terms of measurement speed. When these signals were applied to quasi-linear systems however, some scattering of the spectral estimates was observed when compared to those obtained from a conventional monotonic frequency response analyser. Measurements taken show empirically, that although the use of the MPRBS signal offers a rapid method of obtaining the response of a dynamic system, the spectral estimates obtained are readily corrupted by slight non-linear behaviour in the system under test. The prime composite waveform on the other hand offers both a rapid estimation method and considerably improved spectral estimates over the MPRBS signal in the presence of non-linearities.

Performance tests undertaken on the new analyser demonstrate the feasibility of producing a commercial analyser based upon composite frequency test signals, which has a performance equal to or greater than existing analysers in terms of speed of measurement, frequency range and resolution.

In concluding this thesis the following areas are put forward as appropriate for further study.

(i) It has been shown that the instrument designed and developed during the investigation is unable to achieve maximum resolution because of electrical noise generated within the analyser. In order to fully realise the potential high accuracy, it is seen as necessary to produce a new version of the instrument using Printed Circuit Board technology.

(ii) As discussed in Chapter 9, the new anti-aliasing technique incorporates an interpolation strategy to evaluate system performance between measured points. A further fruitful area of study therefore, would be to evaluate the effect on the accuracy of the new technique of different interpolation algorithms.

(iii) It has been shown that the new anti-aliasing technique is theoretically applicable to any digital pulse train. Because of the widespread use of maximum length Psudo Random Binary Signals for the testing of system dynamics, it is seen to be of great value to apply the technique to such a waveform and produce empirical evidence of the effectiveness of the technique in such an application.

(iv) As discussed in Chapter 2 various signals, each exhibiting different attributes, have been put forward as being suitable for use in system frequency response testing. By incorporating a number of these waveforms within the new analyser a practical evaluation of the relative merits of the waveforms may be carried out.

REFERENCES

1. SCHLUMBERGER INTRUMENTS: 'The 1170 Frequency Response Analyser Manual',Part No A1720200, Slumberger, Victoria Road, Farnborough, Hampshire. Jan 1980.

2. SE LABS(EMI) Ltd. .'Frequency Response Analyser SE2450 Manual', Issue 1, SE Labs North Feltham Trading Estate, Felthamm, Middlesex. April 1981.

3. SCHLUMBERGER INSTRUMENTS: 'The 1250 Frequency Response Analyser Manual', Issue 6, Schlumberger, Victoria Road, Franborough, Hampshire. Oct 1985.

4. ELSDEN, C.S. and LEY, A.J.: 'A digital transfer Function analyser based on pulse rate techniques'. Automatica, Vol. 5, pp 51-60, 1969.

5. SOLATRON INSTRUMENTATION GROUP: 'The 1200 Signal Processor', SI/Dyn/021/Issue 2/3/3.82. The Solatron Electronic Group Ltd, Farnborough, Hampshire. 1981.

6. BALCOMB, J.D., DEMUTH, H.B., GYFTOPOULOS, E.P.: 'A Cross Correlation Method for Measuring the Impulse Response of Reactor Systems', Nuclear Science and Eng, Vol 11, pp 159-166, 1961.

7. HUGHES, M.T.G. and NORTON, A.R.M.: ' The Measurement of Control System Characteristics by Means of a Cross Correlator', Proc IEE, Vol 109, Part B, No 43, 1962.

8. SCHWARZENBACH, J. and GILL, K.F.: 'System Modelling and Control', Arnold.

9. NIKIFORUK, P.N. and GUPTA, M.M.: 'A Bibliography on the Properties, Generation and Control System Applications of Shift-Register Sequences', Inst. J. Control, Vol 2, pp 217-234, 1969.

10. BRIGGS, P.A.N., HAMMOND, P.H., HUGHES, M.T.G. and PLUMB, G.O.: 'Correlation analysis of processdynamics using pseudo-random binary test perturbations'. Proc. IME, Vol. 179, Pt. 3H.

11. CORRAN, E.R., CUMMINS, J.D. and HOPKINSON, A.: 'Identification of some Cross-Flow Heat Exchanger Dynamic Responses by Measurement with Low-Level Binary Pseudo-Random Input Signals', Atomic Energy Establishment, Winfrith, Dorset, Report No 373, 1964.

12. GODFREY, K.R.: 'Theory of the Correlation Method of Dynamic Analysis and its Application to Industrial Processes and Nuclear Plant', Trans Inst Meas and Control, Vol 2, T65-72 May 1969.

13. SCHROEDER, M R: 'Synthesis of low peak-factor signals and binary sequences of low auto-correlation'. IEEE Trans. on Info. Theory, pp 85-89, Jan 1970.

14. VAN BRUSSEL, H.: 'Comparative Assessment of Harmonic, Random, Sweptsine and Shock Excitation Methods for the Identification of Machine Tool Structures with Rotating Spindles', Annals of the CIRP, Vol 24,pp 291-296, 1975.

15. BUCKNER, M.R. and KERLIN, T.W.: 'Optimum binary signals for reactor frequency response measurements'. Nuclear Science and Engineering, 1972, 49, 255-262.

16. HARRIS S L, MELLICHAMP D A: 'On Line Identification of Process Dynamics: Multifrequency Binary Sequences'. Ind Eng Chem, Process Des Dev, 1980, 19, 166-174.

17. PAEHLIKE, K.D. and RAKE, H.: 'Binary Multifrequency Signals - Synthesis and Application', 5th IFAC Symposium, "Identification and System Parameter Estimation", Darmstadt, pp 589-596, Sept 1979.

18. VAN DEN BOS, A.: 'Construction of binary multifrequency signals'. 1^{st} IFAC Symposium, 'Identification in automatic control systems', Prague, Paper 4.6, June 1967.

19. LAMB, J.D. and REES, D.: 'Digital Processing of System Response to Obtain Frequency Response Characteristics using the Fast Fourier Transform', IEE Conf Pub 103, "The Use of Digital Computers in Measurement", pp 141-146, 1973.

20. BARKER, T.D., DAVEY, B.A.:'System identification using psudo-random signals and the discrete Fourier

transform'. Proc. IEE. Vol. 122, No.3, pp 305-311, 1975.

21. FLOWER, J.O., FORGE, S.C., KNOTT. G.:'On-line computer determination of the unsteady lift characteristics of hydrofoils by a multifrequency technique'. Int. Shipbuilding Progress, pp 313-316, 1976.

22. REES, D.: 'System Identification in the Presence of Non-Linearities using Deterministic Signals and the Fast Fourier Transform', Proc IEE Colloquium "Identification of Non-Linear Systems", Digest No 1978/31, May 1978.

23. FLOWER, J.O., KNOTT G.F., FORGE S.C.: 'Application of Schroeder-phased Harmonic Signals to Practical Identification'. Measurement and Control, 1978, 11, 69-73.

24. FLOWER, J.O., FORGE, S.C., RATCLIFFE, N.G. and ROUST, C.B.: 'Dynamic measurements of a nuclear reactor using low-peak-factor excitation signals'. Nuclear Science and Engineering, 68, 110-115, 1978.

25. VAN DEN BOS, A.: 'A new method for synthesis of low-peak factor signals. IEEE Trans. on Acoustics, Speech and Signal Processing, ASSP-35, pp 120-122, 1987.

26. VAN DER OUDERAA E, SCHOUKENS J, RENNEBOOG J: 'Peak Factor minimisation using a time-frequency domain swapping algorithm'. IEEE Trans on Instrumentation and

Measurement, 1988, IM-37, 145-147.

27. REES, D: 'Systems identification using composite frequency signals with low peak factors'. Proc. Int. AMSE Conf. "Modelling and Simulation", Sorrento, (Italy) Sept 29 - Oct 1, 1986, Vol.1 and 2, p 13-24.

28. POWER, H M and SIMPSON, R J: 'Introduction to Dynamics and Control'. McGraw Hill, pp 134-143.

29. JAWAD, S M and PAYNE, P A: 'Advanced frequency response techniques for automatic testing'. Proc. Automatic Test and Test Instrumentation Conference, pp 40-51, 1983.

30. BRIGHAM, E.I.: 'The fast Fourier transform', Prentice Hall, 1974.

31. COOLEY, J.S. and TUKEY, J.W.: 'An algorithm for the machine calculation of complex Fourier Series'. Math. Computation, 19, pp 297-301, 1965.

32. DAVIES, W.D.T.: 'System identification for self adaptive control'. Wiley Interscience. 1970.

33. HEWLETT PACKARD: 'The 3582A spectrum analyser manual'. Hewlett Packard Europe, PO Box 999, 1180 AZ Amstelveen, The Netherlands, May 1983.

34. TAKEDA RIKEN: 'TR9405 performance and feature of digital spectrum analyser'. Application Note No 4-2E Takeda Riken Industry Co Ltd 1-32.1, Asahi-cho, Nerima-ku, Tokyo 176 Japan. 1982.

35. SCHLUMBERGER INSTRUMENTS: 'The SI1220 multichannel spectrum analyser' SI/MDV/002/Issue 1/Jan 8. The Salatron Electronic Group Ltd, Farnborough, Hampshire. Jan 1989.

36. HEWLETT PACKARD: 'The 3562A Dynamic Signal Analyser' Hewlett Packard Europe, PO Box 999, 1180 AZ, Amstelveen, The Netherlands. June 1986

37. ADVANTEST: 'The R9211E Digital Spectrum Analyser' The Advantest Corporation, 4-1 Nishi-shinjuku 2-chrome, Shinjuku-ku, Tokyo, Japan. Aug 1989.

38. GESPAC: 'GESMPU-4A High Performance 68000 Processor Module (16 bits)', Gespac SA 3, chemin des Autx, CH - 1228 Geneve. 1984.

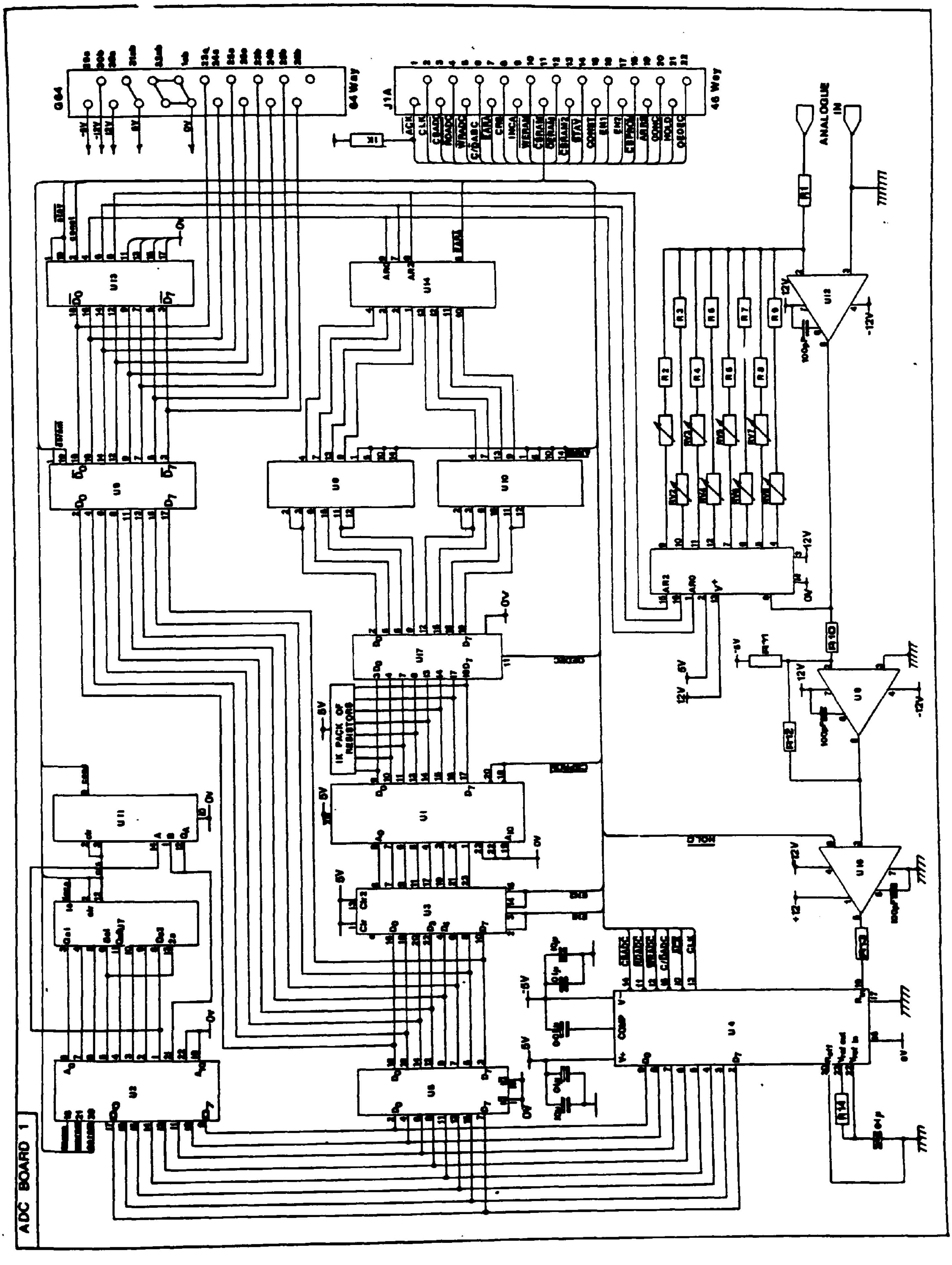
39. RABINER, L. R. and GOLD, B.: 'Theory and application of digital signal processing'. Prentice-Hall. 1975.

40. FAIRCHILD, V: 'Zooming in the Frequency Domain' Electronics Industry, Dec 1982.

41. FEEDBACK INSTRUMENTS: 'Process control simulator PCS327 manual'. Manual 327-1. Feedback Instruments Ltd, Park Road, Cowborough, Sussex. Rev 1081.

Appendix A

Instrument Circuit Diagrams

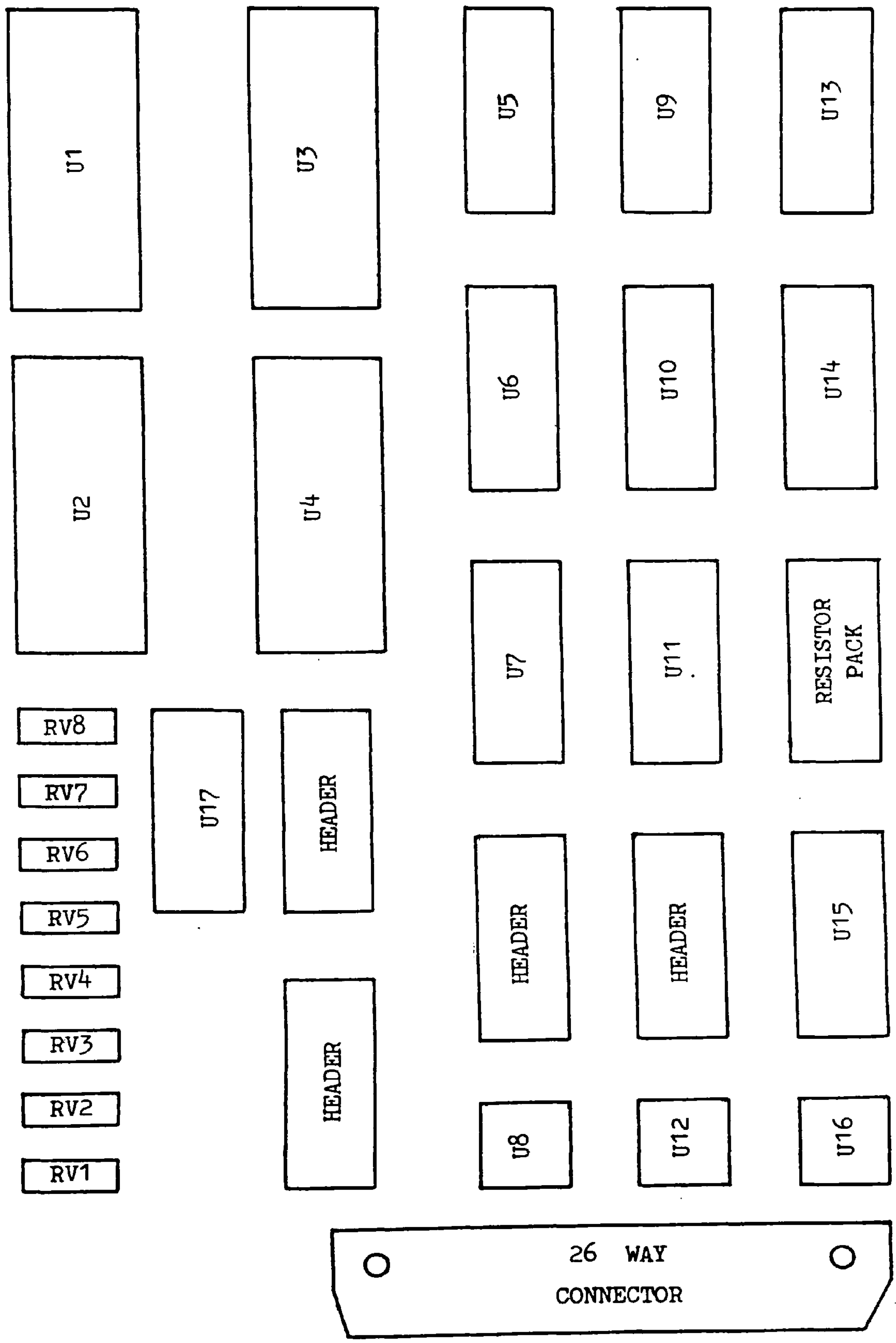

ADC BOARD 1
64 Way
46 Way
ANALOGUE IN
1K PACK OF RESISTORS

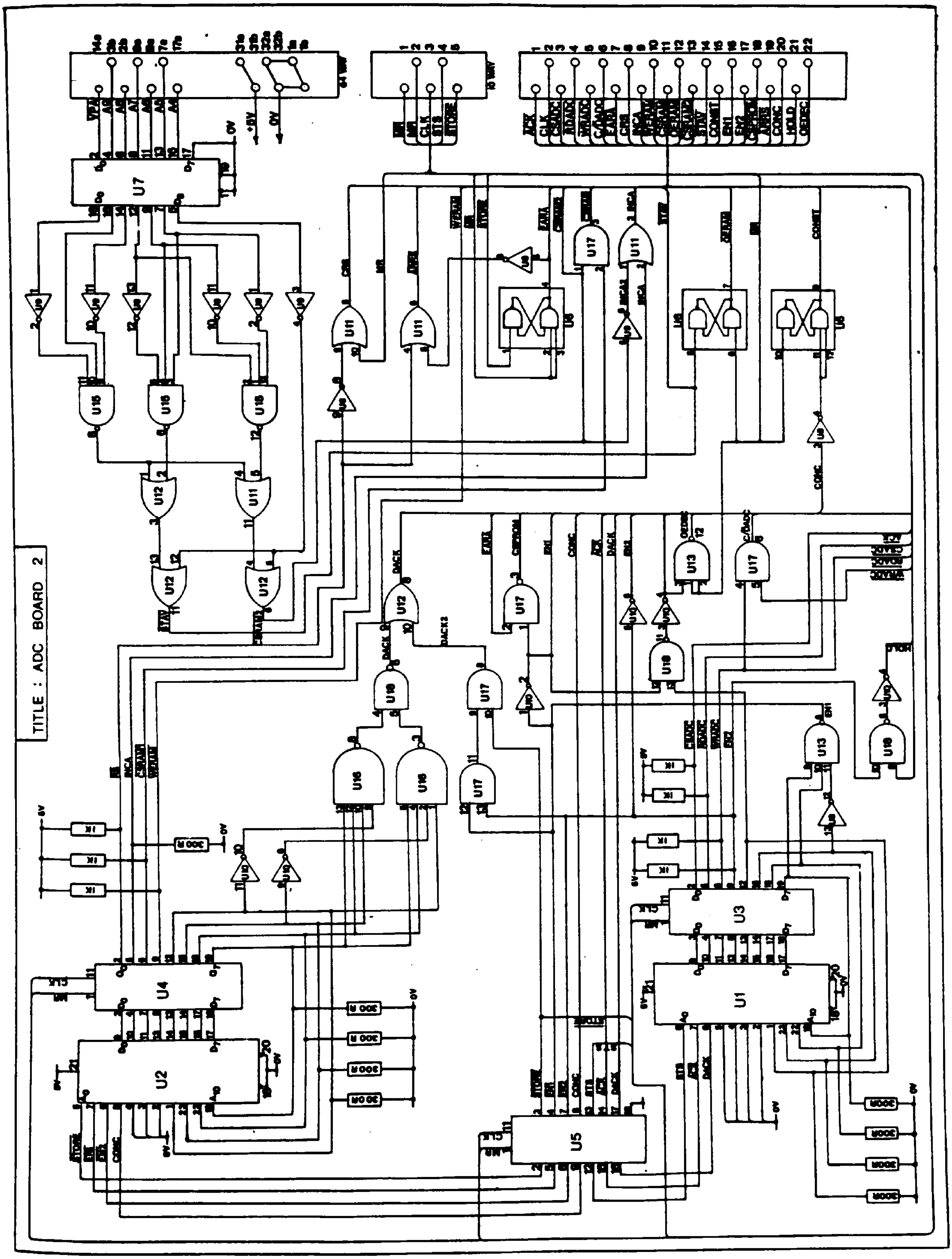
COMPONENTS LIST FOR ADC BOARD 1

Component	Part No.	Description
U1	2716	EPROM 450nS ACCESS TIME
U2	6116-3	2048 x 8 BIT CMOS RAM 150nS ACCESS TIME
U3	74116	TTL DUAL 4 BIT LATCH
U4	AM6112	12 BIT ADC
U5	74LS244	TTL NON-INVERTING OCTAL LATCH
U6	74LS279	TTL QUAD $\overline{S}$-$\overline{R}$ LATCHES
U7	74LS393	TTL DUAL 4 BIT COUNTER
U8	2505-5	HIGH SLEW RATE OPERATIONAL AMPLIFIER
U9	74LS240	TTL INVERTING OCTAL BUFFER
U10	74LS279	TTL QUAD $\overline{S}$-$\overline{R}$ LATCHES
U11	74LS93	TTL 4 BIT COUNTER
U12	2505-5	HIGH SLEW RATE OPERATIONAL AMPLIFIER
U13	74LS240	TTL INVERTING OCTAL BUFFER
U14	74LS148	TTL 8 TO 3 LINE PRIORITY ENCODER
U15	DG508	MULTIPLEXED ANALOGUE SWITCH
U16	LF398	SAMPLE AND HOLD IC
U17	74LS374	TTL OCTAL D-TYPE FLIP FLOPS
R1		10K RESISTOR TOL. ± 5%
R2		5K6 RESISTOR TOL. ± 5%
R3		8K2 RESISTOR TOL. ± 5%
R4		8K2 RESISTOR TOL. ± 5%
R5		12K RESISTOR TOL. ± 5%
R6		22K RESISTOR TOL. ± 5%
R7		47K RESISTOR TOL. ± 5%
R8		82K RESISTOR TOL. ± 5%
R9		470K RESISTOR TOL. ± 5%

Component	Part No.	Description
R10		10K RESISTOR TOL. ± 5%
R11		10K RESISTOR TOL. ± 5%
R12		10K RESISTOR TOL. ± 5%
R13		50R RESISTOR TOL. ± 2%
R14		50R RESISTOR TOL. ± 2%
RV1		1K 20 TURN POT
RV2		1K 20 TURN POT
RV3		5K 20 TURN POT
RV4		1K 20 TURN POT
RV5		5K 20 TURN POT
RV6		10K 20 TURN POT
RV7		20K 20 TURN POT
RV8		100K 20 TURN POT

