SIGNAL ANALYSIS BASED ON THE BINARY

FOURIER REPRESENTATION

by

Basil Al-Kayssi

This Thesis is submitted for the degree of

Master of Science

University of Edinburgh

December 1973

<u>CONTENTS</u>   <u>Page</u>

ACKNOWLEDGEMENTS

SUMMARY

The Walsh Functions are binary orthogonal functions whose values are restricted to $(+1, -1)$ and a generalised frequency can be associated with each function. These functions are ideal for digital processing techniques. An experimental Walsh Function digital spectrum analyser was constructed whose bandwidth covers the range 1 - 2048 zero crossings per second. This processor was connected to a P.D.P.-8 computer which acts as a back up store and also provided further processing capability. The results obtained from the experimental system confirmed the predictions derived from a theoretical survey of the use of Walsh functions applied to signal analysis. Due to the nature of the Walsh Functions the spectra of periodic functions in the Walsh Transform domain will be phase dependant. In the case of sinusoidal waveforms it is shown that this phase dependance can be minimised if successive Walsh power spectra are averaged. Based on the results obtained from the experimental analyser a simple Walsh power spectrum processor is proposed for the detection of low frequency signals.

GLOSSARY OF SYMBOLS

▷	Computer to T.T.L.
▷	T.T.L. to Computer
▷∘	Inverter
⟩∘	NAND
⟩	AND
⟩∘	NOR
⟩	OR
⟩⟩	Exclusive-OR

CHAPTER I

INTRODUCTION TO SPECTRUM ANALYSIS

1.1 Introduction

Spectrum Analysis is a fundamental signal processing operation of great utility in many branches of science. This operation may be performed numerically by special or general-purpose digital computers as well as continuously by analog computers or R.L.C. networks. Of special interest in many cases is the frequency domain description of signals and linear systems. Such descriptions are valid for both continuous and discrete processing. In areas such as speech communications, seismology, sonar, radar and medical technology, workers often make use of this signal processing technique. In spite of the usefulness of this technique its use was concentrated in the higher frequency range due to the limitations imposed by the instruments used.

1.2 Analog Spectrum Analysis

The simplest analog method of resolving a signal into its spectral components is to use a narrow band-pass filter and a signal averager to measure the portion of the signal passed by the filter. By this means an estimate of the magnitude of the sinosoidal component at the filter centre frequency is obtained. If the filter is tunable, spectrum analysis can be carried out over a range of frequencies. This is the principle of the wave analyser.

A basic limitation of the above analog technique which applies to all analogue spectrum analysers is the band-width of the filter.

A period of time equal to approximately the reciprocal of the filter pass band must be spent at each frequency to obtain an accurate estimate of the spectral content at that frequency. This limitation is not serious at high frequencies in the mega cycle range but at lower frequencies a single frequency measurement may take an unacceptably long time. If the signal to be analysed is random each reading must be averaged over a longer period of time to get a statistically smooth result. Measurements made in this way may correctly measure the spectrum at a certain frequency while missing a short term event at another. The measurement time-band width problem can be alleviated by using a bank of parallel filters. Such a method is effective when constant percentage bandwidth filters are used. In the case when higher resolution is needed necessitating the use of constant bandwidth filters the above method of using a parallel bank of filters proves impractical.

An alternative approach to using a parallel bank of filters with narrow constant bandwidths is a hybrid technique using a mixture of digital and analog circuitry. In this method the signal is lowpass filtered to prevent aliasing, then the output of the filter is sampled and digitised by an analog to digital converter at a rate f_s which is at least twice the cut-off frequency of the low-pass filter. The digitised samples are loaded into a memory and after N samples have been taken over a period $T = N/f_s$ the data is read out of memory non-distructively at a much higher rate and applied to a digital to analog converter. The output of the D/A converter is hetrodyned with an accurate voltage controlled oscillator and applied to a fixed narrow band-pass filter which feeds a signal averager. The output of the signal averager at the end of the N

sample scan constitutes one spectral estimate.   If the frequency

of the controlled oscillator is stepped at the end of each complete

scan then a number of spectral points can be formed.   The time to

compute all the spectral points might be greater than the period

between samples which leads to an effective dead band between T-

second records and the loss of signal information.   The problem