COMPONENT LAYOUT FOR ADC BOARD 1

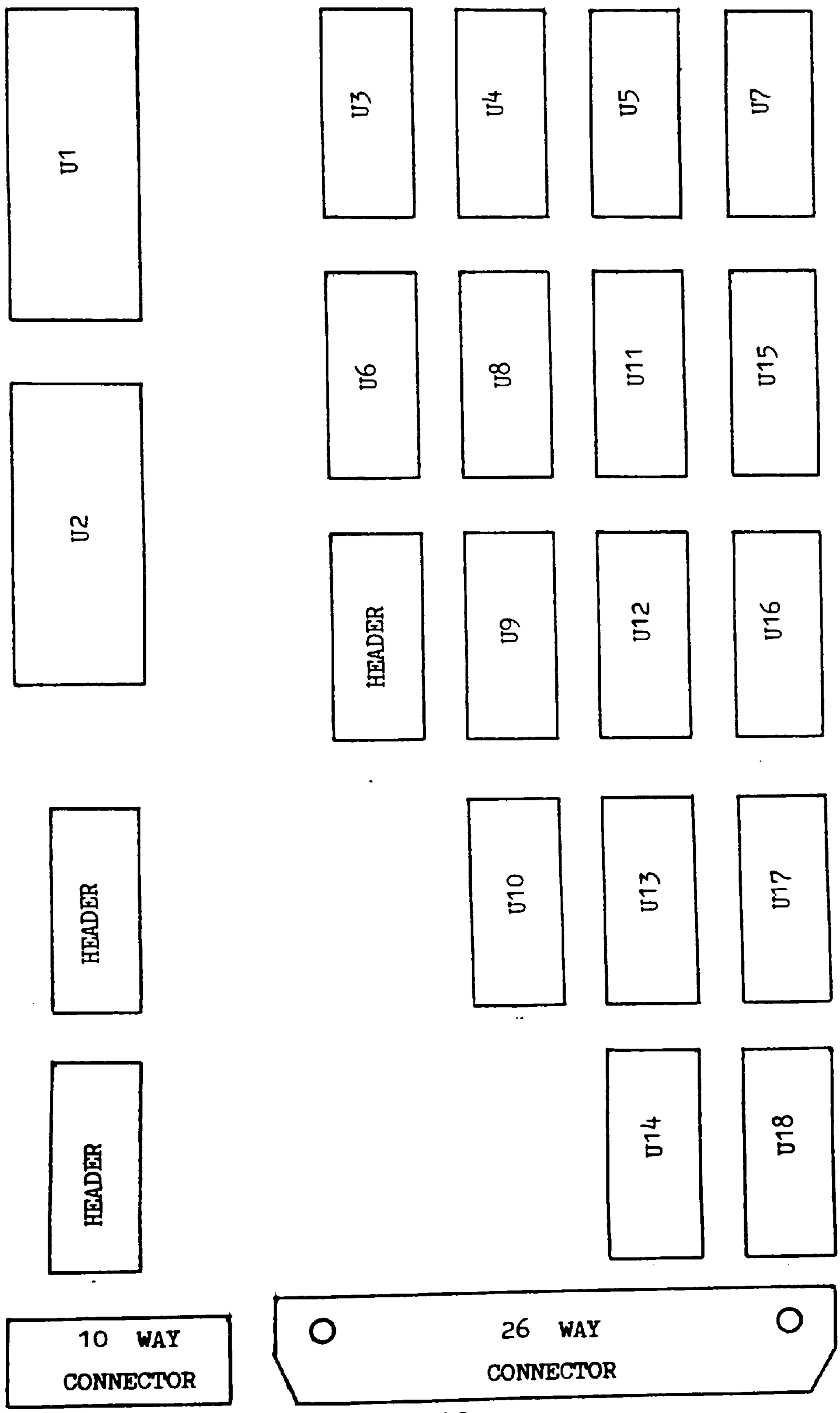


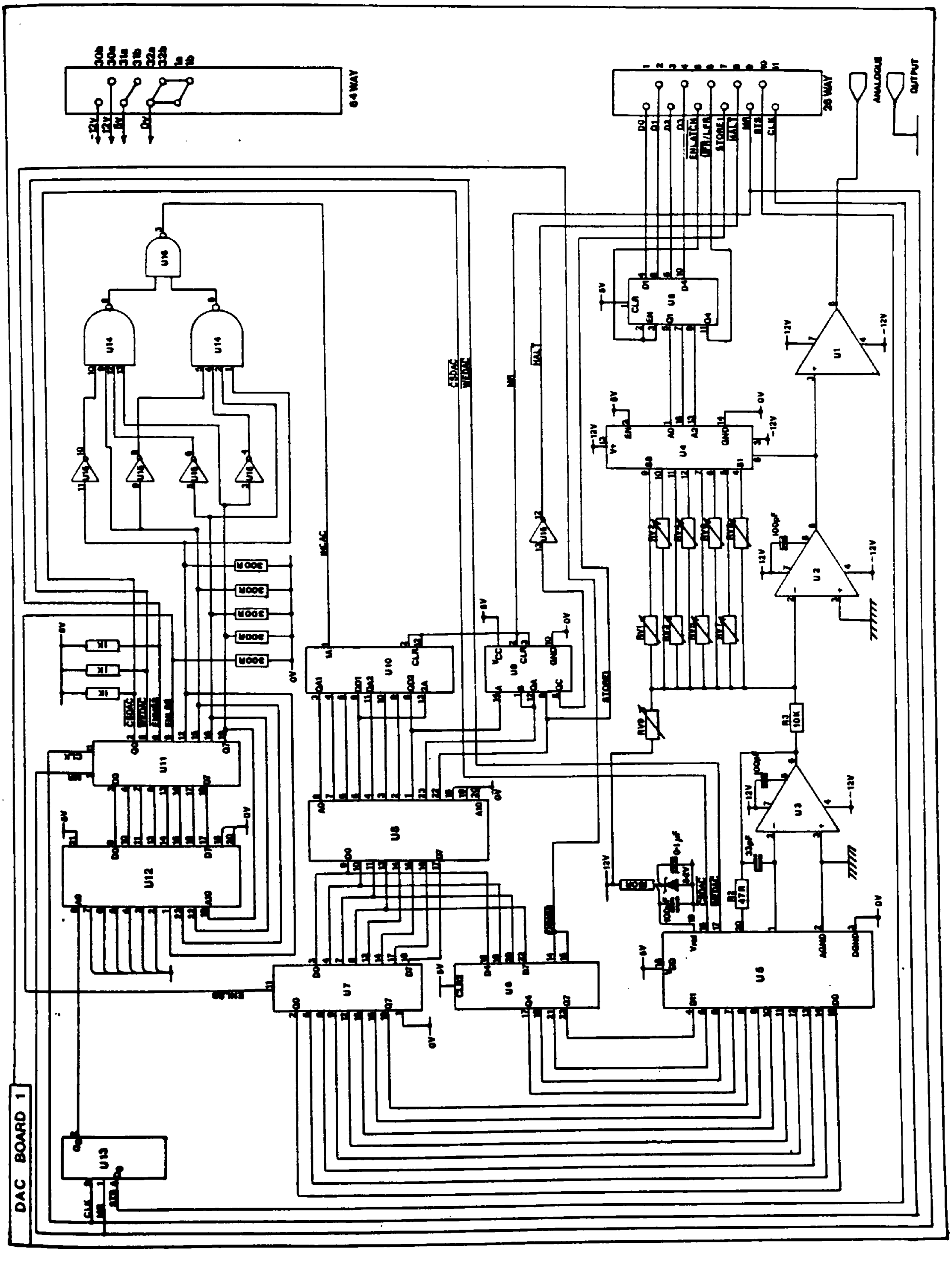

TITLE : ADC BOARD 2

COMPONENTS LIST FOR ADC BOARD 2

Component	Part No.	Description
U1	2716	EPROM 450nS ACCESS TIME
U2	2716	EPROM 450nS ACCESS TIME
U3	74LS374	TTL OCTAL D-TYPE FLIP FLOPS
U4	74LS374	TTL OCTAL D-TYPE FLIP FLOPS
U5	74LS273	TTL OCTAL D-TYPE FLIP FLOPS
U6	74LS279	TTL QUAD $\overline{S}$-$\overline{R}$ LATCHES
U7	74LS244	TTL NON-INVERTING OCTAL LATCH
U8	74LS04	TTL HEX INVERTERS
U9	74LS14	TTL HEX SCHMITT TRIGGER INVERTERS
U10	74LS04	TTL HEX INVERTERS
U11	74LS32	TTL QUADRUPLE 2 INPUT OR GATES
U12	74LS32	TTL QUADRUPLE 2 INPUT OR GATES
U13	74LS10	TTL TRIPLE 3 INPUT NAND GATES
U14		SPARE SOCKET
U15	74LS10	TTL TRIPLE 3 INPUT NAND GATES
U16	74LS20	TTL DUAL 4 INPUT NAND GATES
U17	74LS08	TTL QUADRUPLE 2 INPUT AND GATES
U18	74LS00	TTL QUADRUPLE 2 INPUT NAND GATES

COMPONENT LAYOUT FOR ADC BOARD 2

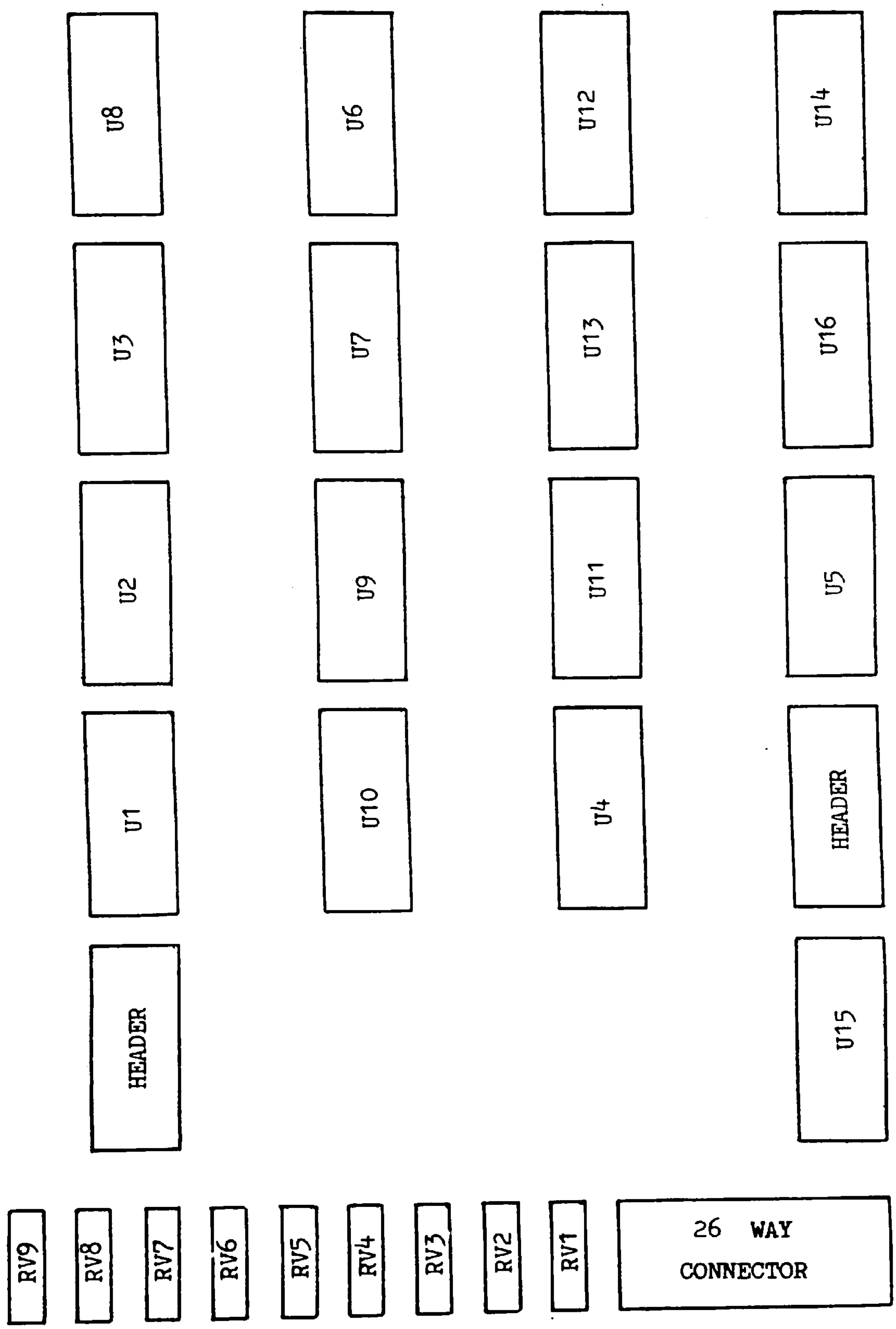


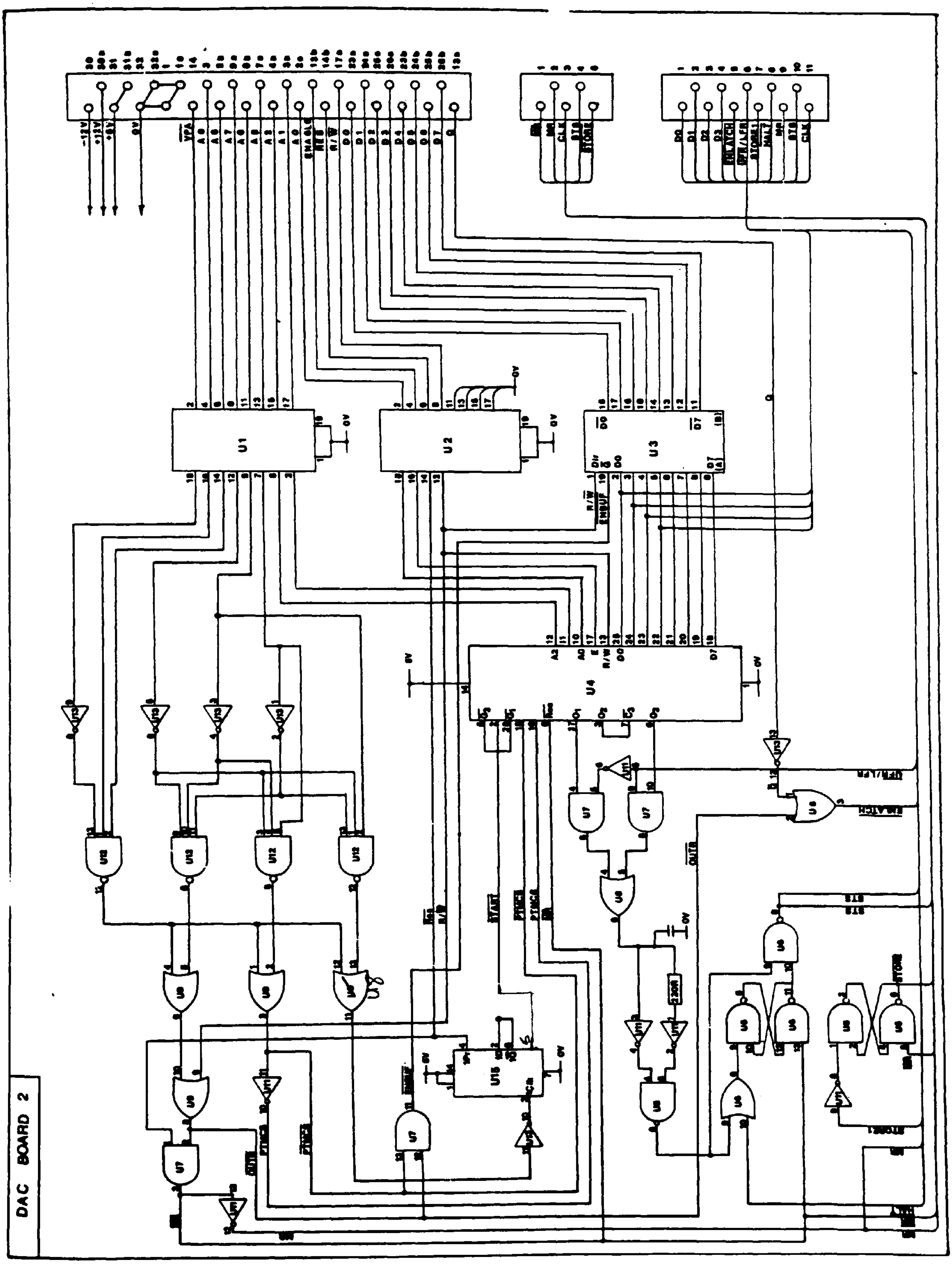

DAC BOARD 1
64 WAY
26 WAY

COMPONENTS LIST FOR DAC BOARD 1

Component	Part No.	Description
U1	LM310	VOLTAGE FOLLOWER
U2	2505-5	HIGH SLEW RATE OPERATIONAL AMPLIFIER
U3	2505-5	HIGH SLEW RATE OPERATIONAL AMPLIFIER
U4	DG508	MULTIPLEXED ANALOGUE SWITCH
U5	7545	12 BIT DAC
U6	74116	TTL DUAL 4 BIT LATCH
U7	74LS373	TTL OCTAL D-TYPE LATCHES
U8	2716	EPROM 450nS ACCESS TIME
U9	74LS93	TTL 4 BIT COUNTER
U10	74LS393	TTL DUAL 4 BIT COUNTER
U11	74LS374	TTL OCTAL D-TYPE FLIP FLOPS
U12	2716	EPROM 450nS ACCESS TIME
U13	74LS175	TTL QUAD D-TYPE FLIP FLOPS
U14	74LS20	TTL DUAL 4 INPUT NAND GATES
U15	74LS04	TTL HEX INVERTERS
U16	74LS00	TTL QUADRUPLE 2 INPUT NAND GATES
RV1		20K 20 TURN POT
RV2		20K 20 TURN POT
RV3		10K 20 TURN POT
RV4		10K 20 TURN POT
RV5		5K 20 TURN POT
RV6		5K 20 TURN POT
RV7		5K 20 TURN POT
RV8		1K 20 TURN POT
RV9		**50K 20 TURN POT**

COMPONENT LAYOUT FOR DAC BOARD 1

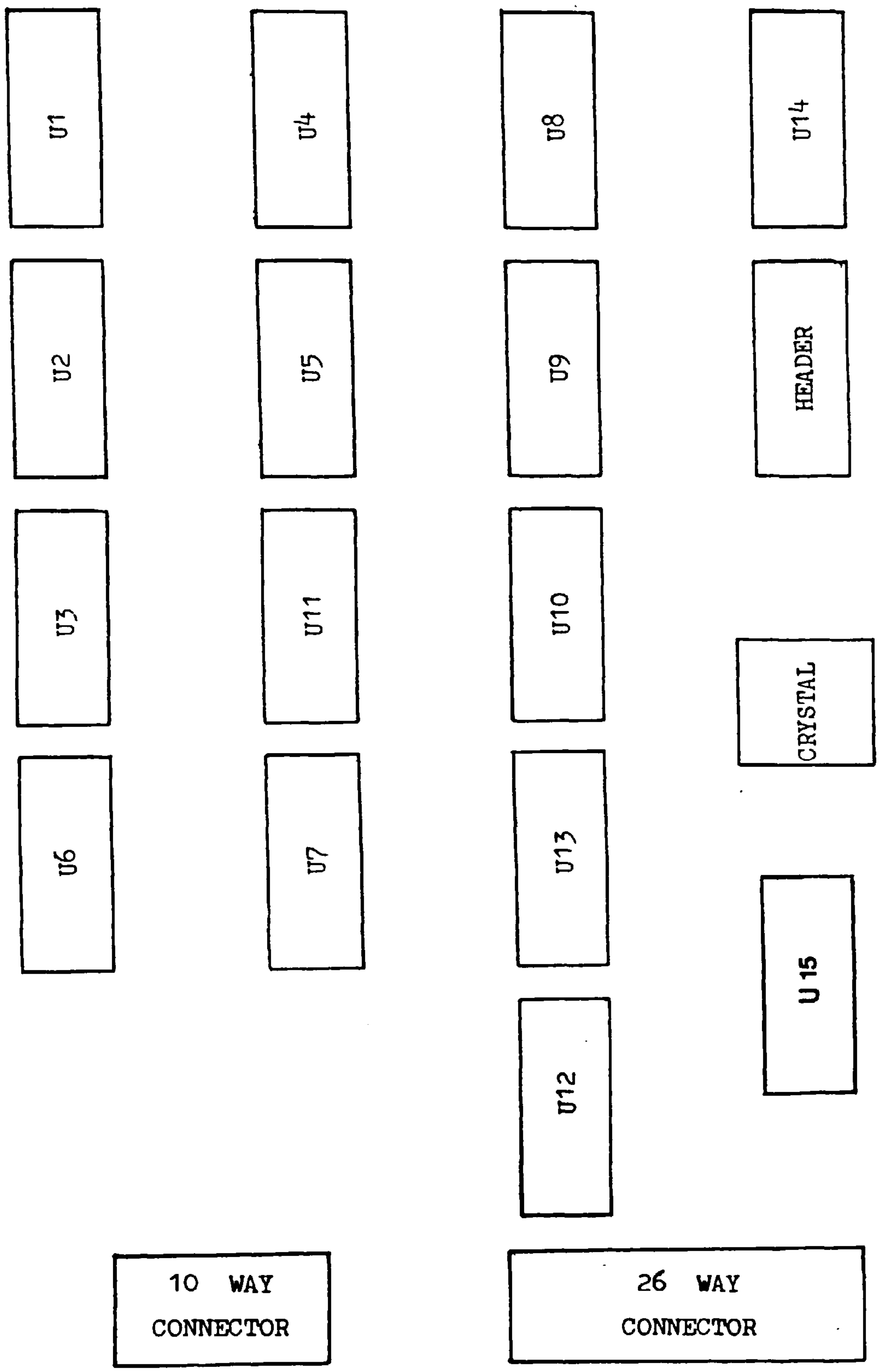



DAC BOARD 2

COMPONENTS LIST FOR DAC BOARD 2

Component	Part No.	Description
U1	74LS244	TTL NON-INVERTING OCTAL LATCH
U2	74LS244	TTL NON-INVERTING OCTAL LATCH
U3	74LS640	TTL OCTAL BUS TRANCEIVER
U4	6840	PROGRAMMABLE TIMER MODULE
U5	74LS00	TTL QUADRUPLE 2 INPUT NAND GATES
U6	74LS00	TTL QUADRUPLE 2 INPUT NAND GATES
U7	74LS08	TTL QUADRUPLE 2 INPUT AND GATES
U8	74LS32	TTL QUADRUPLE 2 INPUT OR GATES
U9	74LS32	TTL QUADRUPLE 2 INPUT OR GATES
U10	74LS10	TTL TRIPLE 3 INPUT NAND GATES
U11	74LS14	TTL HEX SCHMITT TRIGGER INVERTERS
U12	74LS10	TTL TRIPLE 3 INPUT NAND GATES
U13	74LS04	TTL HEX INVERTERS
U14	74LS320	TTL CRYSTAL CONTROLLED OSCILLATOR
U15	**74LS74**	**DUAL D-TYPE F-F**

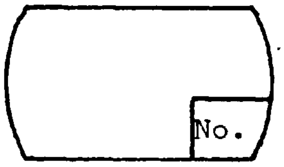
COMPONENT LAYOUT FOR DAC BOARD 2



Appendix B

State Diagrams for Instrument Sequencers

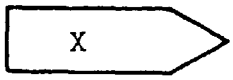
NOTATION USED FOR STATE DIAGRAMS



- State.

- Output signal from sequencer.

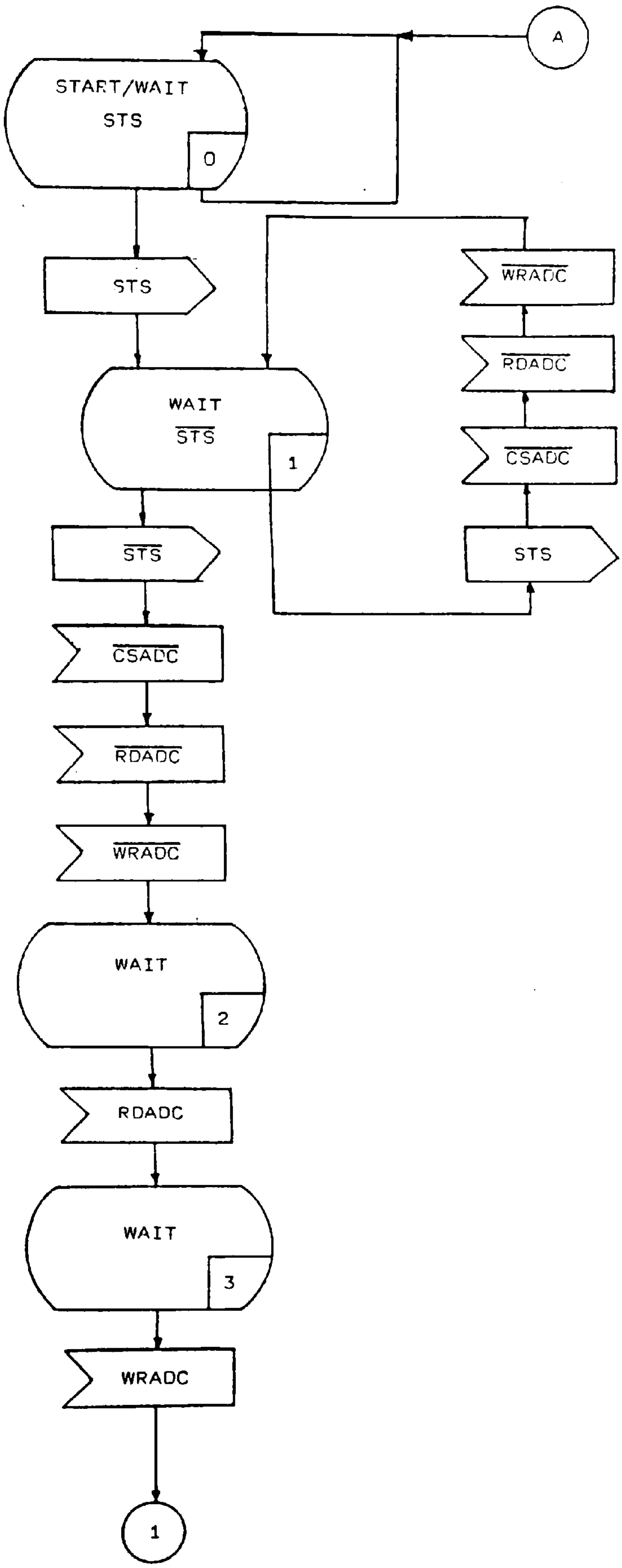
- Input signal to sequencer.

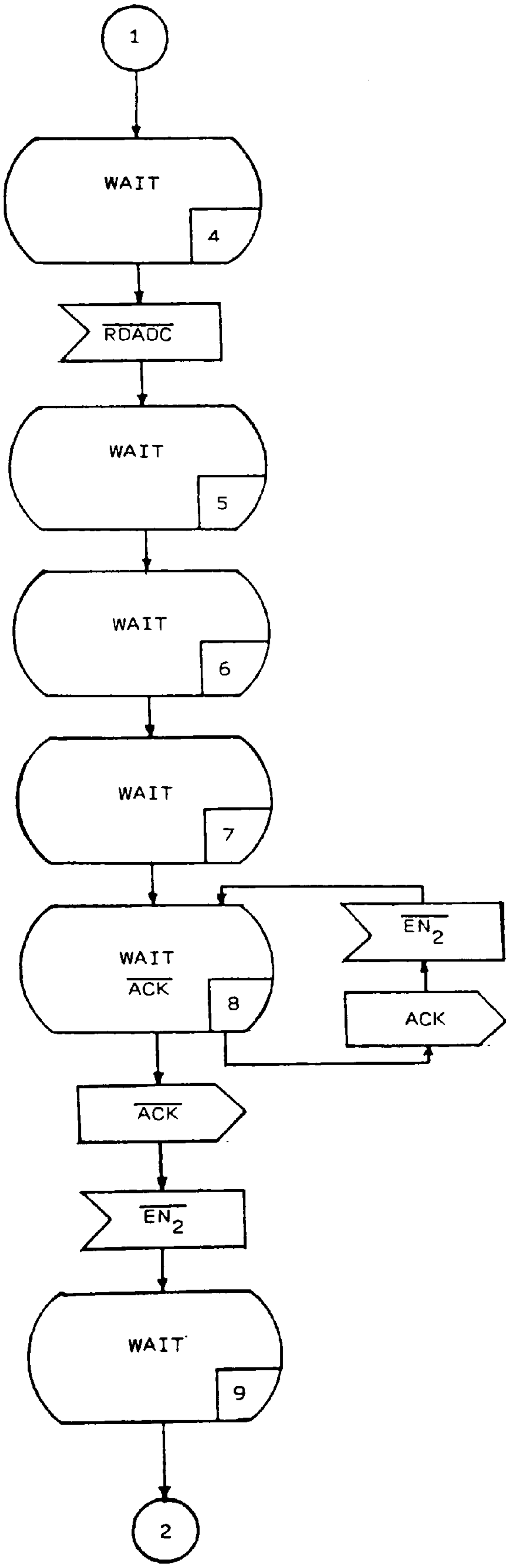


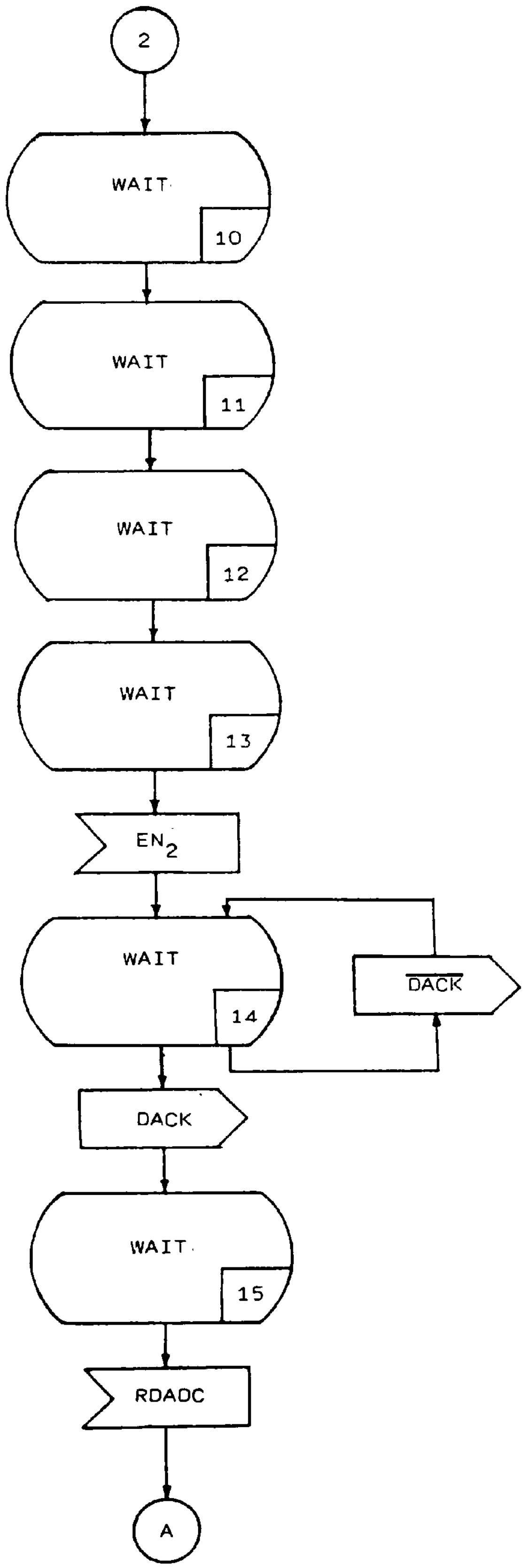
- Signal at high level.

- Signal at low level.

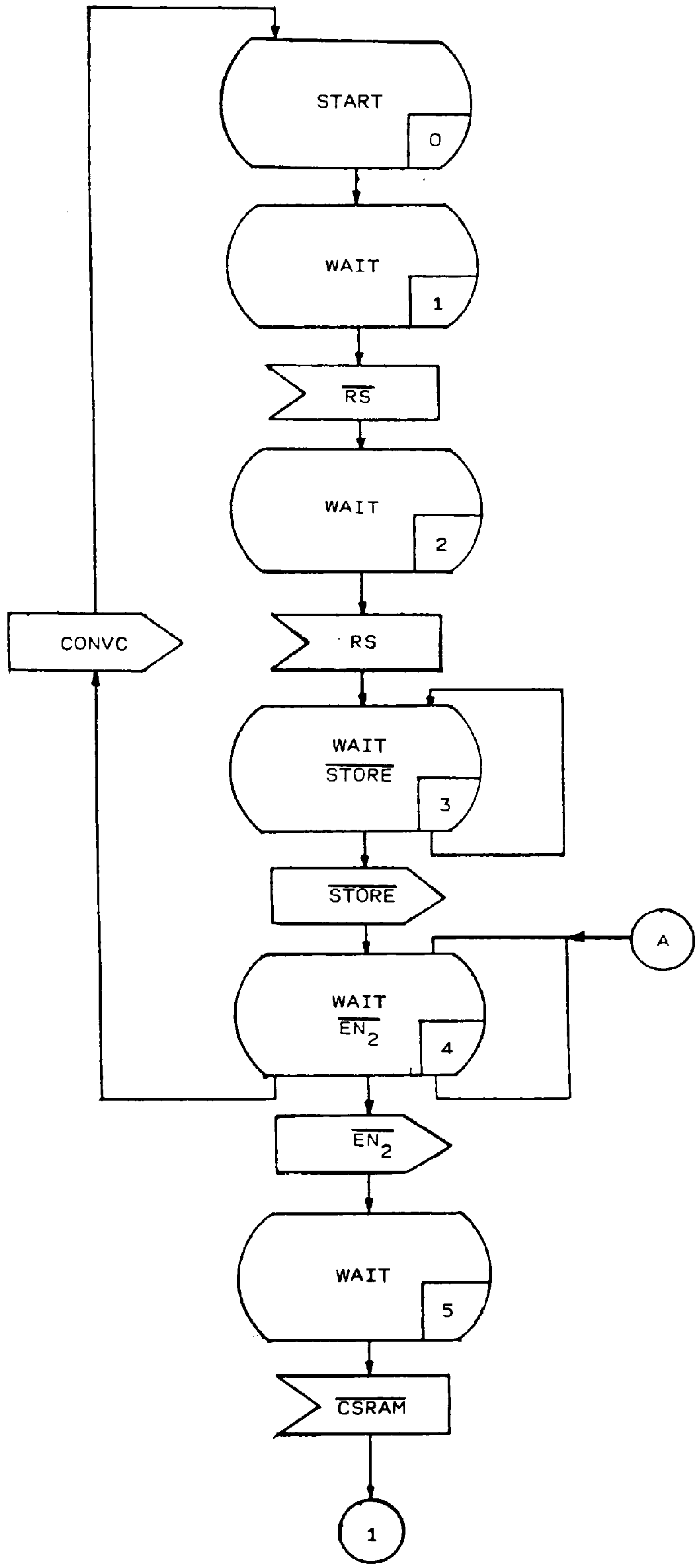
STATE DIAGRAM FOR ADC CONTROL

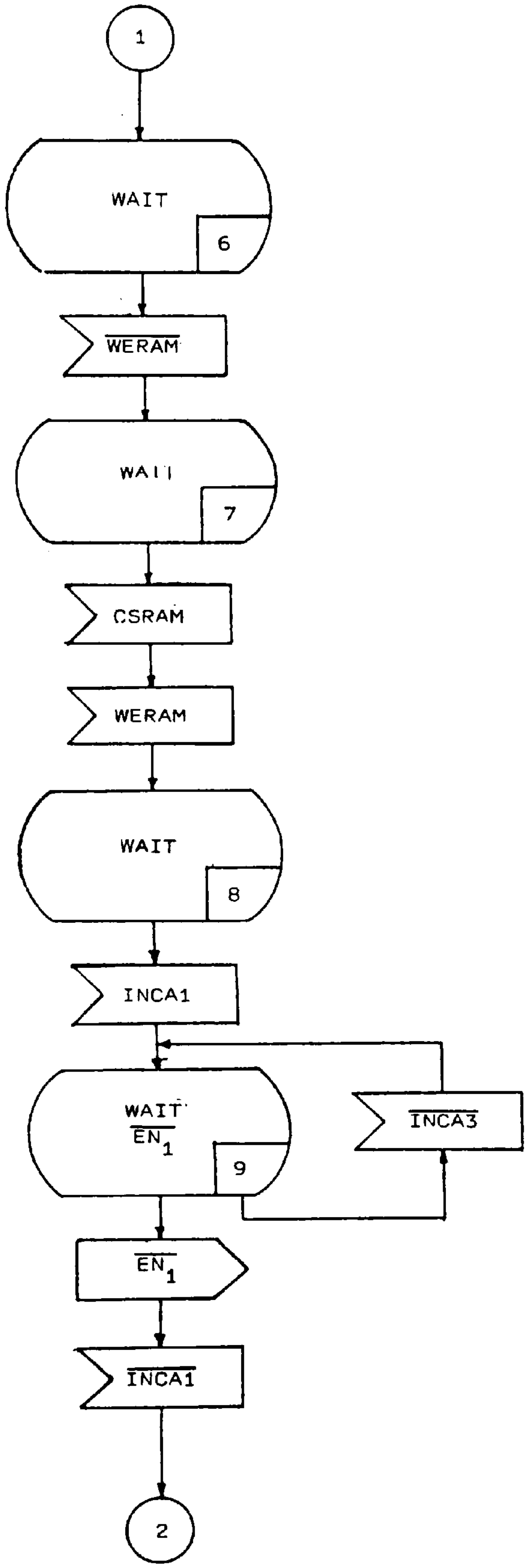


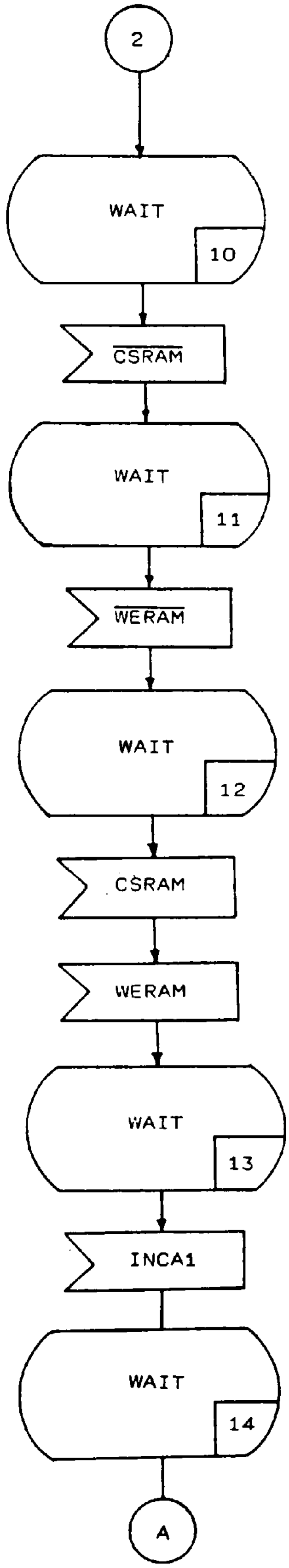

1
WAIT
4
RDADC
WAIT
5
WAIT
6
WAIT
7
WAIT
ACK
8
EN2
ACK
ACK
EN2
WAIT
9
2


2
WAIT
10
WAIT
11
WAIT
12
WAIT
13
EN2
WAIT
14
DACK
DACK
WAIT
15
RDADC
A

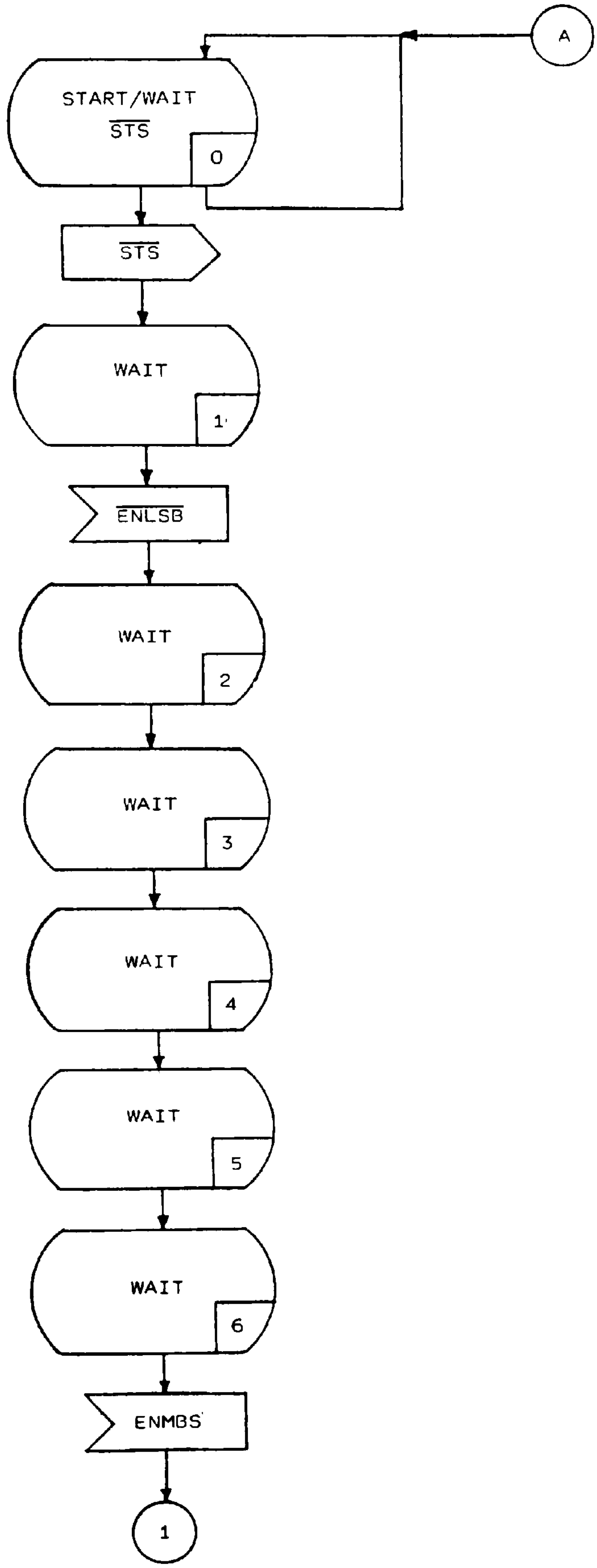
STATE DIAGRAM FOR DATA FLOW CONTROL FOR ADC BOARD

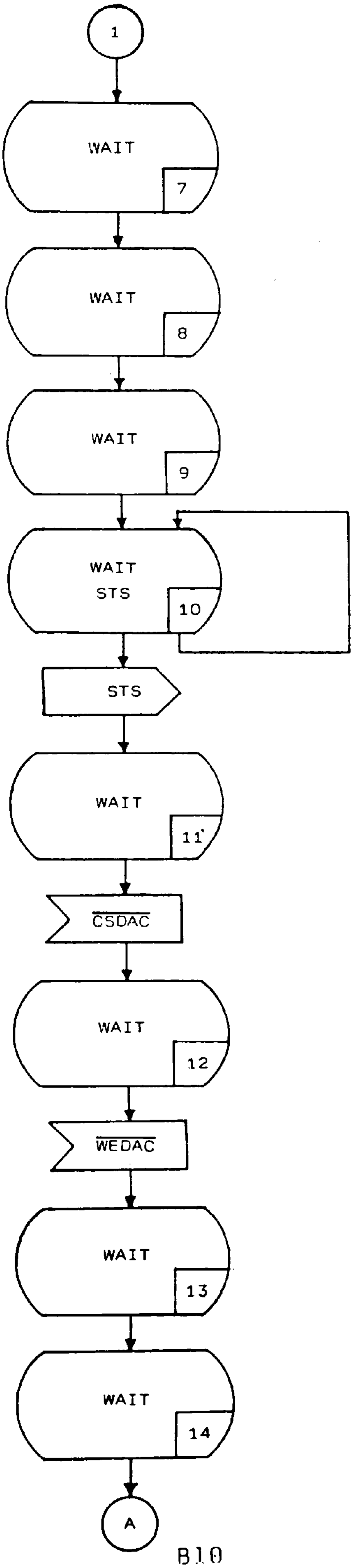



1
WAIT
6
$\overline{WERAM}$
WAIT
7
CSRAM
WERAM
WAIT
8
INCA1
WAIT
$\overline{EN_1}$
9
$\overline{INCA3}$
$\overline{EN_1}$
$\overline{INCA1}$
2


2
WAIT
10
$\overline{CSRAM}$
WAIT
11
$\overline{WERAM}$
WAIT
12
CSRAM
WERAM
WAIT
13
INCA1
WAIT
14
A

STATE DIAGRAM FOR DATA FLOW CONTROL FOR DAC BOARD




1
WAIT
7
WAIT
8
WAIT
9
WAIT
STS
10
STS
WAIT
11
CSDAC
WAIT
12
WEDAC
WAIT
13
WAIT
14
A

B10

CURRENT STATE				CONTROL INPUTS				NEXT STATE				CONTROL OUTPUTS			
A_{10}	A_9	A_8	A_7	A_3	DACK A_2	$\overline{ACK}$ A_1	STS A_0	D_7	D_6	D_5	D_4	$\overline{EN}_2$ D_3	$\overline{WRADC}$ D_2	$\overline{RDADC}$ D_1	$\overline{CSADC}$ D_0
L	L	L	L	X	X	X	L	L	L	L	L	H	H	H	H
L	L	L	L	X	X	X	H	L	L	L	H	H	H	H	H
L	L	L	H	X	X	X	L	L	L	H	L	H	L	L	L
L	L	L	H	X	X	X	H	L	L	L	H	H	L	L	L
L	L	H	L	X	X	X	X	L	L	H	H	H	L	H	L
L	L	H	H	X	X	X	X	L	H	L	L	H	H	H	L
L	H	L	L	X	X	X	X	L	H	L	H	H	H	L	L
L	H	L	H	X	X	X	X	L	H	H	L	H	H	L	L
L	H	H	L	X	X	X	X	L	H	H	H	H	H	L	L
L	H	H	H	X	X	X	X	H	L	L	L	H	H	L	L
H	L	L	L	X	X	L	X	H	L	L	H	L	H	L	L
H	L	L	L	X	X	H	X	H	L	L	L	L	H	L	L
H	L	L	H	X	X	X	X	H	L	H	L	L	H	L	L
H	L	H	L	X	X	X	X	H	L	H	H	L	H	L	L
H	L	H	H	X	X	X	X	H	H	L	L	L	H	L	L
H	H	L	L	X	X	X	X	H	H	L	H	L	H	L	L
H	H	L	H	X	X	X	X	H	H	H	L	H	H	L	L
H	H	H	L	X	L	X	X	H	H	H	L	H	H	L	L
H	H	H	L	X	H	X	X	H	H	H	H	H	H	L	L
H	H	H	H	X	X	X	X	L	L	L	H	H	H	H	L

CONTROL MICROPROGRAM FOR ADC

CURRENT STATE				CONTROL INPUTS				NEXT STATE				CONTROL OUTPUTS			
				CONC	$\overline{EN_2}$	$\overline{EN_1}$	$\overline{STORE}$					$\overline{WERAM}$	$\overline{CSRAM}$	INCA1	$\overline{RS}$
A_{10}	A_9	A_8	A_7	A_3	A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
L	L	L	L	X	X	X	X	L	L	L	H	H	H	L	H
L	L	L	H	X	X	X	X	L	L	H	L	H	H	L	L
L	L	H	L	X	X	X	X	L	L	H	H	H	H	L	H
L	L	H	H	X	X	X	L	L	H	L	L	H	H	L	H
L	L	H	H	X	X	X	H	L	L	H	H	H	H	L	H
L	H	L	L	L	L	X	X	L	H	L	H	H	H	L	H
L	H	L	L	L	H	X	X	L	H	L	L	H	H	L	H
L	H	L	L	H	X	X	X	L	L	L	L	H	H	L	H
L	H	L	H	X	X	X	X	L	H	H	L	H	L	L	H
L	H	H	L	X	X	X	X	L	H	H	H	L	L	L	H
L	H	H	H	X	X	X	X	H	L	L	L	H	H	L	H
H	L	L	L	X	X	X	X	H	L	L	H	H	H	H	H
H	L	L	H	X	X	L	X	H	L	H	L	H	H	L	H
H	L	L	H	X	X	H	X	H	L	L	H	H	H	L	H
H	L	H	L	X	X	X	X	H	L	H	H	H	L	L	H
H	L	H	H	X	X	X	X	H	H	L	L	L	L	L	H
H	H	L	L	X	X	X	X	H	H	L	H	H	H	L	H
H	H	L	H	X	X	X	X	H	H	H	L	H	H	H	H
H	H	H	L	X	X	X	X	L	H	L	L	H	H	H	H

CONTROL MICROPROGRAM FOR DATA FLOW ON ADC BOARD

CURRENT STATE				CONTROL INPUTS				NEXT STATE				CONTROL OUTPUTS			
							$\overline{STS}$					ENMSB	$\overline{ENLSB}$	$\overline{WEDAC}$	$\overline{CSDAC}$
A_{10}	A_9	A_8	A_7	A_3	A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
L	L	L	L	X	X	X	L	L	L	L	H	L	H	H	H
L	L	L	L	X	X	X	H	L	L	L	L	L	H	H	H
L	L	L	H	X	X	X	X	L	L	H	L	L	L	H	H
L	L	H	L	X	X	X	X	L	L	H	H	L	L	H	H
L	L	H	H	X	X	X	X	L	H	L	L	L	L	H	H
L	H	L	L	X	X	X	X	L	H	L	H	L	L	H	H
L	H	L	H	X	X	X	X	L	H	H	L	H	L	H	H
L	H	H	L	X	X	X	X	L	H	H	H	H	L	H	H
L	H	H	H	X	X	X	X	H	L	L	L	H	L	H	H
H	L	L	L	X	X	X	X	H	L	L	H	H	L	H	H
H	L	L	H	X	X	X	X	H	L	H	L	H	L	H	H
H	L	H	L	X	X	X	L	H	L	H	L	H	L	H	H
H	L	H	L	X	X	X	H	H	L	H	H	H	L	H	H
H	L	H	H	X	X	X	X	H	H	L	L	H	L	H	L
H	H	L	L	X	X	X	X	H	H	L	H	H	L	L	L
H	H	L	H	X	X	X	X	H	H	H	L	H	L	L	L
H	H	H	L	X	X	X	X	L	L	L	L	H	L	L	L

1.3 CONTROL MICROPROGRAM FOR DATA FLOW ON DAC BOARD

APPENDIX C

High Level Designs of Instrument Software

Note: The High Level design format adopted in this appendix is a pseudocode based upon Pascal Control Structures.

User Interface and Signal Processing Modules

MODULE - fridft:s

FUNCTION - Provides user access to all system facilities via main menu;

```
FROM  VDU2.s  IMPORTS  ClearScreen,IntroOnScreen,DisplayMenu,

                           ExitMessage,ErrorStatement;

FROM  ftest:s  IMPORTS  SystemTest;

FROM  bode:s  IMPORTS  BodePrint;

FROM  busiee:s IMPORTS  IEEEControl,IEEEBode:

BEGIN

Clear terminal screen (ClearScreen);

Display Introduction (IntroOnScreen);

Clear terminal screen (ClearScreen);

WHILE  NOT exit  DO

      BEGIN

      Display Main System Menu (DisplayMenu);

      Read KEYINT from terminal;

      If (KEYINT >0) and (KEYINT <6) THEN

         CASE KEYINT OF

            1:  Perform frequency response analysis (SystemTest);

            2:  Display bode plot of results to screen (BodePrint);

            3:  Allow access to ATE via IEEE (IEEEControl);

            4:  Display bode plot of results to IEEE printer (IEEEBode);

            5:  Exit = TRUE; Clear terminal screen (ClearScreen);
```

```
     ELSE

     BEGIN
            Clear terminal screen (ClearScreen);

            Display error message (ErrorStatment);

     END;

Display exit message (exit message)

END fridft.
```

MODULE - ftest:s

FUNCTION - obtain the transfer function of the system under test;

EXPORTS SystemTest;

FROM pause:s IMPORTS PAUSE;

FROM VDU2:s IMPORTS ClearScreen;

FROM tfreq:s IMPORTS TestFreq;

FROM Ginn:s IMPORTS RAN,OP;

FROM range:s IMPORTS INIT;

FROM ft:s IMPORTS FFT;

FROM angmag:s IMPORTS ANG;

FROM pow:s IMPORTS POWELL;

FROM prbs:s IMPORTS PRBS;

FROM tranfn:s IMPORTS TF;

BEGIN SystemTest.

REPEAT

Clear terminal screen (ClearScreen):

Display waveform menu;

Input waveform choice and determine number of component hamonics;

UNTIL valid choice is made;

Clear terminal screen (ClearScreen);

Input required frequency range from user and determine initial test waveform period and number of test repititions (TestFreq);

Input required output waveform magnitude (RAN);

Select waveform required;

```
FOR each test repetition DO

      Determine register value for timer on DAC board (OP);

      Set waveform output range, set timer on DAC board and
      initalise test (INIT);

      Read settings of autorange circuitry after test;

      Load relavent complex real array with captured integer
      values;

END;

FOR each test repetition DO

      Calculate fourier transform of captured data (FFT);

      Calculate polar rotation (ANG);

      Initialise array with spectral estimates of output waveform
      (SpecEst);

      Calculate the transfer function of the system under test (TF);

END SystemTest.
```

MODULE tfreq:s

FUNCTION - To input frequency range from user and determine the initial test wave form period and the required number of test repetitions;

EXPORTS TestFreq;

FROM VDU2:s IMPORTS ClearScreen;

BEGIN TestFreq.

REPEAT

    Clear terminal screen (ClearScreen);

    Display initial test frequency menu;

    Input choice of inital test frequency;

UNTIL valid choice is made;

Clear terminal screen (ClearScreen);

From initial test frequency selected determine max test frequency possible;

REPEAT

    Clear terminal screen (ClearScreen);

    Display final frequency menu;

    Input choice of final test frequency;

UNTIL valid choice is made

END TestFreq.

MODULE - TRANSFER tranfn:s

FUNCTION - Calculates the transfer function of the system under test from the array of spectral estimates of captured data and array of spectral estimates of test waveform. Saves result for future analysis;

EXPORTS TF;

BEGIN TF.

FOR J = 1 TO 128 DO

> Calculate the modulus of the transfer function (no units);
>
> Calculate the phase of the transfer function (degrees);
>
> Calculate the gain in dB's;
>
> Display the frequency, phase and modulus of the transfer function;
>
> Store results in array for subsequent analysis;

END TF.

MODULE - angmag:s

FUNCTION - Converts the array of spectral estimates passed from cartesian to polar notation - result returned in passed array. Real array -> magnitude. Imaginary array -> angle;

EXPORTS ANG;

BEGIN ANG.

FOR count = 1 to 256 DO

Calculate the magnitude of the FFT from the cartesian co-ordinates;

Calculate the phase of the FFT from the cartesian co-ordinates;

Load value of magnitude into real array;

Load value of angle into imaginary array;

END ANG.

MODULE - ft:s

FUNCTION - Calculates the fourier transform of the real complex array of captured data and returns result in same array;

EXPORTS FFT;

BEGIN FFT.

Calculate fourier transform of the passed array using a 256 point radix-2 Cooley Tukey Algerithm;

END FFT;

MODULE - range:s

FUNCTION - Sets waveform output range, test waveform, and timer on DAC board. Initialises a frequency response test, waits for completion and loads an integer array with captured data;

EXPORTS INIT;

BEGIN INIT.

Set output voltage range and select test waveform;

Sets timer registers to give correct test frequency range;

Start frequency response test;

WHILE conversion not complete DO nothing;

Load integer array with captured data;

END INIT;

MODULE - ginn:s

FUNCTION - Contains two proceedures.

Proceedure RAN - Obtains the required output waveform magnitude from the user.

OP - Determines the register value for timer on DAC board from initial test frequency.

EXPORTS RAN,OP;

FROM VDU:s IMPORTS ClearScreen;

BEGIN RAN.

REPEAT

Clear terminal screen (ClearScreen);

Display menu of avaiable output ranges for test waveform;

Input choice of output range;

UNTIL Choice is valid;

END RAN.

BEGIN OP

Calculate required register value for timer on DAC board using initial test frequency required and system clock rate;

END OP.

MODULE - specest:s

FUNCTION - Loads array with spectral estimates of the output test waveform;

EXPORTS SpecEst;

BEGIN SpecEst.

CASE waveform OF

Prime: Load array with Prime MFTS estimates

PRBS: Load array with PRBS estimates

END SpecEst.

MODULE - specest:s

FUNCTION - Loads array with spectral estimates of the output test waveform;

EXPORTS SpecEst;

BEGIN SpecEst.

CASE waveform OF

Prime: Load array with Prime MFTS estimates

PRBS: Load array with PRBS estimates

END SpecEst.

Terminal Driver Modules

MODULE - rewrit:s

FUNCTION - Initialises SCN 2661 USART and sets up output buffer;

EXPORTS Q8RWRI;

BEGIN Q8RWRI.

Initialise USART Async.comm., no panty, 8 bits,

1 stopbit, internal clock, 300 baud,

NRTS low, reset error flags, NOTR low;

Initialise file output buffer in FCB;

END Q8RWRI.

MODULE - reset:s

FUNCTION - Initialise output file buffer, put <SP> into input file buffer, sets EOLN in FCB;

EXPORTS Q8RSET;

BEGIN Q8RSET.

Initialise input buffer;

Put <SP> into input file buffer;

Set EOLN in FCB;

END Q8RSET.

MODULE - get:s

FUNCTION - Loads file input, buffer with received character;

EXPORTS Q8GET;

FROM put:s IMPORTS Q8PUT;

BEGIN

WHILE USART receive buffer empty DO nothing;

Input character;

IF input character not <CR> THEN

Clear EOLN in FCB;

Check valid character if invalid input character = null; Echo character (Q8PUT);

ELSE

Set EOLN in FCB;

Set file input buffer to <SP>;

Echo <CR>, <LF> (Q8PUT);

END Q8GET.

MODULE writln:s

FUNCTION - Outputs <CR> <LF> to terminal;

EXPORTS Q8WRLN;

BEGIN Q8WRLN.

WHILE transmit buffer of USART not empty DO nothing;

Place <CR> in transmit buffer

WHILE transmit buffer of USART not empty DO nothing;

Place <LF> in transmit buffer.

END Q8WRLN.

MODULE Put:s

FUNCTION - Loads transmit buffer of USART with output character;

EXPORTS Q8PUT;

BEGIN Q8PUT.

WHILE transmit buffer full DO nothing;

Load USART transmit buffer with contents of output file buffer;

END Q8PUT.

APPENDIX D

Software Listings

```
PROGRAM FRIDFT (I,O) ;

(*****************************************************************************)
(*                                                                           *)
(*      MODULE - fridft:s                                                    *)
(*                                                                           *)
(*      FUNCTION - Provides user access to all system facilities via         *)
(*                 main menu                                                 *)
(*                                                                           *)
(*      FROM - VDU2      IMPORT - CearScreen,IntroOnScreen,Displaymenu       *)
(*                                ExitMessage,ErrorStatement                 *)
(*                                                                           *)
(*      FROM - ftest:s   IMPORT - SystemTest                                 *)
(*                                                                           *)
(*      FROM - bode:s    IMPORT - BodePrint                                  *)
(*                                                                           *)
(*      AUTHOR - Lee Jones                                                   *)
(*                                                                           *)
(*                                                                           *)
(*                                                                           *)
(*****************************************************************************)
CONST
        SIZE = 256 ;
TYPE
        storage = RECORD
                        Frequency : Real ;
                        Magnitude : Real ;
                        Phase : Real ;
                        DecibelMag : Real ;
                    END ; (* End of Record *)
        (* Storage array for post-mortem analysis and graph printing *)
        rdata = ARRAY [1..128] OF storage ;

VAR
        I,O : TEXT ;
        Exit : BOOLEAN ;
        Keyint,AryNumber,HarmNumber : INTEGER ;
        Measureflag : BOOLEAN ;
        (* These arrays used to store all the transfer function results *)
        Res1,Res2,Res3 : rdata ;


PROCEDURE ClearScreen ; EXTERNAL ;
PROCEDURE IntroOnScreen ; EXTERNAL ;
PROCEDURE DisplayMenu ; EXTERNAL ;
PROCEDURE SystemTest (VAR Measureflag:BOOLEAN;VAR NumOfAry,HarmNumber:INTEGER;
                        VAR Res1,Res2,Res3 : rdata ) ; EXTERNAL ;
PROCEDURE BodePrint ( Measureflag:BOOLEAN; AryNumber,HarmNumber : INTEGER;
                        VAR Res1,Res2,Res3 : rdata ) ; EXTERNAL ;
PROCEDURE ErrorStatement ; EXTERNAL ;
PROCEDURE ExitMessage ; EXTERNAL ;

BEGIN (* fridft *)
        (* Initialisation *)
        Exit := false ;
        Keyint := 0 ;
```

```
        Measureflag := false ;

        Reset (I) ;
        Rewrite (O) ;

        ClearScreen ;
        IntroOnScreen ;
        ClearScreen ;

        WHILE NOT Exit DO
        BEGIN (* Exit wait Loop *)
                DisplayMenu ;
                ReadLn(I);Read(I,Keyint) ;
                IF ( Keyint>0 ) AND ( Keyint<4 )
                THEN
                BEGIN (* Integer valid *)
                        CASE Keyint OF
                                1: SystemTest ( Measureflag,AryNumber,
                                                HarmNumber,Res1,Res2,Res3 ) ;
                                2: BodePrint ( Measureflag,AryNumber,
                                               HarmNumber,Res1,Res2,Res3 ) ;
                                3: Exit := true ;
                        END ; (* Case statement *)
                        ClearScreen ;
                END (* Integer valid *)
                ELSE
                BEGIN (* Integer invalid *)
                        ClearScreen ;
                        ErrorStatement ;
                END ; (* Integer invalid *)
        END ; (* Exit wait Loop *)

        ExitMessage ;

END. (* fridft *)
```

```
(*****************************************************************************)
(*                                                                           *)
(*      MODULE - ftest.                                                      *)
(*                                                                           *)
(*      FUNCTION - Obtain the transfer function of the System                *)
(*                 Under Test.                                               *)
(*                                                                           *)
(*      EXPORT - SystemTest.                                                 *)
(*                                                                           *)
(*      FROM - pause:s  IMPORT - PAUSE.                                      *)
(*                                                                           *)
(*      FROM - VDU2:s   IMPORT - ClearScreen.                                *)
(*                                                                           *)
(*      FROM - tfreq:s  IMPORT - TestFreq.                                   *)
(*                                                                           *)
(*      FROM - Ginn:s   IMPORT - RAN,OP.                                     *)
(*                                                                           *)
(*      FROM - range:s  IMPORT - INIT.                                       *)
(*                                                                           *)
(*      FROM - ft:s     IMPORT - FFT.                                        *)
(*                                                                           *)
(*      FROM - angmag:s IMPORT - ANG.                                        *)
(*                                                                           *)
(*      FROM - tranfn:s IMPORT - TF.                                         *)
(*                                                                           *)
(*      FROM - spcest:s IMPORT - SpecEst                                     *)
(*                                                                           *)
(*      AUTHOR - Lee Jones                                                   *)
(*                                                                           *)
(*****************************************************************************)
UNIT ftest ( SystemTest,INDATA,STAV,RANGE ) ;

CONST
        N = 8 ;
        SIZE = 256 ;
        PI = 3.141592653589793 ;

TYPE
        dat = ARRAY [1..SIZE] OF INTEGER ;
        storage = RECORD
                        Frequency : Real ;
                        Magnitude : Real ;
                        Phase : Real ;
                        DecibelMag : Real ;
                   END ; (* End of Record *)
        (* Storage array for post-mortem analysis and graph printing *)
        rdata = ARRAY [1..128] OF storage ;

VAR
        INDATA : dat ; (* Shared with Assembly routine IORUN *)
        STAV : INTEGER; (*Shared with Assembly routine IORUN *)
                        (*ADC status reg. mapped in Links*)
        RANGE : INTEGER; (* Shared with RAN procedure *)

PROCEDURE SystemTest (VAR MeasureFlag:BOOLEAN;VAR NumOfAry,HarmNumber:INTEGER;
                        VAR Res1,Res2,Res3 : rdata ) ;
```

```
CONST
        ECLOCK = 800000;

TYPE
        (* Forming the array to store system o/p data *)
        data = ARRAY [1..SIZE] OF REAL ;
        (* Forming the array to store the dft of the test signal  *)
        WaveData = ARRAY [1..128] of real;
VAR
        R1EALDT,I1MAGE,R2EALDT,I2MAGE,R3EALDT,I3MAGE : data ;
        MODWAVE,ANGWAVE : WaveData ;
        SysClk,FundMax,FundFreq,Fund1,Fund2,Fund3 : REAL ;
        Count,AryCount,AryNumber,IntNumber,INDEX : INTEGER ;
        FreqTemp : REAL ;
        ReplyOK : BOOLEAN ;
        AUTOSCALE : REAL ;
        I,O:TEXT;
        WaveOp : INTEGER;
PROCEDURE PAUSE;EXTERNAL ;
PROCEDURE ClearScreen ; EXTERNAL ;
PROCEDURE TestFreq ( FundMax:REAL; HarmNumber:INTEGER;VAR AryNumber : INTEGER;
                     VAR FundFreq : REAL ); EXTERNAL;
PROCEDURE RAN; EXTERNAL;
PROCEDURE INIT; EXTERNAL;
PROCEDURE OP(SysClk,FREQval:real); EXTERNAL;
PROCEDURE FFT (N,SIZE:INTEGER ; VAR REALDT,IMAGE : data ) ; EXTERNAL ;
PROCEDURE ANG (SIZE:INTEGER ; VAR REALDT,IMAGE : data ) ; EXTERNAL ;
PROCEDURE TF ( VAR REALDT,IMAGE:data ; MODWAVE,ANGWAVE:WaveData ;AUTORANGE,
                FREQ:REAL ;VAR Result:rdata;HarmNumber:INTEGER) ; EXTERNAL;
PROCEDURE SpecEst (VAR MODWAVE,ANGWAVE:WaveData;VAR WaveOp:INTEGER);EXTERNAL ;

(* Procedure AUTORANGE converts autorange reading from adc to appropriate *)
(* scalling value.                                                         *)
PROCEDURE AUTORANGE (VAR AUTOSCALE : REAL );

BEGIN
        CASE STAV OF
                1: AUTOSCALE := 64;
                3: AUTOSCALE := 16;
                5: AUTOSCALE := 8;
                7: AUTOSCALE := 4;
                9: AUTOSCALE := 2;
                11: AUTOSCALE := 1.6;
                13: AUTOSCALE := 1.333333;
                15: AUTOSCALE := 1;
        END
END;

BEGIN (* SystemTest *)

        REWRITE(O);
        RESET(I);

        (* Initialisation *)
        ReplyOK := TRUE ;
```

```
SysClk := ECLOCK;
Count := 1 ;
FundMax := 100000 / 128 ;
Fund1 := 0.0 ;
Fund2 := 0.0 ;
Fund3 := 0.0 ;

(* Input waveform choice and determine no. of component harmonics *)
REPEAT
        ReplyOk := TRUE;
        ClearScreen;
        WRITELN (0,'          TEST WAVEFORM MENU          ');
        WRITELN (0);
        WRITELN (0,' 1) Prime M.F.T.S.');
        WRITELN (0,' 2) P.R.B.S ');
        WRITELN (0);
        WRITE (0,' PLEASE SELECT OPTION (1/2)');
        READLN(I);READ(I,WaveOp);
        CASE waveOp OF
                1: HarmNumber := 73;
                2: HarmNumber := 128
        OTHERWISE
                ReplyOk := FALSE
        END
UNTIL ReplyOk = TRUE;

ClearScreen;

(* Input req. freq. range from user and determine initial test *)
(* waveform period and no. of test repetitions. *)
TestFreq ( FundMax,HarmNumber,AryNumber,FundFreq );

ClearScreen ;

FreqTemp := FundFreq ; (* Used as a temporary frequency count *)

(* Input req. output waveform magnitude. *)
RAN;

(* Select waveform req. *)
IF WaveOp = 2 THEN
RANGE := RANGE + 8;

FOR AryCount := 1 TO AryNumber DO
    BEGIN (* Data transfer *)

    (* Determine register value for timer on DAC board *)
    OP(SysClk,FreqTemp);

    WRITELN(0,'******************************************');
    WRITELN(0,'******* SYSTEM UNDER TEST ! *************');
    WRITELN(0,'******************************************');

    (* Clear INDATA byte array *)
    FOR Count := 1 TO SIZE DO
        INDATA[Count] := 0 ;
```

```
    (* Set waveform output range,set timer on DAC board and *)
    (* initialise test. *)
    INIT;

    (* Read settings of autorange circuitry after test. *)
    AUTORANGE ( AUTOSCALE );

    CASE AryCount OF

         1: BEGIN (* AryCount = 1 *)
                FOR Count := 1 TO SIZE DO
                   BEGIN
                        R1EALDT [ Count ] := INDATA [ Count ] ;
                        I1MAGE [ Count ] := 0.0 ;
                   END ;
                 Fund1 := FreqTemp ; (* Set Fundamental frequency *)
            END (* AryCount = 1 *);

         2: BEGIN (* AryCount = 2 *)
                FOR Count := 1 TO SIZE DO
                   BEGIN
                        R2EALDT [ Count ] := INDATA [ Count ] ;
                        I2MAGE [ Count ] := 0.0 ;
                   END ;
                Fund2 := FreqTemp ; (* Set Fundamental frequency *)
            END (* AryCount = 2 *);

         3: BEGIN (* AryCount = 3 *)
                FOR Count := 1 TO SIZE DO
                   BEGIN
                        R3EALDT [ Count ] := INDATA [ Count ] ;
                        I3MAGE [ Count ] := 0.0 ;
                   END ;
                Fund3 := FreqTemp ; (* Set Fundamental frequency*)
           END ; (* AryCount = 3 *)

    END; (* CASE *)

    FreqTemp := FreqTemp * HarmNumber ;

END ; (* FOR *)

(* Calculate transfer function and store results *)
FOR AryCount := 1 TO AryNumber DO
  BEGIN
    CASE AryCount OF

         1: BEGIN
               FFT (N,SIZE,R1EALDT,I1MAGE );
               ANG (SIZE,R1EALDT,I1MAGE ) ;
               SpecEst(MODWAVE,ANGWAVE,WaveOp) ;
               TF(R1EALDT,I1MAGE,MODWAVE,ANGWAVE,AUTOSCALE,Fund1,Res1,
                                                             HarmNumber)

            END;
```

```
            2: BEGIN (* AryCount = 2 *)
                  FFT (N,SIZE,R2EALDT,I2MAGE );
                  ANG (SIZE,R2EALDT,I2MAGE ) ;
                  SpecEst(MODWAVE,ANGWAVE,WaveOp) ;
                  TF(R2EALDT,I2MAGE,MODWAVE,ANGWAVE,AUTOSCALE,Fund2,Res2,
                                                            HarmNumber)
               END;

            3: BEGIN (* AryCount = 3 *)
                  FFT (N,SIZE,R3EALDT,I3MAGE );
                  ANG (SIZE,R3EALDT,I3MAGE ) ;
                  SpecEst(MODWAVE,ANGWAVE,WaveOp) ;
                  TF(R2EALDT,I2MAGE,MODWAVE,ANGWAVE,AUTOSCALE,Fund3,Res3,
                                                            HarmNumber)
              END;
         END ;(* CASE *)

         Measureflag := true ; (* Measurements made from S.U.T. *)
      END; (* FOR *)
      NumOfAry := AryNumber;
END ;. (* SystemTest *)
```

```
(***************************************************************************)
(*                                                                         *)
(*      MODULE - tfreq                                                     *)
(*                                                                         *)
(*      FUNCTION - To input required frequency range from user and         *)
(*                 determine from this the initial test vaveform period    *)
(*                 and the required number of test repititions.            *)
(*                                                                         *)
(*      EXPORT - TestFreq                                                  *)
(*                                                                         *)
(*      FROM - VDU2      IMPORT - ClearScreen                              *)
(*                                                                         *)
(*      AUTHOR - Lee Jones                                                 *)
(*                                                                         *)
(***************************************************************************)
UNIT SYSTUP ( I,O,TestFreq );

VAR I,O : TEXT;

PROCEDURE TestFreq( FundMax : real ;HarmNumber : INTEGER;
                    VAR AryNumber: INTEGER; VAR FundFreq :REAL);
CONST
        F0MIN = 0.01;
        F1MIN = 0.05;
        F2MIN = 0.1;
        F3MIN = 0.5;
        F4MIN = 1.0;
        F5MIN = 5.0;
        F6MIN = 10.0;
        F7MIN = 100.0;
        F8MIN = 500.0;
        (* F9MIN = FundMax *)
VAR
        Index,Keyint : INTEGER ;
        FreqTemp : REAL;
        ReplyOK : BOOLEAN ;

PROCEDURE ClearScreen ; EXTERNAL;

BEGIN (* TempInput *)
   Reset (I) ;
   Rewrite (O) ;
   REPEAT
        ReplyOk := TRUE;
        ClearScreen;
        Writeln (O) ;
        WRITELN(O,'0)   FundFreq =  ',F0MIN:9:2 );
        Writeln(O,'1)   FundFreq =  ',F1MIN:9:2 );
        Writeln(O,'2)   FundFreq =  ',F2MIN:9:2 );
        Writeln(O,'3)   FundFreq =  ',F3MIN:9:2 );
        Writeln(O,'4)   FundFreq =  ',F4MIN:9:2 );
        Writeln(O,'5)   FundFreq =  ',F5MIN:9:2 );
        WRITELN(O,'6)   FundFreq =  ',F6MIN:9:2 );
        Writeln(O,'7)   FundFreq =  ',F7MIN:9:2 );
        Writeln(O,'8)   FundFreq =  ',F8MIN:9:2 );
```

```
      WRITELN(O,'9)   FundFreq = ',FundMax:9:2 );
      Writeln(O,'Which init. frequency do you require  [1,2,3,4 or 5]  ?');
      Readln(I); Read (I,Keyint);
      CASE KeyInt OF
              0: FundFreq := F0MIN;
              1: FundFreq := F1MIN;
              2: FundFreq := F2MIN;
              3: FundFreq := F3MIN;
              4: FundFreq := F4MIN;
              5: FundFreq := F5MIN;
              6: FundFreq := F6MIN;
              7: FundFreq := F7MIN;
              8: FundFreq := F8MIN;
              9: FundFreq := FundMax
      OTHERWISE
              ReplyOk := FALSE
      END (* CASE end *)
   UNTIL ReplyOk = TRUE;


   REPEAT
      ReplyOk := TRUE;
      ClearScreen;
      FreqTemp := FundFreq;
      Index := 1;
      WHILE (FreqTemp <= FundMax) AND  (Index < 4 ) DO
              BEGIN
                      FreqTemp := FreqTemp * HarmNumber;
                      WRITELN (O);
                      WRITELN (O,INDEX,') Freqmax = ',FreqTemp);
                      WRITELN (O);
                      INDEX := INDEX+1
              END;
      WRITELN (O,' SELECT MAX. FREQUENCY ');
      READLN(I);READ (I,KeyInt);
      IF KeyInt < INDEX THEN
        CASE KeyInt OF
              1: AryNumber := 1;
              2: AryNumber := 2;
              3: AryNumber := 3
         END
      ELSE
              ReplyOk := FALSE
   UNTIL ReplyOk = TRUE
END;.
```

```
(*************************************************************************
(*                                                                       *
(*      MODULE - tranfn:s                                                *
(*                                                                       *
(*      FUNCTION - Calculates the transfer function of the system under  *
(*                 test from the array of spectral estimates of capture  *
(*                 data and array of spectral estimates of test waveform.*
(*                 saves result for future analysis.                     *
(*                                                                       *
(*      EXPORT - TF                                                      *
(*                                                                       *
(*      AUTHOR - Lee Jones                                               *
(*                                                                       *
(*                                                                       *
(*************************************************************************

UNIT tranfn (TF,RANGE);
CONST
        (* Used to prevent overflow when using natural logorithms *)
        MINREAL = 0.000000000000000001 ; (*   1 X 10-18   *)
TYPE
   (* Defining the arrays passed to this procedure from the main program *)
   data = ARRAY[1..256] OF REAL;
   mdata = ARRAY[1..128] OF REAL;
   storage = RECORD
                Frequency : REAL ;
                Magnitude : REAL ;
                Phase     : REAL ;
                Dbmag     : REAL ;
             END ; (* End of RECORD *)
   rdata = ARRAY[1..128] OF storage ;
VAR
        RANGE : INTEGER;

(* Defining the procedure name and all parameters passed to it *)
PROCEDURE TF (VAR REALDT,IMAGE:data; MODIP,ANGIP:mdata;AUTOSCALE,FREQ:REAL;
                              VAR Result:rdata;HarmNumber:INTEGER);
(* Result used to store final results for post-mortem analysis *)

TYPE
   (* Forming an array so that the freq. can be calc. and printed in a loop *)
   fdata = ARRAY[1..128] OF REAL;

VAR
   (* Defining the variables within this procedure *)
   FR : fdata;
   I,O : TEXT;
   CONT,N,X : INTEGER;
   SCALE,MAGN,PH,DECMAG : REAL;
(* Procedure begin *)
PROCEDURE PAUSE;EXTERNAL;
PROCEDURE ClearScreen;EXTERNAL;
BEGIN
   (* Setting up the LCD ready to be written to *)
   REWRITE(O);
   RESET(I);
```

```
FR[1] := 0.0; (* D.C. Term *)
FOR N:= 1 TO 127 DO
      FR[N+1]:= FREQ*N;
WRITELN(O);              (* Leaving a space *)
(* Writing to the LCD *)
WRITELN(O,'The spectral estimates of the transfer');
WRITELN(O,'function will shortly be displayed');
WRITELN(O);
WRITELN (O,  'FREQ(Hz)    MAGN(T.F.)    PH(T.F.)    ACTUAL MAGN (dB)'');

(* CASE statement to scale result to take account of output range. *)
CASE RANGE OF
               8,0: SCALE := 64;
               9,1: SCALE := 16;
               10,2: SCALE := 8;
               11,3: SCALE := 4;
               12,4: SCALE := 2;
               13,5: SCALE := 1.6;
               14,6: SCALE := 1.333333;
               15,7: SCALE := 1;
END;

(* For/do loop to calc the magnitude and phase of the T.F. at each  *)
(*  multiple of the fundamental frequency. *)
CASE HarmNumber OF  (* CASE of Prime Numbers *)
    73: BEGIN
           FOR N := 1 TO 128 DO
            BEGIN
               (* Calculating the modulus of the T.F. *)
               MAGN := REALDT[N]/MODIP[N] * SCALE/AUTOSCALE :
               (* Calculating the phase of the T.F. *)
               PH := IMAGE[N]-ANGIP[N]+(5.0E-6*FR[N]*360);
               (* Making all -ve angles into +ve angles *)
               IF PH < 0.0 THEN
                  PH := PH + 360;
               (* Calculate Gain Magnitude in dB's for display *)
               (* Prevent computer overflow on ln(0) and ln(1)/ln(10) *)
               IF MAGN < MINREAL
               THEN
                        MAGN := MINREAL;
               IF ln(MAGN) = 0.0
               THEN
                         DECMAG := 0.0
               ELSE
                         DECMAG := 20*(ln(MAGN)/ln(10.0));


               (* Store results for future post analysis *)
               Result[N].Frequency := FR[N] ;
               Result[N].Magnitude := MAGN ;
               Result[N].Phase := PH ;
               Result[N].Dbmag := DECMAG

           END;        (* End of for/do loop *)

           BEGIN
```

```
        FOR N := 1 TO 21 DO
        (* Convert index N to point at Prime Numbers  *)
         BEGIN
            CASE N OF
                     1:   X:=1;
                     2:   X:=4;
                     3:   X:=6;
                     4:   X:=8;
                     5:   X:=12;
                     6:   X:=14;
                     7:   X:=18;
                     8:   X:=20;
                     9:   X:=24;
                    10:   X:=30;
                    11:   X:=32;
                    12:   X:=38;
                    13:   X:=42;
                    14:   X:=44;
                    15:   X:=48;
                    16:   X:=54;
                    17:   X:=60;
                    18:   X:=62;
                    19:   X:=68;
                    20:   X:=72;
                    21:   X:=74
             END; (* END of conversion table *)
            CASE X OF
            (*  Paging of screen output  *)
                   74  :BEGIN
                          WRITELN(0,Result[X].Frequency:8:3,
                                    Result[X].Magnitude:10:3,
                                    Result[X].Phase:11:3,
                                    Result[X].Dbmag:18:3);
                          WRITE(0,'       Press Return to Continue');
                          READLN(I);
                          ClearScreen
                        END;
                 OTHERWISE:
                 (* output results to LCD screen *)
                  BEGIN
                     WRITELN(0,Result[X].Frequency:8:3,
                               Result[X].Magnitude:10:3,
                               Result[X].Phase:11:3,
                               Result[X].Dbmag:18:3)
                  END (* END of result table *)
             END (* END of CASE X *)
      END (* END of for/do loop *)
    END
END; (* END of CASE Prime Numbers *)
128: BEGIN
       FOR N := 1 TO 128 DO
        BEGIN

          (* Paging of screen output *)
          CASE N OF
              22,44,66,
```

```
                    88,110 : BEGIN
                                 WRITE(0,'      Press Return to Continue');
                                 READLN(I);
                                 ClearScreen;
                                 WRITELN(0,'FREQ(Hz)    MAGN(T.F.)    PH(T.F.)    A
CTUAL MAGN (db)')
                              END;
               (*      128 : BEGIN
                                 WRITELN(0);
                                 WRITELN(0);
                                 WRITELN(0);
                                 WRITELN(0);
                                 WRITE(0,'      Press Return to Continue');
                                 READLN(I);
                                 ClearScreen
                              END; *)
              OTHERWISE:
            END;
            (* Calculating the modulus of the T.F. *)
            MAGN := REALDT[N]/MODIP[N] * SCALE/AUTOSCALE ;
            (* Calculating the phase of the T.F. *)
            PH := IMAGE[N]-ANGIP[N]+(5.0E-6*FR[N]*360);
            (* Making all -ve angles into +ve angles *)
            IF PH < 0.0 THEN
               PH := PH + 360;
            (* Calculate Gain Magnitude in dB's for display *)
            (* Prevent computer overflow on ln(0) and ln(1)/ln(10) *)
            IF MAGN < MINREAL
            THEN
                   MAGN := MINREAL;
            IF ln(MAGN) = 0.0
            THEN
                   DECMAG := 0.0
            ELSE
                   DECMAG := 20*(ln(MAGN)/ln(10.0));

           (* Writing the freq., magnitude, phase and decibel GAIN of the T.F.
 *)
           WRITELN (0,FR[N]:8:3,MAGN:10:3,PH:11:3,DECMAG:18:3);

           (* Store results for future post analysis *)
           Result[N].Frequency := FR[N] ;
           Result[N].Magnitude := MAGN ;
           Result[N].Phase := PH ;
           Result[N].Dbmag := DECMAG

         END;        (* End of for/do loop *)
         WRITELN(0);
         WRITE(0,'      Press Return to Continue');
         READLN(I);
         ClearScreen
    END
  END
END;.    (*End of procedure *)
```

```
(***************************************************************************)
(*                                                                         *)
(*      MODULE - angmag.                                                   *)
(*                                                                         *)
(*      FUNCTION - Converts the array of spectral estimates passed         *)
(*                 from cartesian to polar notation. Results returned      *)
(*                 in passed array - Real array = magnitude                *)
(*                                Imaginary array = angle                  *)
(*                                                                         *)
(*      EXPORT - ANG                                                       *)
(*                                                                         *)
(*      AUTHOR - Lee Jones                                                 *)
(*                                                                         *)
(***************************************************************************)
UNIT AGL (ANG);
TYPE
   data = ARRAY[1..256] OF REAL;
(* Defining the procedure and any parameters passed to it *)
PROCEDURE ANG (SIZE:INTEGER;VAR REALDT,IMAGE:data);
(* Defining procedure constants *)
CONST
        PI = 3.141592653589793;

VAR
        O : TEXT;
        COUNT:INTEGER;
        MAGFFT,TANG,ANGLE,Z,MAG,Harmonic : REAL;

(***************************************************************************)
(*                                                                         *)
(* This function calcs. the arctan of the input variable                   *)
(*                                                                         *)
(***************************************************************************)
FUNCTION LOG( X:REAL):REAL;
VAR     LOG1:REAL;
BEGIN
        IF ABS(X) > 1 THEN
        BEGIN
                LOG1:=-1;
                REPEAT
                    X:=ABS(X)/10;
                    LOG1:=LOG1+1;
                UNTIL X<1;
                LOG:=LOG1
        END
        ELSE
                IF X<>0 THEN
                BEGIN
                    LOG1:=0;
                    REPEAT
                        X:=ABS(X)*10;
                        LOG1:=LOG1+1;
                    UNTIL X>1;
                    LOG:=-LOG1
                END
                ELSE
```

```
                    LOG:= -20
END;(* Function LOG *)

(****************************************************************************)
(*                                                                          *)
(* Function to calculate the arctangent of input variable.                  *)
(*                                                                          *)
(****************************************************************************)
FUNCTION atangent (Z : REAL):REAL;

CONST
        PI = 3.141592653589793;

VAR
        POWER,P,CNT:INTEGER;
        SUM,M,W:REAL;

BEGIN    (* Function start *)
        SUM := 0.0;
        P := -1;
        POWER :=1;
        WHILE POWER < 6  DO
        BEGIN
            CNT :=1;
            IF Z > -1 THEN
            BEGIN
                IF Z > 1 THEN
                BEGIN
                    M := 1/Z;
                    W := 1/Z;
                END
                ELSE
                BEGIN
                    M :=Z;
                    W :=Z;
                END;
            END
            ELSE
            BEGIN
                M := 1/Z;
                W := 1/Z;
            END;
            WHILE CNT < POWER DO
            BEGIN
              M :=M*W;
              CNT :=CNT+1;
            END;
            M := M/POWER;
            P := -1*P;
            IF P > 0 THEN
                SUM :=SUM+M
            ELSE
                SUM :=SUM-M;
            POWER :=POWER+2;
        END;
        IF Z < 1 THEN
```

```
        BEGIN
            IF Z < -1 THEN
                atangent := -PI/2 - SUM
            ELSE
                atangent := SUM;
        END
        ELSE
        atangent := PI/2 - SUM;
END;    (* of function atangent *)


BEGIN  (* Procedure start *)
     REWRITE(O);
     (* Loop to calc. the magnitude and phase of the 1st *)
     (* 128 harmonics of the DFT of the system o/p data          *)
     FOR COUNT := 1 TO 128 DO
        BEGIN
          (* Calculating the magnitude *)
          MAG := SQR(REALDT[COUNT])+SQR(IMAGE[COUNT]);
          IF MAG = 1.0 THEN
             MAGFFT := MAG
          ELSE
             MAGFFT:= SQRT(100*MAG)/10;
          (* Calculating the phase *)
          IF REALDT[COUNT] <> 0.0 THEN
            BEGIN
              IF (( LOG(IMAGE[COUNT])-LOG(REALDT[COUNT]))> 6) THEN
              BEGIN
                TANG:=-90;
                IF ( REALDT[COUNT]>0) AND (IMAGE[COUNT]>0) THEN
                TANG:=90;
                IF (REALDT[COUNT]<0) AND (IMAGE[COUNT]<0) THEN
                TANG:=90
              END;
              IF ((LOG(IMAGE[COUNT])-LOG(REALDT[COUNT]))<-6) THEN
              TANG:=0;
              IF ((TANG<>0) OR (ABS(TANG)<>90)) THEN
              BEGIN
                 Z := IMAGE[COUNT]/REALDT[COUNT];
                         TANG := atangent (Z);
                 TANG:= TANG*180/PI;
              END;
              IF IMAGE[COUNT] <> 0.0 THEN
                (* This loop calcs angles in any of the 4 quadrants *)
                BEGIN
                  IF REALDT[COUNT] < 0.0 THEN
                    ANGLE := TANG +180
                  ELSE
                    ANGLE := TANG
                END
              ELSE
                (* This loop determines angle on real axis- 0 or -180*)
                BEGIN
                  IF REALDT[COUNT] <0.0 THEN
                    ANGLE := -180
                  ELSE
```

```
                ANGLE := 0.0
            END;
        END
      (* This loop determines angle on imag. axis ie, 90 or 270 *)
      ELSE
        BEGIN
          IF IMAGE[COUNT] < 0.0 THEN
            ANGLE := 270
          ELSE
            BEGIN
              IF IMAGE[COUNT] > 0.0 THEN
                ANGLE := 90
              ELSE
                ANGLE := 0.0
            END
        END;
    IF ANGLE > 180 THEN
       ANGLE := ANGLE-360;

    (* Loading the magnitude and angle into the real and imag. *)
    (* arrays respectively.                                     *)
    REALDT[COUNT] := MAGFFT;
    IMAGE[COUNT] := ANGLE;
    Harmonic := ( COUNT - 1 ) ;
    END;
END;.    (* End of procedure *)
```

```
(*************************************************************************)
(*                                                                       *)
(*      MODULE - ft                                                      *)
(*                                                                       *)
(*      FUNCTION - Calculates the Fourier Transform of the complex       *)
(*                 real array of captured data. Returns results in       *)
(*                 passed array                                          *)
(*                                                                       *)
(*      EXPORT - FFT                                                     *)
(*                                                                       *)
(*      AUTHOR - Lee Jones       LASTUPDATE - 5.1.88.                    *)
(*                                                                       *)
(*************************************************************************)
   UNIT FT (FFT);
   TYPE
        data = ARRAY[1..256] OF REAL;
   (* Stating procedure name and any parameters passed to it *)
   PROCEDURE FFT (N,SIZE:INTEGER;VAR REALDT,IMAGE:data);
        LABEL 1;
        (* Definition of procedure constants *)
        CONST
              PI = 3.14156265389793;
              INVAL = 2;
        VAR
            O : TEXT;
            UREAL,UIMAG,WREAL,WIMAG,TREAL,TIMAG : REAL;
            UR,UI,X : REAL;
            NBD2,NBM1,COUNT,COUNT2,COUNT3,KTEMP,ME,LPK,DUMMY :INTEGER;
            VAL : INTEGER;
        SCONST
              COSINE: ARRAY[1..8] OF REAL =
              (-1.000000E+00,0.0,7.0710677E-01,9.2387950E-01,
                9.8078531E-01,9.9518472E-01,9.9879545E-01,9.9969882E-01);

              SINE: ARRAY[1..8] OF REAL =
              (0.0,1.0,7.0710677E-01,3.8268343E-01,
                1.9509032E-01,9.8017141E-02,4.9067676E-02,2.4541229E-02);

        (* Definition of procedure variables *)
        (* Declaration of functions *)
        (* Function for exponential *)
        FUNCTION exponent (INVAL : INTEGER;COUNT3 : INTEGER):INTEGER;
        (* Raises number (INVAL) to positive integer power (COUNT3) *)
        VAR A,ANSWER : INTEGER;
        BEGIN
            ANSWER:=1;
            A:=0;
            WHILE (A<COUNT3) DO
            BEGIN
               ANSWER:= ANSWER*INVAL;
               A:=A+1;
            END;
            exponent:=ANSWER
        END;
        BEGIN
            REWRITE(O);
```

```
WRITELN (O);
WRITELN (O);
NBD2 := SIZE DIV 2;
NBM1 := SIZE-1;
COUNT := 1;
FOR COUNT2 := 1 TO NBM1 DO
   BEGIN  (* Start of the FOR/DO Loop *)
     IF COUNT2 < COUNT THEN
      BEGIN
           TREAL := REALDT [COUNT];
           TIMAG := IMAGE [COUNT];
           REALDT [COUNT] := REALDT [COUNT2];
           IMAGE [COUNT] := IMAGE [COUNT2];
           REALDT [COUNT2] := TREAL;
           IMAGE [COUNT2] := TIMAG;
     END;
     KTEMP := NBD2;
1:   IF KTEMP < COUNT THEN
       BEGIN
           COUNT := COUNT - KTEMP;
           KTEMP := KTEMP DIV 2;
           GOTO 1
       END
     ELSE
       COUNT := COUNT + KTEMP
   END;   (* End of FOR/DO Loop *)
FOR  COUNT3 := 1 TO N DO
   BEGIN
     UREAL := 1.0;
     UIMAG := 0.0;
     ME := exponent(INVAL,COUNT3);  (* Calls function exponent *)
     KTEMP := ME DIV 2;
     X := PI/KTEMP;
     (* a + jb = Cos x + j Sin x          *)
     WREAL := cosine [COUNT3];
     WIMAG := sine [COUNT3];
     WIMAG := -WIMAG;
     FOR COUNT := 1 TO KTEMP DO
        BEGIN    (*Start of 1st inner Loop*)
            VAL := COUNT;
            WHILE  (VAL <= SIZE) DO
                 BEGIN
                    COUNT2 := VAL;
                    LPK:= COUNT2 + KTEMP;
                    TREAL:=(REALDT[LPK]*UREAL)-(IMAGE[LPK]*UIMAG);
                    TIMAG:=(REALDT[LPK]*UIMAG)+(IMAGE[LPK]*UREAL);
                    REALDT[LPK]:=REALDT[COUNT2] - TREAL;
                    IMAGE[LPK]:=IMAGE[COUNT2] - TIMAG;
                    REALDT[COUNT2]:=REALDT[COUNT2] + TREAL;
                    IMAGE[COUNT2]:=IMAGE[COUNT2] + TIMAG;
                    VAL := VAL + ME;
                 END;
           UR := UREAL;
           UI := UIMAG;
           UREAL:=(UR*WREAL)-(UI*WIMAG);
           UIMAG:=(UR*WIMAG)+(UI*WREAL);
```

```
            END;   (*End of 1st inner Loop*)
        END; (*End of for/do Loop *)
END;.  (* End of proceedure fft *)
```

```
(****************************************************************************)
(*                                                                          *)
(*      MODULE - Ginn                                                       *)
(*                                                                          *)
(*      FUNCTION - Contains two procedures :                                *)
(*                 RAN - Obtains the required output waveform magnitude     *)
(*                       from the user.                                     *)
(*                                                                          *)
(*                 OP - Determines the register value for the timer on      *)
(*                      the DAC board from initial test frequency.          *)
(*                                                                          *)
(*      EXPORT - RAN,OP                                                     *)
(*                                                                          *)
(*      FROM -  VDU2    IMPORT - ClearScreen                                *)
(*                                                                          *)
(*      AUTHOR - Lee Jones                                                  *)
(*                                                                          *)
(****************************************************************************)
UNIT Ginn (R1,R2,R3,OP,RAN,RANGE,I,O);

VAR
        R1,R2,R3 : 0..65535;
        RANGE : INTEGER;
        I,O:TEXT;
        (* Values for above variables are passed to assembly *)
        (* routines that is why they are externally defined. *)


(* OP Procedure to calculate R1 - divider count for PTM from Fundreq *)
PROCEDURE OP(SysClk,Fundreq:real);
VAR
        SCF:REAL; (* Sample Clock Frequency *)
BEGIN
        REWRITE(O);
        SCF:= Fundreq * 512 ; (* 512 samples per waveform *)
        R1:= ROUND( SysClk / SCF );
        WRITELN (O,'Fundreq = ',Fundreq:6:2,'R1= ',R1);
END;


PROCEDURE RAN;
VAR
        VALID:BOOLEAN;
        RANOP:CHAR;
        I,O:TEXT;

 BEGIN
    REWRITE(O);
    RESET(I);
    REPEAT
        WRITELN(O);
        WRITE(O,'  Which one of the o/p ranges listed below do you  ');
        WRITELN(O,'require ?');
        WRITELN(O);
        WRITELN(O,'FOR O/P RANGE +/_ 0.125 Volts .....PRESS KEY 1');
        WRITELN(O,'FOR O/P RANGE +/_ 0.5 Volts........PRESS KEY 2');
```

```
    WRITELN(O,'FOR O/P RANGE +/_ 1.0 Volts........PRESS KEY 3');
    WRITELN(O,'FOR O/P RANGE +/_ 2.0 Volts........PRESS KEY 4');
    WRITELN(O,'FOR O/P RANGE +/_ 4.0 Volts........PRESS KEY 5');
    WRITELN(O,'FOR O/P RANGE +/_ 5.0 Volts........PRESS KEY 6');
    WRITELN(O,'FOR O/P RANGE +/_ 6.0 Volts........PRESS KEY 7');
    WRITELN(O,'FOR O/P RANGE +/_ 8.0 Volts........PRESS KEY 8');

    VALID:=TRUE;
    READLN(I);
    READ(I,RANOP);
    CASE  RANOP OF   (* To find I/P value from keyboard *)
            '1':RANGE:=0;
            '2':RANGE:=1;
            '3':RANGE:=2;
            '4':RANGE:=3;
            '5':RANGE:=4;
            '6':RANGE:=5;
            '7':RANGE:=6;
            '8':RANGE:=7
      OTHERWISE
            BEGIN    (*   ONLY FOR NOW *)
                VALID:=FALSE;
                WRITELN(O,'YOUR ENTERED VALUE IS WRONG. TRY AGAIN')
            END
    END;   (*  OF CASE *)
  UNTIL VALID=TRUE
END;. (* OF PROCEDURE RANGE *)
```

```
(******************************************************************************)
(*                                                                            *)
(*      MODULE - spcest                                                       *)
(*                                                                            *)
(*      FUNCTION - Loads array with spectral estimates of the output          *)
(*                 test waveform.                                             *)
(*                                                                            *)
(*      EXPORT - SpecEst                                                      *)
(*                                                                            *)
(*      AUTHOR - Lee Jones                                                    *)
(*                                                                            *)
(******************************************************************************)
UNIT spcest ( SpecEst );
TYPE
        WaveData = ARRAY[1..128] OF REAL;
PROCEDURE SpecEst (VAR MODWAVE,ANGWAVE:WaveData;VAR WaveOp:INTEGER);

BEGIN
    CASE WaveOp OF
        (* Initialise MODWAVE and ANGWAVE with spectral estimates of *)
        (* PRBS M.F.T.S. *)
        2:BEGIN
           MODWAVE[     1]:=  5.2825500E+0005;
           MODWAVE[     2]:=  3.4684495E+0004;
           MODWAVE[     3]:=  3.4350907E+0004;
           MODWAVE[     4]:=  3.4331704E+0004;
           MODWAVE[     5]:=  3.3282797E+0004;
           MODWAVE[     6]:=  3.2423360E+0004;
           MODWAVE[     7]:=  3.4737454E+0004;
           MODWAVE[     8]:=  3.3060733E+0004;
           MODWAVE[     9]:=  3.2646630E+0004;
           MODWAVE[    10]:=  3.1319513E+0004;
           MODWAVE[    11]:=  3.4423426E+0004;
           MODWAVE[    12]:=  3.1640803E+0004;
           MODWAVE[    13]:=  3.5160695E+0004;
           MODWAVE[    14]:=  3.2875236E+0004;
           MODWAVE[    15]:=  3.1182639E+0004;
           MODWAVE[    16]:=  3.3688952E+0004;
           MODWAVE[    17]:=  3.1622967E+0004;
           MODWAVE[    18]:=  3.4723017E+0004;
           MODWAVE[    19]:=  2.9593374E+0004;
           MODWAVE[    20]:=  3.6905814E+0004;
           MODWAVE[    21]:=  3.4173439E+0004;
           MODWAVE[    22]:=  3.1966315E+0004;
           MODWAVE[    23]:=  2.9202803E+0004;
           MODWAVE[    24]:=  2.7039093E+0004;
           MODWAVE[    25]:=  3.5919967E+0004;
           MODWAVE[    26]:=  3.6919428E+0004;
           MODWAVE[    27]:=  2.8118439E+0004;
           MODWAVE[    28]:=  3.4616414E+0004;
           MODWAVE[    29]:=  3.1771310E+0004;
           MODWAVE[    30]:=  3.5540752E+0004;
           MODWAVE[    31]:=  2.6425864E+0004;
           MODWAVE[    32]:=  3.9760247E+0004;
           MODWAVE[    33]:=  3.0156118E+0004;
```

```
MODWAVE[   34]:=   3.0191083E+0004;
MODWAVE[   35]:=   3.3019056E+0004;
MODWAVE[   36]:=   3.4150872E+0004;
MODWAVE[   37]:=   2.6895028E+0004;
MODWAVE[   38]:=   1.6232497E+0004;
MODWAVE[   39]:=   4.4672599E+0004;
MODWAVE[   40]:=   3.9847855E+0004;
MODWAVE[   41]:=   2.9453626E+0004;
MODWAVE[   42]:=   2.6857758E+0004;
MODWAVE[   43]:=   4.2226676E+0004;
MODWAVE[   44]:=   3.4225939E+0004;
MODWAVE[   45]:=   2.7119398E+0004;
MODWAVE[   46]:=   2.2410479E+0004;
MODWAVE[   47]:=   1.1175767E+0004;
MODWAVE[   48]:=   5.1202643E+0004;
MODWAVE[   49]:=   3.5445798E+0004;
MODWAVE[   50]:=   2.3367627E+0004;
MODWAVE[   51]:=   4.0096525E+0004;
MODWAVE[   52]:=   2.5235960E+0004;
MODWAVE[   53]:=   1.6629706E+0004;
MODWAVE[   54]:=   4.5319564E+0004;
MODWAVE[   55]:=   2.4859087E+0004;
MODWAVE[   56]:=   3.9949748E+0004;
MODWAVE[   57]:=   2.1133038E+0004;
MODWAVE[   58]:=   4.0616199E+0004;
MODWAVE[   59]:=   2.9286507E+0004;
MODWAVE[   60]:=   3.9710149E+0004;
MODWAVE[   61]:=   3.0763628E+0004;
MODWAVE[   62]:=   2.6018647E+0004;
MODWAVE[   63]:=   3.9297749E+0004;
MODWAVE[   64]:=   2.4695162E+0004;
MODWAVE[   65]:=   1.4764733E+0004;
MODWAVE[   66]:=   4.3991409E+0004;
MODWAVE[   67]:=   2.6812270E+0004;
MODWAVE[   68]:=   2.3410539E+0004;
MODWAVE[   69]:=   4.6731755E+0004;
MODWAVE[   70]:=   2.4852051E+0004;
MODWAVE[   71]:=   3.2721394E+0004;
MODWAVE[   72]:=   4.1877469E+0004;
MODWAVE[   73]:=   3.2004363E+0004;
MODWAVE[   74]:=   1.6305987E+0004;
MODWAVE[   75]:=   9.2342210E+0003;
MODWAVE[   76]:=   4.7069048E+0004;
MODWAVE[   77]:=   4.0926059E+0004;
MODWAVE[   78]:=   3.7847407E+0004;
MODWAVE[   79]:=   3.3272256E+0004;
MODWAVE[   80]:=   1.6138837E+0004;
MODWAVE[   81]:=   3.5107837E+0004;
MODWAVE[   82]:=   3.0110361E+0004;
MODWAVE[   83]:=   3.3700324E+0004;
MODWAVE[   84]:=   2.1506244E+0004;
MODWAVE[   85]:=   4.6657076E+0004;
MODWAVE[   86]:=   2.4360225E+0004;
MODWAVE[   87]:=   4.0528332E+0004;
MODWAVE[   88]:=   2.1925150E+0004;
MODWAVE[   89]:=   4.6389572E+0004;
```

```
MODWAVE[  90]:=  2.2164624E+0004;
MODWAVE[  91]:=  8.4461670E+0003;
MODWAVE[  92]:=  3.3200424E+0004;
MODWAVE[  93]:=  2.3883685E+0004;
MODWAVE[  94]:=  3.6685153E+0004;
MODWAVE[  95]:=  4.0853083E+0004;
MODWAVE[  96]:=  2.4446701E+0004;
MODWAVE[  97]:=  4.6287901E+0004;
MODWAVE[  98]:=  2.8013565E+0004;
MODWAVE[  99]:=  2.0538254E+0004;
MODWAVE[ 100]:=  3.6454995E+0004;
MODWAVE[ 101]:=  3.4666715E+0004;
MODWAVE[ 102]:=  2.8996229E+0004;
MODWAVE[ 103]:=  4.2421310E+0004;
MODWAVE[ 104]:=  2.0199135E+0004;
MODWAVE[ 105]:=  1.5269947E+0004;
MODWAVE[ 106]:=  5.3129816E+0004;
MODWAVE[ 107]:=  1.4951023E+0004;
MODWAVE[ 108]:=  4.7019141E+0004;
MODWAVE[ 109]:=  1.5461647E+0004;
MODWAVE[ 110]:=  5.0024109E+0004;
MODWAVE[ 111]:=  2.4298149E+0004;
MODWAVE[ 112]:=  1.9321421E+0004;
MODWAVE[ 113]:=  8.1274090E+0003;
MODWAVE[ 114]:=  4.5866547E+0004;
MODWAVE[ 115]:=  6.1542160E+0003;
MODWAVE[ 116]:=  3.0586388E+0004;
MODWAVE[ 117]:=  3.9494121E+0004;
MODWAVE[ 118]:=  4.6742820E+0004;
MODWAVE[ 119]:=  1.4241770E+0004;
MODWAVE[ 120]:=  2.6226146E+0004;
MODWAVE[ 121]:=  1.7284642E+0004;
MODWAVE[ 122]:=  4.7490572E+0004;
MODWAVE[ 123]:=  2.2991670E+0004;
MODWAVE[ 124]:=  3.6159028E+0004;
MODWAVE[ 125]:=  4.1270394E+0004;
MODWAVE[ 126]:=  1.3868756E+0004;
MODWAVE[ 127]:=  4.3570940E+0004;
MODWAVE[ 128]:=  3.6906588E+0004;
  ANGWAVE[     1]:=   0.0000000E+0000;
  ANGWAVE[     2]:=  -1.9180000E+0001;
  ANGWAVE[     3]:=  -1.7962000E+0001;
  ANGWAVE[     4]:=   4.5728000E+0001;
  ANGWAVE[     5]:=  -1.6820000E+0001;
  ANGWAVE[     6]:=   1.4691200E+0002;
  ANGWAVE[     7]:=   4.8627000E+0001;
  ANGWAVE[     8]:=   5.1890000E+0001;
  ANGWAVE[     9]:=  -1.7374000E+0001;
  ANGWAVE[    10]:=  -9.6235000E+0001;
  ANGWAVE[    11]:=   1.4914400E+0002;
  ANGWAVE[    12]:=   1.3501800E+0002;
  ANGWAVE[    13]:=   4.2329000E+0001;
  ANGWAVE[    14]:=  -7.4530000E+0000;
  ANGWAVE[    15]:=   6.4185000E+0001;
  ANGWAVE[    16]:=  -1.6746300E+0002;
  ANGWAVE[    17]:=  -9.1440000E+0000;
```

```
ANGWAVE[    18]:=  1.6376300E+0002;
ANGWAVE[    19]:= -8.5282000E+0001;
ANGWAVE[    20]:=  8.6369000E+0001;
ANGWAVE[    21]:=  1.5925800E+0002;
ANGWAVE[    22]:=  1.0143600E+0002;
ANGWAVE[    23]:=  1.5657000E+0002;
ANGWAVE[    24]:= -1.4468800E+0002;
ANGWAVE[    25]:=  4.0863000E+0001;
ANGWAVE[    26]:= -3.0905000E+0001;
ANGWAVE[    27]:= -2.4590000E+0000;
ANGWAVE[    28]:= -1.3613800E+0002;
ANGWAVE[    29]:=  7.3894000E+0001;
ANGWAVE[    30]:= -1.6057900E+0002;
ANGWAVE[    31]:= -1.4679600E+0002;
ANGWAVE[    32]:= -2.2641000E+0001;
ANGWAVE[    33]:= -3.9100000E-0001;
ANGWAVE[    34]:= -8.9491000E+0001;
ANGWAVE[    35]:=  1.6906400E+0002;
ANGWAVE[    36]:= -1.8945000E+0001;
ANGWAVE[    37]:= -8.2801000E+0001;
ANGWAVE[    38]:= -5.7298000E+0001;
ANGWAVE[    39]:=  1.1053100E+0002;
ANGWAVE[    40]:=  1.7700900E+0002;
ANGWAVE[    41]:=  1.7209000E+0002;
ANGWAVE[    42]:= -4.1240000E+0001;
ANGWAVE[    43]:=  1.2272200E+0002;
ANGWAVE[    44]:= -1.7877600E+0002;
ANGWAVE[    45]:=  1.6406000E+0002;
ANGWAVE[    46]:= -1.3573100E+0002;
ANGWAVE[    47]:= -1.3181500E+0002;
ANGWAVE[    48]:=  4.5462000E+0001;
ANGWAVE[    49]:=  6.9194000E+0001;
ANGWAVE[    50]:=  8.7750000E+0001;
ANGWAVE[    51]:= -1.8703000E+0001;
ANGWAVE[    52]:=  3.2335000E+0001;
ANGWAVE[    53]:= -8.0030000E+0000;
ANGWAVE[    54]:= -1.2698000E+0002;
ANGWAVE[    55]:= -1.3134000E+0002;
ANGWAVE[    56]:=  8.9100000E+0001;
ANGWAVE[    57]:=  1.1120800E+0002;
ANGWAVE[    58]:= -1.4487700E+0002;
ANGWAVE[    59]:= -1.1803700E+0002;
ANGWAVE[    60]:= -2.7467000E+0001;
ANGWAVE[    61]:= -1.3899600E+0002;
ANGWAVE[    62]:= -6.0445000E+0001;
ANGWAVE[    63]:=  1.4450000E+0001;
ANGWAVE[    64]:=  2.0726000E+0001;
ANGWAVE[    65]:=  5.5953000E+0001;
ANGWAVE[    66]:= -1.6447000E+0002;
ANGWAVE[    67]:= -4.2329000E+0001;
ANGWAVE[    68]:=  6.8946000E+0001;
ANGWAVE[    69]:= -1.6790500E+0002;
ANGWAVE[    70]:=  1.3253700E+0002;
ANGWAVE[    71]:= -2.1905000E+0001;
ANGWAVE[    72]:= -1.3727900E+0002;
ANGWAVE[    73]:= -4.7145000E+0001;
```

```
ANGWAVE[   74]:= -4.0491000E+0001;
ANGWAVE[   75]:= -1.0292600E+0002;
ANGWAVE[   76]:= -1.6689300E+0002;
ANGWAVE[   77]:=  1.3451400E+0002;
ANGWAVE[   78]:= -1.1092600E+0002;
ANGWAVE[   79]:= -1.7231300E+0002;
ANGWAVE[   80]:= -8.2397000E+0001;
ANGWAVE[   81]:=  1.6179800E+0002;
ANGWAVE[   82]:=  1.1378800E+0002;
ANGWAVE[   83]:= -2.2516000E+0001;
ANGWAVE[   84]:= -1.4405900E+0002;
ANGWAVE[   85]:=  1.2876400E+0002;
ANGWAVE[   86]:=  5.6856000E+0001;
ANGWAVE[   87]:= -1.3762900E+0002;
ANGWAVE[   88]:= -2.5553000E+0001;
ANGWAVE[   89]:= -1.7610900E+0002;
ANGWAVE[   90]:= -1.5688500E+0002;
ANGWAVE[   91]:= -1.2389200E+0002;
ANGWAVE[   92]:=  1.1063100E+0002;
ANGWAVE[   93]:= -5.5307000E+0001;
ANGWAVE[   94]:=  2.6672000E+0001;
ANGWAVE[   95]:=  1.0340800E+0002;
ANGWAVE[   96]:= -1.0612600E+0002;
ANGWAVE[   97]:=  1.1841800E+0002;
ANGWAVE[   98]:= -1.5182300E+0002;
ANGWAVE[   99]:=  9.5932000E+0001;
ANGWAVE[  100]:= -4.6560000E+0001;
ANGWAVE[  101]:=  4.7489000E+0001;
ANGWAVE[  102]:= -1.6016000E+0002;
ANGWAVE[  103]:=  6.2062000E+0001;
ANGWAVE[  104]:= -1.9866000E+0001;
ANGWAVE[  105]:=  1.4471000E+0002;
ANGWAVE[  106]:= -1.2633200E+0002;
ANGWAVE[  107]:= -1.3514500E+0002;
ANGWAVE[  108]:=  1.4558900E+0002;
ANGWAVE[  109]:= -7.9021000E+0001;
ANGWAVE[  110]:=  1.3440100E+0002;
ANGWAVE[  111]:=  1.1211800E+0002;
ANGWAVE[  112]:= -1.4638900E+0002;
ANGWAVE[  113]:=  4.6351000E+0001;
ANGWAVE[  114]:= -7.0042000E+0001;
ANGWAVE[  115]:= -1.4871800E+0002;
ANGWAVE[  116]:=  7.2711000E+0001;
ANGWAVE[  117]:= -5.1906000E+0001;
ANGWAVE[  118]:=  1.6865000E+0001;
ANGWAVE[  119]:=  2.9642000E+0001;
ANGWAVE[  120]:= -1.1821200E+0002;
ANGWAVE[  121]:=  1.6532000E+0002;
ANGWAVE[  122]:=  9.8496000E+0001;
ANGWAVE[  123]:= -6.0848000E+0001;
ANGWAVE[  124]:=  1.4467600E+0002;
ANGWAVE[  125]:=  3.6181000E+0001;
ANGWAVE[  126]:= -9.9776000E+0001;
ANGWAVE[  127]:=  9.9672000E+0001;
ANGWAVE[  128]:=  1.5188500E+0002;
END;
```

```
(* Initialise MODWAVE and ANGWAVE with spectral estimates of *)
(* Prime M.F.T.S. *)
1:BEGIN

    MODWAVE[   1]:=  5.2403300E+0005;
    MODWAVE[   2]:=  6.5590000E+0000;
    MODWAVE[   3]:=  1.0000000E+0000;
    MODWAVE[   4]:=  4.3124109E+0004;
    MODWAVE[   5]:=  1.0000000E+0000;
    MODWAVE[   6]:=  4.3109605E+0004;
    MODWAVE[   7]:=  1.0000000E+0000;
    MODWAVE[   8]:=  4.3122078E+0004;
    MODWAVE[   9]:=  1.0000000E+0000;
    MODWAVE[  10]:=  3.3410000E+0000;
    MODWAVE[  11]:=  1.0000000E+0000;
    MODWAVE[  12]:=  4.3117387E+0004;
    MODWAVE[  13]:=  1.0000000E+0000;
    MODWAVE[  14]:=  4.3115132E+0004;
    MODWAVE[  15]:=  1.0000000E+0000;
    MODWAVE[  16]:=  6.1550000E+0000;
    MODWAVE[  17]:=  1.0000000E+0000;
    MODWAVE[  18]:=  4.3120348E+0004;
    MODWAVE[  19]:=  1.0000000E+0000;
    MODWAVE[  20]:=  4.3109448E+0004;
    MODWAVE[  21]:=  1.0000000E+0000;
    MODWAVE[  22]:=  1.8200000E+0000;
    MODWAVE[  23]:=  1.0000000E+0000;
    MODWAVE[  24]:=  4.3125370E+0004;
    MODWAVE[  25]:=  1.0000000E+0000;
    MODWAVE[  26]:=  4.5400000E-0001;
    MODWAVE[  27]:=  1.0000000E+0000;
    MODWAVE[  28]:=  3.2120000E+0000;
    MODWAVE[  29]:=  1.0000000E+0000;
    MODWAVE[  30]:=  4.3112073E+0004;
    MODWAVE[  31]:=  1.0000000E+0000;
    MODWAVE[  32]:=  4.3116587E+0004;
    MODWAVE[  33]:=  1.0000000E+0000;
    MODWAVE[  34]:=  7.7590000E+0000;
    MODWAVE[  35]:=  1.0000000E+0000;
    MODWAVE[  36]:=  1.9890000E+0000;
    MODWAVE[  37]:=  1.0000000E+0000;
    MODWAVE[  38]:=  4.3128150E+0004;
    MODWAVE[  39]:=  1.0000000E+0000;
    MODWAVE[  40]:=  7.0220000E+0000;
    MODWAVE[  41]:=  1.0000000E+0000;
    MODWAVE[  42]:=  4.3115126E+0004;
    MODWAVE[  43]:=  1.0000000E+0000;
    MODWAVE[  44]:=  4.3113400E+0004;
    MODWAVE[  45]:=  1.0000000E+0000;
    MODWAVE[  46]:=  3.4120000E+0000;
    MODWAVE[  47]:=  1.0000000E+0000;
    MODWAVE[  48]:=  4.3109508E+0004;
    MODWAVE[  49]:=  1.0000000E+0000;
    MODWAVE[  50]:=  6.5390000E+0000;
    MODWAVE[  51]:=  1.0000000E+0000;
```

```
MODWAVE[    52]:=  2.6830000E+0000;
MODWAVE[    53]:=  1.0000000E+0000;
MODWAVE[    54]:=  4.3118047E+0004;
MODWAVE[    55]:=  1.0000000E+0000;
MODWAVE[    56]:=  5.6040000E+0000;
MODWAVE[    57]:=  1.0000000E+0000;
MODWAVE[    58]:=  1.0542000E+0001;
MODWAVE[    59]:=  1.0000000E+0000;
MODWAVE[    60]:=  4.3118020E+0004;
MODWAVE[    61]:=  1.0000000E+0000;
MODWAVE[    62]:=  4.3130718E+0004;
MODWAVE[    63]:=  1.0000000E+0000;
MODWAVE[    64]:=  9.8820000E+0000;
MODWAVE[    65]:=  1.0000000E+0000;
MODWAVE[    66]:=  2.6170000E+0000;
MODWAVE[    67]:=  1.0000000E+0000;
MODWAVE[    68]:=  4.3122123E+0004;
MODWAVE[    69]:=  1.0000000E+0000;
MODWAVE[    70]:=  4.4460000E+0000;
MODWAVE[    71]:=  1.0000000E+0000;
MODWAVE[    72]:=  4.3116554E+0004;
MODWAVE[    73]:=  1.0000000E+0000;
MODWAVE[    74]:=  4.3119775E+0004;
MODWAVE[    75]:=  1.0000000E+0000;
MODWAVE[    76]:=  2.1400000E+0000;
MODWAVE[    77]:=  1.0000000E+0000;
MODWAVE[    78]:=  1.3801000E+0001;
MODWAVE[    79]:=  1.0000000E+0000;
MODWAVE[    80]:=  3.6010000E+0000;
MODWAVE[    81]:=  1.0000000E+0000;
MODWAVE[    82]:=  1.0309000E+0001;
MODWAVE[    83]:=  1.0000000E+0000;
MODWAVE[    84]:=  8.6270000E+0000;
MODWAVE[    85]:=  1.0000000E+0000;
MODWAVE[    86]:=  7.8120000E+0000;
MODWAVE[    87]:=  1.0000000E+0000;
MODWAVE[    88]:=  8.9010000E+0000;
MODWAVE[    89]:=  1.0000000E+0000;
MODWAVE[    90]:=  6.3220000E+0000;
MODWAVE[    91]:=  1.0000000E+0000;
MODWAVE[    92]:=  5.0990000E+0000;
MODWAVE[    93]:=  1.0000000E+0000;
MODWAVE[    94]:=  3.7740000E+0000;
MODWAVE[    95]:=  1.0000000E+0000;
MODWAVE[    96]:=  3.6300000E+0000;
MODWAVE[    97]:=  1.0000000E+0000;
MODWAVE[    98]:=  3.9090000E+0000;
MODWAVE[    99]:=  1.0000000E+0000;
MODWAVE[   100]:=  3.7190000E+0000;
MODWAVE[   101]:=  1.0000000E+0000;
MODWAVE[   102]:=  7.8340000E+0000;
MODWAVE[   103]:=  1.0000000E+0000;
MODWAVE[   104]:=  3.3290000E+0000;
MODWAVE[   105]:=  1.0000000E+0000;
MODWAVE[   106]:=  3.4650000E+0000;
MODWAVE[   107]:=  1.0000000E+0000;
```

```
MODWAVE[  108]:=  7.5900000E+0000;
MODWAVE[  109]:=  1.0000000E+0000;
MODWAVE[  110]:=  2.3330000E+0000;
MODWAVE[  111]:=  1.0000000E+0000;
MODWAVE[  112]:=  2.9160000E+0000;
MODWAVE[  113]:=  1.0000000E+0000;
MODWAVE[  114]:=  5.3660000E+0000;
MODWAVE[  115]:=  1.0000000E+0000;
MODWAVE[  116]:=  4.2870000E+0000;
MODWAVE[  117]:=  1.0000000E+0000;
MODWAVE[  118]:=  4.4490000E+0000;
MODWAVE[  119]:=  1.0000000E+0000;
MODWAVE[  120]:=  5.9360000E+0000;
MODWAVE[  121]:=  1.0000000E+0000;
MODWAVE[  122]:=  1.1853000E+0001;
MODWAVE[  123]:=  1.0000000E+0000;
MODWAVE[  124]:=  9.0480000E+0000;
MODWAVE[  125]:=  1.0000000E+0000;
MODWAVE[  126]:=  4.9630000E+0000;
MODWAVE[  127]:=  1.0000000E+0000;
MODWAVE[  128]:=  3.6730000E+0000;
  ANGWAVE[    1]:=  0.0000000E+0000;
  ANGWAVE[    2]:=  5.7049000E+0001;
  ANGWAVE[    3]:=  1.4272600E+0002;
  ANGWAVE[    4]:=  7.2936000E+0001;
  ANGWAVE[    5]:= -7.3123000E+0001;
  ANGWAVE[    6]:=  1.1571000E+0001;
  ANGWAVE[    7]:=  7.0307000E+0001;
  ANGWAVE[    8]:=  1.6525200E+0002;
  ANGWAVE[    9]:= -1.4588700E+0002;
  ANGWAVE[   10]:= -1.2940800E+0002;
  ANGWAVE[   11]:= -2.8120000E+0000;
  ANGWAVE[   12]:=  2.4621000E+0001;
  ANGWAVE[   13]:=  1.3926900E+0002;
  ANGWAVE[   14]:=  4.2365000E+0001;
  ANGWAVE[   15]:= -7.5937000E+0001;
  ANGWAVE[   16]:=  8.2580000E+0001;
  ANGWAVE[   17]:=  6.7485000E+0001;
  ANGWAVE[   18]:=  1.7989000E+0002;
  ANGWAVE[   19]:= -1.4888400E+0002;
  ANGWAVE[   20]:= -1.6700000E+0002;
  ANGWAVE[   21]:= -5.6250000E+0000;
  ANGWAVE[   22]:= -1.3219100E+0002;
  ANGWAVE[   23]:=  1.3528100E+0002;
  ANGWAVE[   24]:=  4.0477000E+0001;
  ANGWAVE[   25]:= -7.8750000E+0001;
  ANGWAVE[   26]:=  1.0384600E+0002;
  ANGWAVE[   27]:=  6.4651000E+0001;
  ANGWAVE[   28]:=  1.7924900E+0002;
  ANGWAVE[   29]:= -1.5179100E+0002;
  ANGWAVE[   30]:= -1.3376900E+0002;
  ANGWAVE[   31]:= -8.4380000E+0000;
  ANGWAVE[   32]:=  7.0480000E+0001;
  ANGWAVE[   33]:=  1.3965600E+0002;
  ANGWAVE[   34]:=  1.4833500E+0002;
  ANGWAVE[   35]:= -8.1562000E+0001;
```

```
ANGWAVE[     36]:=  5.1036000E+0001;
ANGWAVE[     37]:=  6.1791000E+0001;
ANGWAVE[     38]:=  1.5630000E+0000;
ANGWAVE[     39]:= -1.5465100E+0002;
ANGWAVE[     40]:=  9.3865000E+0001;
ANGWAVE[     41]:= -1.1250000E+0001;
ANGWAVE[     42]:= -1.6881000E+0001;
ANGWAVE[     43]:=  1.3471900E+0002;
ANGWAVE[     44]:=  4.8444000E+0001;
ANGWAVE[     45]:= -8.4375000E+0001;
ANGWAVE[     46]:=  8.3510000E+0000;
ANGWAVE[     47]:=  5.8884000E+0001;
ANGWAVE[     48]:=  1.6764300E+0002;
ANGWAVE[     49]:= -1.5748500E+0002;
ANGWAVE[     50]:= -6.4744000E+0001;
ANGWAVE[     51]:= -1.4063000E+0001;
ANGWAVE[     52]:= -7.0617000E+0001;
ANGWAVE[     53]:=  1.3073100E+0002;
ANGWAVE[     54]:= -1.4572300E+0002;
ANGWAVE[     55]:= -8.7187000E+0001;
ANGWAVE[     56]:=  1.1770600E+0002;
ANGWAVE[     57]:=  5.5887000E+0001;
ANGWAVE[     58]:= -1.5525100E+0002;
ANGWAVE[     59]:= -1.6030700E+0002;
ANGWAVE[     60]:= -1.1478100E+0002;
ANGWAVE[     61]:= -1.6877000E+0001;
ANGWAVE[     62]:=  1.0407500E+0002;
ANGWAVE[     63]:=  1.2727400E+0002;
ANGWAVE[     64]:= -1.3529800E+0002;
ANGWAVE[     65]:= -9.0000000E+0001;
ANGWAVE[     66]:= -1.5694500E+0002;
ANGWAVE[     67]:=  5.2726000E+0001;
ANGWAVE[     68]:=  1.2250200E+0002;
ANGWAVE[     69]:= -1.6312300E+0002;
ANGWAVE[     70]:= -1.2486300E+0002;
ANGWAVE[     71]:= -1.9693000E+0001;
ANGWAVE[     72]:=  1.2814500E+0002;
ANGWAVE[     73]:=  1.2411300E+0002;
ANGWAVE[     74]:=  1.5295900E+0002;
ANGWAVE[     75]:= -9.2812000E+0001;
ANGWAVE[     76]:= -2.7643000E+0001;
ANGWAVE[     77]:=  4.9269000E+0001;
ANGWAVE[     78]:= -1.1795400E+0002;
ANGWAVE[     79]:= -1.6593700E+0002;
ANGWAVE[     80]:=  1.3858300E+0002;
ANGWAVE[     81]:= -2.2515000E+0001;
ANGWAVE[     82]:= -7.7275000E+0001;
ANGWAVE[     83]:=  1.2111600E+0002;
ANGWAVE[     84]:= -1.0957900E+0002;
ANGWAVE[     85]:= -9.5625000E+0001;
ANGWAVE[     86]:= -2.7029000E+0001;
ANGWAVE[     87]:=  4.5281000E+0001;
ANGWAVE[     88]:= -9.7022000E+0001;
ANGWAVE[     89]:= -1.6875000E+0002;
ANGWAVE[     90]:= -1.6667600E+0002;
ANGWAVE[     91]:= -2.5349000E+0001;
```

```
            ANGWAVE[     92]:= -1.9926000E+0001;
            ANGWAVE[     93]:=  1.1820900E+0002;
            ANGWAVE[     94]:=  4.3711000E+0001;
            ANGWAVE[     95]:= -9.8438000E+0001;
            ANGWAVE[     96]:= -1.4917800E+0002;
            ANGWAVE[     97]:=  4.9656000E+0001;
            ANGWAVE[     98]:=  4.3798000E+0001;
            ANGWAVE[     99]:= -1.7156200E+0002;
            ANGWAVE[    100]:=  1.7628400E+0002;
            ANGWAVE[    101]:= -2.8209000E+0001;
            ANGWAVE[    102]:=  4.3813000E+0001;
            ANGWAVE[    103]:=  1.1534900E+0002;
            ANGWAVE[    104]:= -3.3385000E+0001;
            ANGWAVE[    105]:= -1.0125000E+0002;
            ANGWAVE[    106]:= -7.7077000E+0001;
            ANGWAVE[    107]:=  4.4719000E+0001;
            ANGWAVE[    108]:= -1.5908900E+0002;
            ANGWAVE[    109]:= -1.7437500E+0002;
            ANGWAVE[    110]:= -3.1757000E+0001;
            ANGWAVE[    111]:= -3.1116000E+0001;
            ANGWAVE[    112]:= -3.3054000E+0001;
            ANGWAVE[    113]:=  1.1251500E+0002;
            ANGWAVE[    114]:= -4.4740000E+0001;
            ANGWAVE[    115]:= -1.0406300E+0002;
            ANGWAVE[    116]:=  1.1256700E+0002;
            ANGWAVE[    117]:=  4.0731000E+0001;
            ANGWAVE[    118]:=  1.3803300E+0002;
            ANGWAVE[    119]:= -1.7718800E+0002;
            ANGWAVE[    120]:= -1.7274000E+0002;
            ANGWAVE[    121]:= -3.4113000E+0001;
            ANGWAVE[    122]:=  1.2466000E+0001;
            ANGWAVE[    123]:=  1.0969300E+0002;
            ANGWAVE[    124]:= -5.6558000E+0001;
            ANGWAVE[    125]:= -1.0687700E+0002;
            ANGWAVE[    126]:= -1.0419500E+0002;
            ANGWAVE[    127]:=  3.7274000E+0001;
            ANGWAVE[    128]:= -1.7453900E+0002;
         END;

    END;(* CASE *)

END;.
```

```
(**************************************************************************)
(*                                                                        *)
(*      MODULE - vdu2:s                                                   *)
(*                                                                        *)
(*      FUNCTION - Contains screen messages.                              *)
(*                                                                        *)
(*                                                                        *)
(*      EXPORT - ClearScreen,IntroOnScreen,DisplayMenu                    *)
(*               ErrorStatement,ExitMessage.                              *)
(*                                                                        *)
(*                                                                        *)
(*      AUTHOR - Lee Jones                                                *)
(*                                                                        *)
(**************************************************************************)

UNIT VDU ( I,O,ClearScreen,IntroOnScreen,DisplayMenu,
                                        ErrorStatement,ExitMessage) ;

CONST
        SCREENLGTH = 24 ;


VAR
        I,O : TEXT ;

PROCEDURE ClearScreen ;

(*      Clears VDU screen *)

        VAR
                Count : INTEGER ;

        BEGIN (* ClearScreen *)
                Rewrite (O) ;
                FOR Count := 1 TO SCREENLGTH DO
                        Writeln (O);
        END ; (* ClearScreen *)

PROCEDURE IntroOnScreen ;

(*      Displays welcome message to user *)

        VAR
                ContKey : CHAR ;

        PROCEDURE PAUSE;EXTERNAL;
        BEGIN (* IntroOnScreen *)

          Reset (I) ;
          Rewrite (O) ;

          Writeln (O,'                    The Polytechnic Of Wales') ;
          Writeln (O,'                    ------------------------') ;
          Writeln (O) ;
          Writeln (O,'         Frequency Response Instrument Prototype 2') ;
```

```
          Writeln (O,'                -------------------------------------------');
          Writeln (O) ;
          Writeln (O,'  Instrument enables systems to be analysed in the ') ;
          Writeln (O,'   frequency range 0.0001Hz to 57.031kHz            ') ;
          Writeln (O) ;
          Writeln (O,'   .                    Software Version 1.0');
          Writeln (O) ;
          PAUSE;
       END ; (* IntroOnScreen *)

PROCEDURE DisplayMenu ;

(* Displays Instrument Command Menu *)

BEGIN (* DisplayMenu *)

       Rewrite (O) ;
       Reset (I) ;

    Writeln (O,'          Frequency Response Instrument Command Menu');
    Writeln (O,'          -------------------------------------------');
    Writeln (O) ;
    Writeln (O,'1)   Obtain frequency response and transfer function of');
    Writeln (O,'     System Under Test  [ S.U.T. ] ');
    Writeln (O,'2)   Display frequency response of last test results as');
    Writeln (O,'     Bode plots on screen');
    Writeln (O,'3)   Terminate instrument operations ');
    Writeln (O) ;
    Writeln (O,'     Which option do you require  [ 1,2 or 3 ]   ? ');

END ; (* DisplayMenu *)

PROCEDURE ErrorStatement ;

(* User Error Statement *)

BEGIN (* ErrorStatement *)

       Rewrite (O) ;
Writeln(O,'** Please enter correct INTEGER reply for command menu options **');
       Writeln (O) ;

END ; (* ErrorStatement *)

PROCEDURE ExitMessage ;

(* Termination message to user *)

BEGIN (* ExitMessage *)

       Rewrite (O) ;
       ClearScreen ;

       Writeln (O) ;
       Writeln (O) ;
       Writeln (O) ;
```

```
  Writeln (0,'                        Instrument shutdown. ') ;
  Writeln (0,'                        -------------------- ') ;
        Writeln (0) ;
        Writeln (0) ;
        Writeln (0) ;
        Writeln (0) ;
  Writeln (0,'   Press F.R.I. RESET button to restart instrument operation.') ;
        Writeln (0) ;
        Writeln (0) ;

END;. (* ExitMessage *)
```

```
******************************************************************************
*                                                                            *
*       MODULE - rewrit.                                                     *
*                                                                            *
*       FUNCTION - Initialises SCN2661 USART and sets up output              *
*                  buffer.                                                   *
*                                                                            *
*       EXPORT - Q8RWRI                                                      *
*                                                                            *
*       AUTHOR - Lee Jones                                                   *
*                                                                            *
******************************************************************************
        MODEL   68000
*
REWRIT  RSECT   ROM
        ENTRY   MODE,COMD,STATUS,DREG
        ENTRY   Q8RWRI
******************************************************************************
*       USART EQUATE TABLE
*
******************************************************************************
MODE EQU        H'810005'       MODE1/MODE2 register
COMD    EQU     H'810007'       Command register
STATUS  EQU     H'810003'       Status register
DREG    EQU     H'810001'       Recieve/transmit data holding register
******************************************************************************
*       USART INITIALISATION
*
*       INITIALISATION ROUTINE FOR SCN 2661 USART ON GESMPU-4A BOARD.
*       JUMPER 4 SHOULD BE ABSENT SINCE INTERNAL BAUD RATE SELECTION IS ENABLED
*       AND DERIVED FROM A 5.0688 MHZ CRYSTAL. PINS 9 AND 25 ARE OUTPUT CLOCKS
*       IN THIS MODE.
******************************************************************************
Q8RWRI  MOVE.B  #B'11001101',MODE.L     Async, no parity, 8 bits,1 stop bit
        MOVE.B  #B'00111111',MODE.L     Internal clock, 300 baud
        MOVE.B  #B'00110111',COMD.L     NRTS low, reset error flags, NDTR low
******************************************************************************
*       REWRITE THE OUTPUT FILE 'O'
******************************************************************************
        MOVEA.L #OBUFF,A1       Load file output buffer address
*                               into A1
        MOVE.L  A1,(A0)         Transfer A1 to FCB bufp pointer
        RTS
*
*
************ RESERVE MEMORY FOR FILE OUTPUT BUFFER ***************************
*
        MODEL   68000
*
BUFF1   RSECT   RAM
        ENTRY   OBUFF
*
OBUFF   RESB    1       File output buffer
*
        END
```

```
***************************************************************************
*                                                                         *
*                     Procedure to Close a file  ( text )                 *
*                   ------------------------------------------            *
*                                                                         *
*                                                                          *
***************************************************************************
        MODEL   68000
*
CLOSE   RSECT   ROM
*
        ENTRY   Q8CLOS
*
*
Q8CLOS  RTS     Performs no function  ( All text files left open )
*
        END
```

```
*************************************************************************
*                                                                       *
*       MODULE - reset.                                                 *
*                                                                       *
*       FUNCTION - Initialises input file buffer,puts <sp> into        *
*                  input file buffer,sets EOLN in FCB.                  *
*                                                                       *
*       EXPORT - Q8RSET                                                 *
*                                                                       *
*       AUTHOR - Lee Jones                                              *
*                                                                       *
*************************************************************************
        MODEL   68000
*
RESET   RSECT   ROM
        ENTRY   Q8RSET
BLANK   EQU     H'20'   ASCII Space character
*************************************************************************
*
*  On entry :-     Register A0 has address of FCB bufp pointer
*
*************************************************************************
Q8RSET  MOVEA.L  #IBUFF,A1              Load file input buffer address into A1
        MOVE.L   A1,(A0)                Transfer A1 to FCB bufp pointer
*
        MOVE.B  #BLANK,IBUFF.L          Put 'space' character into file buffer
*
        BSET    #5,10(A0)               Set EOLN in FCB Status Register
*
        RTS
*
*
**************** RESERVE MEMORY FOR FILE INPUT BUFFER *********************
*
        MODEL   68000
*
BUFF2   RSECT   RAM
        ENTRY   IBUFF
*
IBUFF   RESB    1       File input buffer
*
        END
```

```
*****************************************************************************
*                                                                           *
*       MODULE - get.                                                       *
*                                                                           *
*       FUNCTION - Loads file input buffer with received character.         *
*                                                                           *
*                                                                           *
*       EXPORT - Q8GET                                                      *
*                                                                           *
*       FROM -  put      IMPORT - Q8PUT                                     *
*                                                                           *
*       AUTHOR - Lee Jones                                                  *
*                                                                           *
*****************************************************************************
         MODEL   68000
*
GET     RSECT   ROM
        EXTRN   Q8PUT,OBUFF,IBUFF,Q8WRLN,STATUS,DREG
        ENTRY   Q8GET
*
*       ASCII Code labels
CR      EQU     H'0D'   ASCII Carriage Return
BLANK   EQU     H'20'   ASCII Space character
*******************************************************************************
*
*  On entry :=   Register A0 has address of FCB bufp pointer
*  NOTE - D0 must not be altered by this routine
*
*******************************************************************************
Q8GET   MOVE.L  D0,-(A7)         Store D0 register on stack
WAIT    BTST    #1,STATUS.L      Test Rx status. - bit 1
*                                   - Buffer Empty
        BEQ.S   WAIT             Loop waiting for Buffer Empty
        MOVE.B  DREG.L,D0        Get USART data
        BCLR    #7,D0            Mask off parity bit - parity not checked
*
        CMPI.B  #CR,D0           Is keyboard character a carriage return?
        BNE.S   VALID            No
*
* Keyboard character is a carriage return
        BSET    #5,10(A0)        Set EOLN flag in STATUS integer of FCB
*                                record
        MOVE.B  #BLANK,IBUFF.L   Set file input buffer character to a
*                                'space'
        MOVE.L  (A7)+,D0         Restore D0 register from stack
        JSR     Q8WRLN.L         Output CR & LF to terminal
        RTS
*
* Keyboard character is valid    ( NOT a carriage return )
VALID   BCLR    #5,10(A0)        Clear EOLN flag in STATUS integer
*                                of FCB record
        BSR.S   INTCHK           Subroutine to validate menu choice
        BSR.S   ECHO             Subroutine to echo characters to terminal
        MOVE.L  (A7)+,D0         Restore D0 register from Stack
        RTS
*
```

```
*****************************************************************************
*
*       Subroutine INTCHK checks to see if character entered was a valid
*       menu choice. i.e. in the range '0'..'9'. Eroneous data recieved
*       is ignored. Valid characters placed in input file buffer.
*
******************************************************************************
INTCHK  CMPI.B  #H'30',D0
        BLT     INVAL
        CMPI.B  #H'39',D0
        BGT     INVAL
        MOVE.B  D0,IBUFF.L      Put valid character into input file buffer
        RTS
INVAL   MOVE    #H'00',D0       Null character moved to D0
        RTS
*
*
******************************************************************************
*
*       Subroutine ECHO character to be placed in file buffer is echoed
*       to the terminal.
*
******************************************************************************
ECHO    MOVE.B  D0,OBUFF.L
        JSR     O8PUT.L
        RTS
*
        END
```

```
*************************************************************************
*                                                                       *
*       MODULE - writln.                                                *
*                                                                       *
*       FUNCTION - Outputs <cr>,<lf> to terminal.                       *
*                                                                       *
*                                                                       *
*       EXPORT - Q8WRLN.                                                *
*                                                                       *
*       AUTHOR - Lee Jones                                              *
*                                                                       *
*************************************************************************
        MODEL   68000
*
WRITLN  RSECT   ROM
        EXTRN   STATUS,DREG
        ENTRY   Q8WRLN
*
*       ASCII Code Labels
CR      EQU     H'OD'   ASCII Carriage Return
LF      EQU     H'OA'   ASCII Line Feed
*
Q8WRLN  BTST    #0,STATUS.L     Test Tx Status . - bit 7 - Buffer Empty
        BEQ.S   Q8WRLN          Loop if last character not output
        MOVE.B  #LF,DREG.L      Transfer a 'Line feed' to USART data register
*
FULL    BTST    #0,STATUS.L     Test Tx Status . - bit 7 - Buffer Empty
        BEQ.S   FULL            Loop if last character not output
        MOVE.B  #CR,DREG.L      Transfer a 'carriage return' to
*                                               USART data register
*
        RTS
*
        END
```

```
************************************************************************
*                                                                      *
*       MODULE - put.                                                  *
*                                                                      *
*       FUNCTION - Loads transmit buffer of USART with output character.  *
*                                                                      *
*                                                                      *
*       EXPORT - Q8PUT                                                 *
*                                                                      *
*       AUTHOR - Lee Jones                                             *
*                                                                      *
************************************************************************
        MODEL   68000
*
PUT     RSECT   ROM
        EXTRN   STATUS,DREG,OBUFF
        ENTRY   Q8PUT
*
*  On entry :=    Register A0 has address of FCB bufp pointer
*
Q8PUT   BTST    #0,STATUS.L      Test Tx status . - bit 0
*                                  - Buffer Empty
        BEQ.S   Q8PUT            Loop if last character not
*                                  output
        MOVE.B  OBUFF.L,DREG.L   Transfer character from file
*                                  buffer to USART data register
*
        RTS
*
        END
```

```
*************************************************************************
*                                                                       *
*       MODULE - RANGE                                                  *
*                                                                       *
*       FUNCTION - Sets waveform output range,selests test waveform     *
*                  and timer on DAC board. Initialises frequency response *
*                  test. Waits for completion and loads integer array with *
*                  captured data.                                       *
*                                                                       *
*                                                                       *
*       EXPORT - INIT                                                   *
*                                                                       *
*       AUTHOR - Lee Jones                                              *
*                                                                       *
*************************************************************************
        MODEL  68000
SETUP   RSECT  ROM
        ENTRY   INIT
*
*
ORANGE  EQU H'800600'
CTRL13  EQU H'800640'
CTRL2   EQU H'800642'
CSRAM   EQU H'800501'
PTMB1   EQU H'800644'
PTMB2   EQU H'800648'
PTMB3   EQU H'80064C'
LATCH1  EQU H'800646'
LATCH2  EQU H'80064A'
LATCH3  EQU H'800640'
START   EQU H'800680'
*************************************************************
*       Set output voltage range and select test waveform
*
*************************************************************
INIT    MOVE.B H'FE000D'.L,ORANGE.L
*************************************************************
*
*       Set timer registers on dac board to give correct
*       test frequency range
*
*************************************************************
TIMER   MOVE.B H'FE0000'.L,PTMB1.L      MSB for timer1 of PTM
        MOVE.B H'FE0001'.L,LATCH1.L     LSB for timer1 of PTM
        MOVE.B H'FE0004'.L,PTMB2.L      MSB for timer2 of PTM
        MOVE.B H'FE0005'.L,LATCH2.L     LSB for timer2 of PTM
        MOVE.B H'FE0008'.L,PTMB3.L      MSB for timer3 of PTM
        MOVE.B H'FE0009'.L,LATCH3.L     LSB for timer3 of PTM
        MOVE.B #H'00',CTRL2.L           Enable access to CTRL3
        MOVE.B #H'90',CTRL13.L          Continuous extrn clock on CX
        MOVE.B #H'93',CTRL2.L           Continuous E clock
        MOVE.B #H'92',CTRL13.L          Continuous E clock
        MOVE.B #H'00',START.L           This starts the PTM counting
****************************************************************
*
*       Test for conversion complete
```

```
*        Load integer array with captured data
*
*******************************************************************
IORUN  MOVE.W   #0,D2             Setting data reg. 2 to zero
       MOVEA.L  #H'FE0010',A0     Start address of data array
*
CHECK  MOVE.B   H'8004c1'.L,D0      STAV ADC status reg. mapped in LinkLs
       BTST     #0,D0             First bit is set ? i.e. end of
       BEQ.S    CHECK             conversion ?
*
STOR   MOVE.W   #256,D1
*
       MOVE.B   CSRAM.L,D3         One dummy read of ADC ram to increment
       MOVE.B   CSRAM.L,D4         counter to point to first captured byte
*
LOOP   MOVE.L   #0,D3
       MOVE.L   #0,D4
       MOVE.B   CSRAM.L,D3
       LSL.W    #8,D3
       MOVE.B   CSRAM.L,D4
       ADD.W    D4,D3
       ANDI.L   #H'00000FFF',D3
       MOVE.W   D3,(A0)+   Increment address counter for the array
       ADDI     #1,D2
       CMPI.W   #256,D2    Compare to see if last data word has
*                          been transfered
       BNE.S    LOOP
PASCAL RTS
       END
```

```
(****************************************************************************)
(* This procedure calculates the fast fourier transform of the input data *)
(* taken form file TESTFI:s          LEE JONES                              *)
(****************************************************************************)
   PROGRAM FT (INPUT,OUTPUT,TESTFI,TESTFT);
   TYPE
        data = ARRAY[1..1024] OF REAL;
    VAR
          I,N,SIZE:INTEGER;
          RTEST:data;
          ITEST:data;
          TESTFI,TESTFT: text;
        FUNCTION exponent (INVAL : INTEGER;COUNT3 : INTEGER):INTEGER;
(*             Raises number (INVAL) to positive integer power (COUNT3) *)
        VAR A,ANSWER : INTEGER;
        BEGIN
             ANSWER:=1;
             A:=0;
             WHILE (A<COUNT3) DO
             BEGIN
                ANSWER:= ANSWER*INVAL;
                A:=A+1;
             END;
             exponent:=ANSWER
        END;

   PROCEDURE FFT (N,SIZE:INTEGER;VAR REALDT,IMAGE:data);
        LABEL 1;
        (* Definition of procedure constants *)
        CONST
                PI = 3.14156265389793;
                INVAL = 2;
(* *)
        VAR
             UREAL,UIMAG,WREAL,WIMAG,TREAL,TIMAG : REAL;
             UR,UI,X : REAL;
             NBD2,NBM1,COUNT,COUNT2,COUNT3,KTEMP,ME,LPK,DUMMY :INTEGER;
             VAL : INTEGER;
        BEGIN
             NBD2 := SIZE DIV 2;
             NBM1 := SIZE-1;
             COUNT := 1;
             FOR COUNT2 := 1 TO NBM1 DO
                BEGIN  (* Start of the FOR/DO Loop *)
                  IF COUNT2 < COUNT THEN
                   BEGIN
                         TREAL := REALDT [COUNT];
                         TIMAG := IMAGE [COUNT];
                         REALDT [COUNT] := REALDT [COUNT2];
                         IMAGE [COUNT] := IMAGE [COUNT2];
                         REALDT [COUNT2] := TREAL;
                         IMAGE [COUNT2] := TIMAG;
                  END;
                  KTEMP := NBD2;
        1:        IF KTEMP < COUNT THEN
                    BEGIN
```

```
                COUNT := COUNT - KTEMP;
                KTEMP := KTEMP DIV 2;
                GOTO 1
              END
            ELSE
              COUNT := COUNT + KTEMP
          END;   (* End of FOR/DO Loop *)
    FOR  COUNT3 := 1 TO N DO
         BEGIN
           UREAL := 1.0;
           UIMAG := 0.0;
           ME := exponent(INVAL,COUNT3);  (* Calls function exponent *)
           KTEMP := ME DIV 2;
           X := PI/KTEMP;
(*                a + jb = Cos x + j Sin x                    *)
           WREAL := cos(X);
           WIMAG := sin(X);
           WIMAG := -WIMAG;
           FOR COUNT := 1 TO KTEMP DO
              BEGIN    (*Start of 1st inner Loop*)
                 VAL := COUNT;
                 WHILE  (VAL <= SIZE) DO
                      BEGIN
                         COUNT2 := VAL;
                         LPK:= COUNT2 + KTEMP;
                         TREAL:=(REALDT[LPK]*UREAL)-(IMAGE[LPK]*UIMAG);
                         TIMAG:=(REALDT[LPK]*UIMAG)+(IMAGE[LPK]*UREAL);
                         REALDT[LPK]:=REALDT[COUNT2] - TREAL;
                         IMAGE[LPK]:=IMAGE[COUNT2] - TIMAG;
                         REALDT[COUNT2]:=REALDT[COUNT2] + TREAL;
                         IMAGE[COUNT2]:=IMAGE[COUNT2] + TIMAG;
                         VAL := VAL + ME;
                      END;
                UR := UREAL;
                UI := UIMAG;
                UREAL:=(UR*WREAL)-(UI*WIMAG);
                UIMAG:=(UR*WIMAG)+(UI*WREAL);
              END;   (*End of 1st inner Loop*)
          END; (*End of for/do Loop *)
END;  (* End of proceedure fft *)
PROCEDURE ANG (SIZE:INTEGER;VAR REALDT,IMAGE:data);
(* Defining procedure constants *)
CONST
   PI = 3.141592653589793;
(* Defining procedure variables *)
VAR
   COUNT:INTEGER;
   MAGFFT,TANG,ANGLE,Z,MAG,Harmonic : REAL;
(* Definition of functions *)
(* This function calcs. the arctan of the input variable Z  *)
FUNCTION LOG( X:REAL):REAL;
VAR LOG1:REAL;
BEGIN
  IF ABS(X) > 1 THEN
  BEGIN
    LOG1:=-1;
```

```
    REPEAT
    X:=ABS(X)/10;
    LOG1:=LOG1+1;
    UNTIL X<1;
    LOG:=LOG1
  END
  ELSE
  IF X<>0 THEN
  BEGIN
    LOG1:=0;
    REPEAT
    X:=ABS(X)*10;
    LOG1:=LOG1+1;
    UNTIL X>1;
    LOG:=-LOG1
   END
   ELSE
    LOG:= -20
END;
FUNCTION atangent (Z : REAL):REAL;
CONST
   PI = 3.14159265358979З;
(* Function variables *)
VAR
   POWER,P,CNT:INTEGER;
   SUM,M,W:REAL;
BEGIN    (* Function start *)
   SUM := 0.0;
   P := -1;
   POWER :=1;
   WHILE POWER < 6  DO
      BEGIN
        CNT :=1;
        IF Z > -1 THEN
           BEGIN
              IF Z > 1 THEN
                 BEGIN
                    M := 1/Z;
                    W := 1/Z;
                 END
              ELSE
                 BEGIN
                    M :=Z;
                    W :=Z;
                 END;
           END
        ELSE
           BEGIN
              M := 1/Z;
              W := 1/Z;
           END;
        WHILE CNT < POWER DO
           BEGIN
              M :=M*W;
              CNT :=CNT+1;
           END;
```

```
          M := M/POWER;
          P := -1*P;
          IF P > 0 THEN
             SUM :=SUM+M
          ELSE
             SUM :=SUM-M;
          POWER :=POWER+2;
        END;
     IF Z < 1 THEN
        BEGIN
          IF Z < -1 THEN
             atangent := -PI/2 - SUM
          ELSE
             atangent := SUM;
        END
     ELSE
        atangent := PI/2 - SUM;
END;     (* of function atangent *)
BEGIN  (* Procedure start *)
      FOR COUNT := 1 TO (SIZE DIV 2) DO
         BEGIN
           (* Calculating the magnitude *)
           MAG := SQR(REALDT[COUNT])+SQR(IMAGE[COUNT]);
           IF MAG = 1.0 THEN
              MAGFFT := MAG
           ELSE
              MAGFFT:= SQRT(100*MAG)/10;
           (* Calculating the phase *)
           IF REALDT[COUNT] <> 0.0 THEN
             BEGIN
               IF (( LOG(IMAGE[COUNT])-LOG(REALDT[COUNT]))> 6) THEN
               BEGIN
                 TANG:=-90;
                 IF ( REALDT[COUNT]>0) AND (IMAGE[COUNT]>0) THEN
                 TANG:=90;
                 IF (REALDT[COUNT]<0) AND (IMAGE[COUNT]<0) THEN
                 TANG:=90
               END;
               IF ((LOG(IMAGE[COUNT])-LOG(REALDT[COUNT]))<-6) THEN
               TANG:=0;
               IF ((TANG<>0) OR (ABS(TANG)<>90)) THEN
               BEGIN
                  Z := IMAGE[COUNT]/REALDT[COUNT];
                  TANG := atangent (Z);
                  TANG:= TANG*180/PI;
               END;
               IF IMAGE[COUNT] <> 0.0 THEN
                 (* This loop calcs angles in any of the 4 quadrants *)
                 BEGIN
                   IF REALDT[COUNT] < 0.0 THEN
                      ANGLE := TANG +180
                   ELSE
                      ANGLE := TANG
                 END
               ELSE
                 (* This loop determines angle on real axis- 0 or -180*)
```

```
                BEGIN
                  IF REALDT[COUNT] <0.0 THEN
                    ANGLE := -180
                  ELSE
                    ANGLE := 0.0
                END;
            END
          (* This loop determines angle on imag. axis ie, 90 or 270 *)
          ELSE
            BEGIN
              IF IMAGE[COUNT] < 0.0 THEN
                ANGLE := 270
              ELSE
                BEGIN
                  IF IMAGE[COUNT] > 0.0 THEN
                    ANGLE := 90
                  ELSE
                    ANGLE := 0.0
                END
            END;
        IF ANGLE > 180 THEN
           ANGLE := ANGLE-360;
        (* Loading the magnitude and angle into the real and imag. *)
        (* arrays respectively.                                    *)
        REALDT[COUNT] := MAGFFT;
        IMAGE[COUNT] := ANGLE;

        Harmonic := ( COUNT - 1 ) ;
        WRITELN(TESTFT,REALDT[COUNT]:10:3,IMAGE[COUNT]:10:3,Harmonic:12:1);
        END;
   END;    (* End of procedure *)

BEGIN
       RESET(TESTFI);
       REWRITE(TESTFT);
       WRITELN('INPUT N - ');
       READLN;READ(N);
       SIZE := exponent(2,N);
       WRITELN(N);
       FOR I:=1 TO SIZE DO
       BEGIN
            ITEST[I]:=0;
            READLN(TESTFI,RTEST[I]);
       END;
       FFT(N,SIZE,RTEST,ITEST);
       ANG(SIZE,RTEST,ITEST);

END.
```

```
PROGRAM ARYFIL ( TESTFT,TSTARY);
VAR TESTFT,TSTARY : text;
    I : INTEGER;
    TEMP : real;
BEGIN
        RESET(TESTFT);
        REWRITE(TSTARY);
        FOR I :=1 TO 128 DO
            BEGIN
                READ(TESTFT,TEMP);
                WRITELN(TSTARY,'     MODPRBS[',I,']:= ',TEMP);
                READLN(TESTFT);
            END;
        RESET(TESTFT);
        FOR I:= 1 TO 128 DO
           BEGIN
                READ(TESTFT,TEMP);READ(TESTFT,TEMP);
                WRITELN(TSTARY,'      ANGPRBS[',I,']:= ',TEMP);
                READLN(TESTFT);
           END;
 END.
```

```
PROGRAM ROMFIL (TESTFI,WAVE);
VAR     TESTFI,WAVE :TEXT;
        I,W,TEMP : INTEGER;
BEGIN
        REWRITE(WAVE);
        WRITELN(WAVE,'          MODEL    68000 ');
        WRITELN(WAVE,'WAVE      ASECT    ROM ');
        WRITELN(WAVE,'          ORG      1024');
        FOR W :=1 TO 2 DO
           BEGIN
                RESET(TESTFI);
                FOR I:= 1 TO 256 DO
                  BEGIN
                        READLN(TESTFI,TEMP);
                        WRITELN(WAVE,'          ACON     ',TEMP);
                  END;
           END;
        WRITELN(WAVE,'          END')
END.
```

APPENDIX E

Numerical Results from Comparative Testing

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
0.000	1.813	0.000
1.500	0.774	356.720
2.500	0.780	354.414
3.500	0.786	351.746
5.500	0.803	347.041
6.500	0.826	343.076
8.500	0.873	339.772
9.500	0.909	336.555
11.500	0.993	329.508
14.500	1.193	315.826
15.500	1.305	313.928
18.500	1.777	289.816
20.500	2.076	261.500
21.500	2.054	240.030
23.500	1.611	212.324
26.500	0.945	185.582
29.500	0.589	170.344
30.500	0.517	167.531
33.500	0.355	160.990
35.500	0.286	156.197
36.500	0.253	157.240

Test Parameters

Fundamental Test Frequency - 0.5Hz
Output Voltage Range -$\pm$1V
Simulator Gain 25%

Measured Response of Linear Third Order System

Test Waveform - Prime Composite

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)	FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
0.000	3.836	0.000	25.000	1.304	195.485
0.500	0.815	358.631	25.500	1.166	194.381
1.000	0.827	357.632	26.000	1.132	187.326
1.500	0.820	355.457	26.500	1.010	185.203
2.000	0.823	354.821	27.000	0.925	182.426
2.500	0.833	352.902	27.500	0.843	179.150
3.000	0.841	350.571	28.000	0.778	177.872
3.500	0.842	357.145	28.500	0.726	174.878
4.000	0.853	351.452	29.000	0.674	171.731
4.500	0.835	350.412	29.500	0.627	170.258
5.000	0.858	345.193	30.000	0.591	174.412
5.500	0.864	351.578	30.500	0.549	168.249
6.000	0.877	350.116	31.000	0.513	166.264
6.500	0.885	344.289	31.500	0.487	163.362
7.000	0.864	341.450	32.000	0.429	167.377
7.500	0.912	342.542	32.500	0.412	162.099
8.000	0.904	340.539	33.000	0.409	156.250
8.500	0.930	338.751	33.500	0.398	158.705
9.000	0.959	338.131	34.000	0.353	159.216
9.500	0.970	337.149	34.500	0.346	156.287
10.000	0.967	337.572	35.000	0.324	155.720
10.500	1.000	334.341	35.500	0.308	160.563
11.000	1.045	333.976	36.000	0.277	157.884
11.500	1.055	330.815	36.500	0.252	160.206
12.000	1.078	333.880	37.000	0.198	158.640
12.500	1.115	326.994	37.500	0.245	153.306
13.000	1.144	325.089	38.000	0.238	149.314
13.500	1.187	321.040	38.500	0.220	153.844
14.000	1.215	321.894	39.000	0.218	152.156
14.500	1.260	318.689	39.500	0.221	159.446
15.000	1.318	315.992	40.000	0.203	145.566
15.500	1.377	314.137	40.500	0.209	147.840
16.000	1.455	312.688	41.000	0.191	150.460
16.500	1.511	309.077	41.500	0.169	160.023
17.000	1.601	303.284	42.000	0.170	150.434
17.500	1.692	299.705	42.500	0.160	148.136
18.000	1.768	295.454	43.000	0.162	145.455
18.500	1.859	288.915	43.500	0.162	153.941
19.000	1.976	284.476	44.000	0.128	148.224
19.500	2.068	277.500	44.500	0.137	153.159
20.000	2.145	269.717	45.000	0.147	150.019
20.500	2.216	258.411	45.500	0.114	146.185
21.000	2.201	252.755	46.000	0.131	152.149
21.500	2.178	244.156	46.500	0.112	152.856
22.000	2.082	236.748	47.000	0.099	150.557
22.500	1.981	224.225	47.500	0.123	152.134
23.000	1.865	221.642	48.000	0.105	146.537
23.500	1.709	214.532	48.500	0.098	139.629
24.000	1.584	205.709	49.000	0.089	141.383
24.500	1.435	199.850			

Test Parameters

Fundamental Frequency -0.5Hz

Output Voltage Range $-\pm 1$V

Simulator Gain -25%

Figure E3 Measure Response of Linear Third Order System

Test Waveform - Modified PRBS

FREQ(Hz)	MAGN(T.F.)	PH(T.F.)
.500000	.785319	358.8
1.00000	.786167	357.8
1.50000	.788814	356.6
2.00000	.790580	355.4
2.50000	.795187	354.3
3.00000	.798834	353.2
3.50000	.801324	352.1
4.00000	.808853	350.9
4.50000	.815181	349.8
5.00000	.821834	348.5
5.50000	.828774	347.2
6.00000	.838183	346.1
6.50000	.847883	345.0
7.00000	.857083	343.7
7.50000	.868782	342.4
8.00000	.880058	341.3
8.50000	.898245	339.9
9.00000	.913543	338.5
9.50000	.934911	337.1
10.0000	.953336	335.5
10.5000	.974237	334.1
11.0000	.997432	332.5
11.5000	1.02341	331.1
12.0000	1.05293	329.2
12.5000	1.08347	327.5
13.0000	1.11833	325.5
13.5000	1.15585	323.4
14.0000	1.19724	321.3
14.5000	1.24824	319.2
15.0000	1.29604	316.6
15.5000	1.35280	313.9
16.0000	1.41947	310.9
16.5000	1.49844	307.5
17.0000	1.57392	304.0
17.5000	1.65857	299.8
18.0000	1.75555	295.3
18.5000	1.85180	289.8
19.0000	1.95099	284.0
19.5000	2.05013	277.2
20.0000	2.12827	269.8
20.5000	2.18861	261.6
21.0000	2.19759	252.5
21.5000	2.17122	243.9
22.0000	2.05839	235.3
22.5000	1.99412	226.8
23.0000	1.86317	219.2
23.5000	1.72473	212.2
24.0000	1.58411	206.0
24.5000	1.45091	200.6
25.0000	1.32859	195.9
25.5000	1.21419	191.7
26.0000	1.11159	187.9
26.5000	1.02494	184.7
27.0000	.943207	181.9
27.5000	.872565	179.1
28.0000	.809058	176.9
28.5000	.749359	174.5
29.0000	.698826	172.5
29.5000	.647287	170.7
30.0000	.605415	169.0
30.5000	.567011	167.4
31.0000	.531003	[illegible]
31.5000	.500811	164.4
32.0000	.473821	163.2
32.5000	.445733	162.2
33.0000	.422452	161.3
33.5000	.398718	159.5
34.0000	.376728	158.7
34.5000	.350484	157.8
35.0000	.338438	156.8
35.5000	.324658	155.7
36.0000	.306803	154.9
36.5000	.295766	154.5
37.0000	.283185	153.2
37.5000	.270617	152.3
38.0000	.258551	151.4
38.5000	.245856	150.8
39.0000	.236350	150.5
39.5000	.225451	149.7
40.0000	.218322	148.8
40.5000	.207504	147.8
41.0000	.200937	147.0
41.5000	.192499	146.3
42.0000	.183886	146.0
42.5000	.176812	145.5
43.0000	.171938	145.0
43.5000	.163631	144.9
44.0000	.158174	143.7
44.5000	.155024	143.3
45.0000	.146854	142.9
45.5000	.144201	142.7
46.0000	.139703	141.3
46.5000	.134178	141.0
47.0000	.129046	140.1
47.5000	.126907	140.1
48.0000	.120138	139.5
48.5000	.117315	139.3
49.0000	.113379	138.3
49.5000	.110904	138.9
50.0000	.105805	138.1

Test Parameters

Fundamental Test Frequency -0.5Hz
Output Voltage Range $-\pm 1$V   Simulator Gain -25%

Figure E4   Measured Response of Linear Third Order System
Test Waveform - Sinusoid (Conventional Analyser)

FREQ.	MAGN.	PHASE
3.000	0.973	329.708
5.000	0.891	305.988
7.000	0.787	288.218
11.000	0.569	255.703
13.000	0.470	240.663
17.000	0.319	220.519
19.000	0.271	213.112
23.000	0.189	192.848
29.000	0.110	171.252
31.000	0.098	170.212
37.000	0.062	160.118
41.000	0.045	155.844
43.000	0.038	143.528
47.000	0.026	146.254
53.000	0.024	143.279
59.000	0.015	125.366
61.000	0.012	137.353
67.000	0.011	119.197
71.000	0.000	134.662
73.000	0.009	126.911

Measured Response of Open-loop Third
Order System with Saturation (β = 33.3%)

Test Waveform - Prime Composite

FREQ.	MAGN.	PHASE
1.000	1.053	344.8
2.000	0.956	332.7
3.000	0.925	329.3
4.000	0.918	324.2
5.000	0.788	304.3
6.000	0.800	289.9
7.000	0.720	290.0
8.000	0.627	280.1
9.000	0.579	274.2
10.000	0.536	260.7
11.000	0.519	261.3
12.000	0.453	252.5
13.000	0.389	249.6
14.000	0.503	232.5
15.000	0.351	236.6
16.000	0.390	216.1
17.000	0.381	224.8
18.000	0.195	210.1
19.000	0.288	201.0
20.000	0.205	209.1
21.000	0.164	209.7
22.000	0.269	197.3
23.000	0.131	175.7
24.000	0.191	189.8
25.000	0.189	181.7
26.000	0.103	157.6
27.000	0.117	161.9
28.000	0.150	169.0
29.000	0.088	180.0
30.000	0.137	184.7
31.000	0.060	163.5
32.000	0.101	177.9
33.000	0.080	182.1
34.000	0.053	172.0
35.000	0.047	150.4
36.000	0.035	135.5
37.000	0.069	164.7

FREQ.	MAGN	PHASE
38.000	0.051	148.4
39.000	0.060	173.9
40.000	0.058	138.5
41.000	0.040	174.0
42.000	0.048	156.1
43.000	0.040	157.4
44.000	0.046	162.4
45.000	0.026	138.4
46.000	0.020	179.9
47.000	0.035	156.1
48.000	0.031	160.4
49.000	0.033	157.3
50.000	0.031	133.8
51.000	0.026	137.6
52.000	0.015	163.5
53.000	0.019	135.9
54.000	0.025	137.7
55.000	0.019	161.4
56.000	0.015	155.2
57.000	0.017	130.6
58.000	0.018	179.0
59.000	0.013	138.7
60.000	0.018	132.1
61.000	0.015	92.09
62.000	0.018	132.5
63.000	0.020	86.22
64.000	0.018	157.6
65.000	0.017	110.1
66.000	0.009	142.8
67.000	0.003	129.7
68.000	0.016	130.5
69.000	0.006	133.9
70.000	0.006	166.7
71.000	0.007	126.2
72.000	0.004	2.615
73.000	0.008	99.47

Measured Response of Open-loop Third Order System with Saturation (β = 33.3%)

Test Waveform - Modified PRBS

FREQ.	MAGN.	PHASE
3.000	0.899	331.3
5.000	0.818	307.1
7.000	0.748	285.6
11.000	0.545	254.8
13.000	0.403	240.6
17.000	0.261	221.2
19.000	0.272	212.2
23.000	0.167	193.8
29.000	0.106	173.3
31.000	0.074	172.1
37.000	0.062	173.3
41.000	0.030	142.6
43.000	0.037	114.9
47.000	0.018	133.7
53.000	0.026	137.1
59.000	0.013	119.8
61.000	0.009	130.0
67.000	0.013	118.3
71.000	0.011	131.8
73.000	0.006	121.3

Measured Response of Open-loop Third Order System with Saturation (β = 50%)

Test Waveform - Prime Composite

FREQ.	MAGN.	PHASE
1.000	0.846	333.6
2.000	0.747	328.5
3.000	0.552	333.1
4.000	0.742	341.8
5.000	0.332	269.8
6.000	0.587	297.0
7.000	0.692	283.6
8.000	0.389	274.0
9.000	0.391	268.2
10.000	0.272	252.5
11.000	0.379	266.6
12.000	0.418	254.9
13.000	0.085	256.4
14.000	0.352	209.5
15.000	0.135	265.7
16.000	0.352	200.8
17.000	0.403	227.6
18.000	0.066	147.0
19.000	0.196	216.6
20.000	0.070	260.3
21.000	0.090	230.3
22.000	0.153	229.2
23.000	0.070	100.0
24.000	0.135	240.0
25.000	0.177	174.4
26.000	0.090	149.6
27.000	0.001	187.5
28.000	0.109	117.8
29.000	0.100	218.4
30.000	0.082	251.0
31.000	0.032	230.5
32.000	0.165	119.6
33.000	0.083	155.1
34.000	0.083	154.4
35.000	0.089	202.7
36.000	0.018	253.7
37.000	0.229	121.6

FREQ.	MAGN.	PHASE
38.000	0.026	75.05
39.000	0.056	185.4
40.000	0.013	174.0
41.000	0.015	336.3
42.000	0.049	152.2
43.000	0.059	135.1
44.000	0.019	359.4
45.000	0.011	85.92
46.000	0.081	0.840
47.000	0.019	81.91
48.000	0.039	100.4
49.000	0.035	220.6
50.000	0.014	244.3
51.000	0.027	132.8
52.000	0.098	150.3
53.000	0.016	176.3
54.000	0.020	148.9
55.000	0.022	105.6
56.000	0.024	109.3
57.000	0.029	142.1
58.000	0.015	16.25
59.000	0.010	22.98
60.000	0.021	208.0
61.000	0.019	238.5
62.000	0.016	163.5
63.000	0.030	89.49
64.000	0.018	6.075
65.000	0.019	160.6
66.000	0.017	258.7
67.000	0.016	80.46
68.000	0.004	12.15
69.000	0.011	146.5
70.000	0.021	203.8
71.000	0.009	43.64
72.000	0.007	23.52
73.000	0.030	72.13

Measured Response of Open-loop Third Order System
with Saturation (β = 50%)

Test Waveform - Modified PRBS

FREQ.	MAGN.	PHASE
3.000	0.515	343.4
5.000	0.525	332.2
7.000	0.541	319.9
11.000	0.559	291.8
13.000	0.542	270.9
17.000	0.419	234.6
19.000	0.350	218.2
23.000	0.229	194.8
29.000	0.128	173.7
31.000	0.107	170.4
37.000	0.064	157.9
41.000	0.048	150.9
43.000	0.046	142.3
47.000	0.037	147.4
53.000	0.021	132.2
59.000	0.017	138.0
61.000	0.015	133.8
67.000	0.014	122.2
71.000	0.009	141.1
73.000	0.012	122.0

Measured Response of Closed-loop Third Order System with Saturation (β = 33.3%)

Test Waveform - Prime Composite

FREQ.	MAGN.	PHASE
1.000	0.504	354.4
2.000	0.509	349.3
3.000	0.509	341.1
4.000	0.517	337.3
5.000	0.524	332.2
6.000	0.531	321.8
7.000	0.538	320.3
8.000	0.542	312.9
9.000	0.552	305.3
10.000	0.556	297.4
11.000	0.555	292.0
12.000	0.551	283.3
13.000	0.536	271.0
14.000	0.512	260.9
15.000	0.489	252.0
16.000	0.457	242.5
17.000	0.420	236.0
18.000	0.382	225.2
19.000	0.348	220.1
20.000	0.313	213.0
21.000	0.280	207.6
22.000	0.251	201.8
23.000	0.229	194.1
24.000	0.203	197.0
25.000	0.184	189.0
26.000	0.162	186.5
27.000	0.153	182.5
28.000	0.136	179.0
29.000	0.129	176.2
30.000	0.118	172.8
31.000	0.107	169.8
32.000	0.093	170.4
33.000	0.086	164.6
34.000	0.083	163.3
35.000	0.078	160.7
36.000	0.064	160.8
37.000	0.065	162.5

FREQ.	MAGN.	PHASE
38.000	0.059	157.9
39.000	0.057	156.9
40.000	0.054	151.9
41.000	0.051	146.7
42.000	0.048	152.7
43.000	0.040	148.3
44.000	0.039	149.0
45.000	0.039	147.3
46.000	0.034	122.5
47.000	0.035	144.9
48.000	0.033	141.5
49.000	0.029	141.6
50.000	0.028	140.5
51.000	0.026	145.7
52.000	0.023	160.9
53.000	0.022	147.6
54.000	0.018	138.1
55.000	0.020	142.0
56.000	0.017	134.5
57.000	0.020	138.9
58.000	0.019	146.2
59.000	0.016	137.2
60.000	0.017	150.4
61.000	0.017	126.2
62.000	0.013	135.5
63.000	0.015	142.5
64.000	0.013	139.6
65.000	0.010	160.2
66.000	0.012	125.7
67.000	0.012	135.3
68.000	0.000	121.1
69.000	0.013	115.5
70.000	0.012	121.4
71.000	0.009	131.3
72.000	0.010	140.8
73.000	0.005	92.54

Measured Response of Third Order Closed-loop System with Saturation (β = 33.3%)

Test Waveform - Modified PRBS

FREQ.	MAGN.	PHASE
3.000	0.513	343.099
5.000	0.523	332.088
7.000	0.543	319.350
11.000	0.555	291.655
13.000	0.527	270.172
17.000	0.415	233.302
19.000	0.336	218.769
23.000	0.224	198.077
29.000	0.127	178.313
31.000	0.107	170.829
37.000	0.061	154.341
41.000	0.051	149.692
43.000	0.047	135.880
47.000	0.035	141.497
53.000	0.027	146.032
59.000	0.017	143.112
61.000	0.014	125.256
67.000	0.010	129.668
71.000	0.008	140.039
73.000	0.007	163.004

Measured Response of Closed-loop Third Order System with Saturation (β = 50%)

Test Waveform - Prime Composite

FREQ.	MAGN.	PHASE
1.000	0.514	352.4
2.000	0.504	344.5
3.000	0.508	345.7
4.000	0.506	339.6
5.000	0.499	326.9
6.000	0.522	318.7
7.000	0.493	317.6
8.000	0.487	313.3
9.000	0.531	303.2
10.000	0.531	290.7
11.000	0.515	293.0
12.000	0.512	277.5
13.000	0.461	273.3
14.000	0.528	263.4
15.000	0.417	247.2
16.000	0.472	235.7
17.000	0.401	245.7
18.000	0.341	218.0
19.000	0.353	220.8
20.000	0.280	202.9
21.000	0.250	213.2
22.000	0.230	209.7
23.000	0.243	182.1
24.000	0.203	195.3
25.000	0.209	195.4
26.000	0.140	174.6
27.000	0.159	183.5
28.000	0.176	184.5
29.000	0.130	183.0
30.000	0.092	180.8
31.000	0.083	162.9
32.000	0.108	162.6
33.000	0.067	175.3
34.000	0.090	178.8
35.000	0.089	161.6
36.000	0.059	130.7
37.000	0.099	164.4

FREQ.	MAGN.	PHASE
38.000	0.069	143.5
39.000	0.070	168.1
40.000	0.046	155.0
41.000	0.014	168.1
42.000	0.036	146.0
43.000	0.027	145.5
44.000	0.065	177.5
45.000	0.033	162.4
46.000	0.052	40.96
47.000	0.030	126.0
48.000	0.033	130.3
49.000	0.043	192.4
50.000	0.032	138.6
51.000	0.045	160.1
52.000	0.036	96.12
53.000	0.018	156.6
54.000	0.035	103.2
55.000	0.014	125.9
56.000	0.031	115.4
57.000	0.027	134.9
58.000	0.017	166.5
59.000	0.013	130.5
60.000	0.019	132.8
61.000	0.018	191.1
62.000	0.009	137.5
63.000	0.020	113.0
64.000	0.022	136.7
65.000	0.017	147.5
66.000	0.013	161.1
67.000	0.014	183.7
68.000	0.006	162.7
69.000	0.008	100.8
70.000	0.008	68.48
71.000	0.005	102.0
72.000	0.005	126.8
73.000	0.020	126.9

Measured Response of Closed-loop Third Order System with Saturation (β = 50%)

Test Waveform - Modified PRBS

APPENDIX F

Time Samples of Instrument Test Signals

Spectra of Instrument Test Signals

1559	3821	2739	1044	904
1551	4024	3294	53	1041
2890	3051	2795	2357	1366
1752	1652	726	4050	593
100	1065	527	2595	85
1669	1527	2968	1037	1356
3809	2433	3596	1592	2826
2899	3202	1320	2047	3223
1137	3344	200	1087	3633
1110	2750	924	728	3970
1382	2114	808	1833	3149
1139	1927	708	2567	2529
1583	1799	2099	2021	3357
2143	1779	2284	1360	3682
1623	2298	459	1244	2455
1044	2340	473	972	1559
1197	1119	2535	273	1355
1306	816	2543	70	800
1681	2886	1204	1043	1299
3003	4087	2342	2442	3368
3829	1968	3994	3029	3567
3183	0	2425	2567	1126
2660	1088	285	1661	498
2886	2140	1195	892	2774
2283	887	2957	750	3894
829	440	2984	1344	3170
160	2311	2712	1980	3286
240	3244	2955	2167	3386
178	1965	2511	2295	1995
495	1022	1951	2315	1810
1455	1331	2471	1796	3635
1960	1847	3050	1754	3621
1617	2777	2897	2975	
1121	3887	2788	3278	
921	3450	2413	1208	
1390	1873	1091	7	
2351	1446	265	2126	
2187	2086	911	4095	
651	2344	1434	3006	
634	2611	1208	1954	
3050	3190	1811	3207	
4041	3053	3265	3654	
1737	2728	3934	1783	
44	3501	3854	850	
1499	4009	3916	2129	
3057	2738	3599	3072	
2502	1268	2639	2763	
2047	871	2134	2247	
3007	461	2477	1317	
3366	124	2973	207	
2261	945	3173	644	
1527	1565	2704	2221	
2073	737	1743	2648	
2734	412	1907	2008	
2850	1639	3443	1750	
3122	2535	3460	1483	

Stored Time Samples of Instrument Prime Composite

Test Waveform

F2

0	0	0	4095	0
4095	4095	0	0	4095
4095	0	4095	0	0
0	4095	0	4095	4095
4095	4095	0	0	4095
4095	0	4095	4095	0
0	4095	4095	4095	0
0	4095	0	4095	4095
4095	4095	4095	4095	0
4095	4095	0	0	0
4095	4095	0	0	4095
4095	0	0	4095	4095
0	0	4095	0	4095
0	4095	4095	4095	0
0	4095	4095	4095	0
4095	0	4095	0	4095
4095	4095	4095	0	4095
0	4095	0	0	4095
4095	4095	4095	4095	0
0	0	0	0	4095
4095	0	4095	0	0
4095	0	4095	0	4095
4095	4095	0	4095	0
0	0	4095	4095	4095
0	4095	0	0	4095
4095	4095	0	0	4095
0	4095	4095	4095	4095
0	0	0	4095	4095
0	4095	4095	0	4095
0	0	0	0	4095
4095	0	0	0	4095
4095	4095	4095	4095	4095
4095	4095	4095	4095	
4095	0	4095	4095	
0	0	4095	0	
4095	4095	4095	0	
4095	0	4095	0	
4095	4095	0	0	
0	0	0	4095	
4095	4095	0	4095	
4095	0	0	0	
4095	4095	0	0	
4095	0	0	0	
0	0	0	0	
4095	4095	4095	0	
0	0	4095	4095	
0	0	0	4095	
0	4095	4095	4095	
0	0	4095	0	
0	0	0	4095	
4095	0	4095	4095	
0	4095	0	0	
0	0	4095	0	
0	4095	0	0	
0	0	0	0	
0	0	0	4095	

Stored Time Samples of Instrument Modified PRBS

Test Waveform

MAGN	PHASE	SPECTRAL NO.	MAGN	PHASE	SPECTRAL NO.
6.559	57.049	1.0	2.683	-70.617	51.0
1.000	142.726	2.0	1.000	130.731	52.0
43124.109	72.936	3.0	43118.047	-145.723	53.0
1.000	-73.123	4.0	1.000	-87.187	54.0
43109.605	11.571	5.0	5.604	117.706	55.0
1.000	70.307	6.0	1.000	55.887	56.0
43122.078	165.252	7.0	10.542	-155.251	57.0
1.000	-145.887	8.0	1.000	-160.307	58.0
3.341	-129.408	9.0	43118.020	-114.781	59.0
1.000	-2.812	10.0	1.000	-16.877	60.0
43117.387	24.621	11.0	43130.718	104.075	61.0
1.000	139.269	12.0	1.000	127.274	62.0
43115.132	42.365	13.0	9.882	-135.298	63.0
1.000	-75.937	14.0	1.000	-90.000	64.0
6.155	82.580	15.0	2.617	-156.945	65.0
1.000	67.485	16.0	1.000	52.726	66.0
43120.348	179.890	17.0	43122.123	122.502	67.0
1.000	-148.884	18.0	1.000	-163.123	68.0
43109.448	-167.000	19.0	4.446	-124.863	69.0
1.000	-5.625	20.0	1.000	-19.693	70.0
1.820	-132.191	21.0	43116.554	128.145	71.0
1.000	135.281	22.0	1.000	124.113	72.0
43125.370	40.477	23.0	43119.775	152.959	73.0
1.000	-78.750	24.0	1.000	-92.812	74.0
0.454	103.846	25.0	2.140	-27.643	75.0
1.000	64.651	26.0	1.000	49.269	76.0
3.212	179.249	27.0	13.801	-117.954	77.0
1.000	-151.791	28.0	1.000	-165.937	78.0
43112.073	-133.769	29.0	3.601	138.583	79.0
1.000	-8.438	30.0	1.000	-22.515	80.0
43116.587	70.480	31.0	10.309	-77.275	81.0
1.000	139.656	32.0	1.000	121.116	82.0
7.759	148.335	33.0	8.627	-109.579	83.0
1.000	-81.562	34.0	1.000	-95.625	84.0
1.989	51.036	35.0	7.812	-27.029	85.0
1.000	61.791	36.0	1.000	45.281	86.0
43128.150	1.563	37.0	8.901	-97.022	87.0
1.000	-154.651	38.0	1.000	-168.750	88.0
7.022	93.865	39.0	6.322	-166.676	89.0
1.000	-11.250	40.0	1.000	-25.349	90.0
43115.126	-16.881	41.0	5.099	-19.926	91.0
1.000	134.719	42.0	1.000	118.209	92.0
43113.400	48.444	43.0	3.774	43.711	93.0
1.000	-84.375	44.0	1.000	-98.438	94.0
3.412	8.351	45.0	3.630	-149.178	95.0
1.000	58.884	46.0	1.000	49.656	96.0
43109.508	167.643	47.0	3.909	43.798	97.0
1.000	-157.485	48.0	1.000	-171.562	98.0
6.539	-64.744	49.0	3.719	176.284	99.0
1.000	-14.063	50.0	1.000	-28.209	100.0

Stored Spectrum of Instrument Prime Composite

Test Waveform

34684.495	-19.180	1.0	25235.960	32.335	51.0
34350.907	-17.962	2.0	16629.706	-8.003	52.0
34331.704	45.728	3.0	45319.564	-126.980	53.0
33282.797	-16.820	4.0	24859.087	-131.340	54.0
32423.360	146.912	5.0	39949.748	89.100	55.0
34737.454	48.627	6.0	21133.038	111.208	56.0
33060.733	51.890	7.0	40616.199	-144.877	57.0
32646.630	-17.374	8.0	29286.507	-118.037	58.0
31319.513	-96.235	9.0	39710.149	-27.467	59.0
34423.426	149.144	10.0	30763.628	-138.996	60.0
31640.803	135.018	11.0	26018.647	-60.445	61.0
35160.695	42.329	12.0	39297.749	14.450	62.0
32875.236	-7.453	13.0	24695.162	20.726	63.0
31182.639	64.185	14.0	14764.733	55.953	64.0
33688.952	-167.463	15.0	43991.409	-164.470	65.0
31622.967	-9.144	16.0	26812.270	-42.329	66.0
34723.017	163.763	17.0	23410.539	68.946	67.0
29593.374	-85.282	18.0	46731.755	-167.905	68.0
36905.814	86.369	19.0	24852.051	132.537	69.0
34173.439	159.258	20.0	32721.394	-21.905	70.0
31966.315	101.436	21.0	41877.469	-137.279	71.0
29202.803	156.570	22.0	32004.363	-47.145	72.0
27039.093	-144.688	23.0	16305.987	-40.491	73.0
35919.967	40.863	24.0	9234.221	-102.926	74.0
36919.428	-30.905	25.0	47069.048	-166.893	75.0
28118.439	-2.459	26.0	40926.059	134.514	76.0
34616.414	-136.138	27.0	37847.407	-110.926	77.0
31771.310	73.894	28.0	33272.256	-172.313	78.0
35540.752	-160.579	29.0	16138.837	-82.397	79.0
26425.864	-146.796	30.0	35107.837	161.798	80.0
39760.247	-22.641	31.0	30110.361	113.788	81.0
30156.118	-0.391	32.0	33700.324	-22.516	82.0
30191.083	-89.491	33.0	21506.244	-144.059	83.0
33019.056	169.064	34.0	46657.076	128.764	84.0
34150.872	-18.945	35.0	24360.225	56.856	85.0
26895.028	-82.801	36.0	40528.332	-137.629	86.0
16232.497	-57.298	37.0	21925.150	-25.553	87.0
44672.599	110.531	38.0	46389.572	-176.109	88.0
39847.855	177.009	39.0	22164.624	-156.885	89.0
29453.626	172.090	40.0	8446.167	-123.892	90.0
26857.758	-41.240	41.0	33200.424	110.631	91.0
42226.676	122.722	42.0	23883.685	-55.307	92.0
34225.939	-178.776	43.0	36685.153	26.672	93.0
27119.398	164.060	44.0	40853.083	103.408	94.0
22410.479	-135.731	45.0	24446.701	-106.126	95.0
11175.767	-131.815	46.0	46287.901	118.418	96.0
51202.643	45.462	47.0	28013.565	-151.823	97.0
35445.798	69.194	48.0	20538.254	95.932	98.0
23367.627	87.750	49.0	36454.995	-46.560	99.0
40096.525	-18.703	50.0	34666.715	47.489	100.0

Stored Spectrum of Instrument Modified PRBS

Test Waveform

28996.229	-160.160	101.0	30586.388	72.711	115.0
42421.310	62.062	102.0	39494.121	-51.906	116.0
20199.135	-19.866	103.0	46742.820	16.865	117.0
15269.947	144.710	104.0	14241.770	29.642	118.0
53129.816	-126.332	105.0	26226.146	-118.212	119.0
14951.023	-135.145	106.0	17284.642	165.320	120.0
47019.141	145.589	107.0	47490.572	98.496	121.0
15461.647	-79.021	108.0	22991.670	-60.848	122.0
50024.109	134.401	109.0	36159.028	144.676	123.0
24298.149	112.118	110.0	41270.394	36.181	124.0
19321.421	-146.389	111.0	13868.756	-99.776	125.0
8127.409	46.351	112.0	43570.940	99.672	126.0
45866.547	-70.042	113.0	36906.588	151.885	127.0
6154.216	-148.718	114.0	4095.000	-180.000	128.0

Magnitude	Phase	Harm No.	Magnitude	Phase	Harm No.
34683.441	-19.880	1.0	23620.303	-3.622	51.0
34347.285	-19.363	2.0	15523.436	-48.531	52.0
34323.930	42.555	3.0	42190.746	-163.425	53.0
33269.258	-19.624	4.0	23078.863	-167.652	54.0
32403.410	143.030	5.0	36984.348	49.214	55.0
34706.039	41.929	6.0	19508.348	71.978	56.0
33019.461	45.307	7.0	37384.102	174.751	57.0
32594.047	-22.991	8.0	26876.260	-158.565	58.0
31255.969	-102.538	9.0	36330.195	-68.706	59.0
34337.477	141.385	10.0	28058.543	-176.683	60.0
31545.160	131.483	11.0	23655.465	-103.284	61.0
35033.664	38.558	12.0	35615.367	-29.075	62.0
32735.213	-16.556	13.0	22306.334	-23.420	63.0
31028.859	53.862	14.0	13292.896	11.487	64.0
33499.203	-177.965	15.0	39472.949	149.867	65.0
31420.020	-20.356	16.0	23975.713	-92.270	66.0
34471.684	151.775	17.0	20860.332	22.046	67.0
29353.441	-97.888	18.0	41493.758	143.913	68.0
36572.191	73.062	19.0	21986.922	82.387	69.0
33831.797	144.799	20.0	28842.850	-70.912	70.0
31613.496	86.731	21.0	36777.012	176.542	71.0
28849.662	139.990	22.0	27999.496	-94.084	72.0
26681.422	-161.259	23.0	14210.338	-90.316	73.0
35402.754	28.549	24.0	8016.249	-154.716	74.0
36342.836	-45.939	25.0	40699.484	139.212	75.0
27643.141	-20.674	26.0	35245.438	78.828	76.0
33986.031	-157.265	27.0	32461.057	-164.842	77.0
31149.145	53.706	28.0	28419.447	136.124	78.0
34795.832	179.109	29.0	13726.755	-135.141	79.0
25833.037	-168.099	30.0	29734.234	105.774	80.0
38809.031	-48.363	31.0	25391.143	56.739	81.0
29386.824	-22.819	32.0	28293.641	-79.929	82.0
29372.688	-112.617	33.0	17975.410	157.256	83.0
32069.410	144.790	34.0	38822.113	68.975	84.0
33110.313	-46.785	35.0	20177.961	-2.382	85.0
26028.762	-108.016	36.0	33412.551	160.355	86.0
15680.033	-83.472	37.0	17991.184	-86.442	87.0
43070.867	83.911	38.0	37884.691	122.511	88.0
38344.102	149.546	39.0	18013.465	139.449	89.0
28285.525	143.465	40.0	6830.716	173.418	90.0
25738.367	-74.171	41.0	26718.395	43.798	91.0
40381.613	93.046	42.0	19124.490	-120.294	92.0
32659.246	151.006	43.0	29224.035	-39.628	93.0
25821.043	136.437	44.0	32377.926	38.475	94.0
21288.363	-169.620	45.0	19273.607	-172.656	95.0
10591.511	-162.392	46.0	36299.668	49.876	96.0
48411.711	15.017	47.0	21851.238	138.655	97.0
33431.211	36.152	48.0	15933.683	27.363	98.0
21984.617	52.722	49.0	28126.072	-118.075	99.0
37627.434	-53.050	50.0	26597.816	-20.735	100.0

Actual Spectrum of MPRBS Signal

Magnitude	Phase	Harm No.	Magnitude	Phase	Harm No.
22122.246	130.371	101.0	21398.283	-7.826	115.0
32179.934	-9.297	102.0	27443.891	-137.724	116.0
15233.894	-91.994	103.0	32258.926	-65.043	117.0
11449.672	72.341	104.0	9761.445	-52.352	118.0
39600.770	160.713	105.0	17849.184	158.520	119.0
11077.193	147.805	106.0	11680.485	81.279	120.0
34626.641	71.036	107.0	31865.875	13.755	121.0
11316.374	-154.620	108.0	15315.561	-146.049	122.0
36386.242	60.813	109.0	23910.963	58.828	123.0
17562.396	35.565	110.0	27089.025	-50.048	124.0
13876.771	131.412	111.0	9035.844	172.680	125.0
5799.366	-35.768	112.0	28169.521	11.429	126.0
32516.980	-149.014	113.0	23679.244	62.964	127.0
4335.046	133.317	114.0	2607.003	90.352	128.0

Actual Spectrum of MPRBS Signal

APPENDIX G

Survey of FFT based

Frequency Response Analysers

APPENDIX G

Specifications for FFT Analysers available in 1983

Analyser	Bandwidth	Resolution	Input Capture Circuitry	Waveform Generation Circuitry	Amp. Accuracy	Phase Accuracy
Takeda Riken TR 9403	DC to 25.6kHz	12 bits	Autoranged 30V peak to 1mV peak in 18 steps Impedance 1MΩ with 100 pf shunt.	N/A	0.5dB (5.9%)	$\pm 5°$
Schlumberger Instruments SI 1200	DC to 30kHz	12 bits	30V peak to 3mV peak Impedance 1MΩ with 50 pf shunt.	Long sequence length PRBS. Composite sine signal (crest factor <2, Shroeder phases) Impulse 0-SV peak	0.4dB (4.7%)	$\pm 2°$
Hewlett Packard AP 3582A	DC to 25.5kHz	12 bits	Adjustable 22.4V peak to 3mV peak in 10 steps Impedance 1MΩ with 95 pf shunt.	Short sequence length PRBS (Periodic), Random Noise (10mV to 500 mV),	0.8dB (9.6%)	$\pm 5°$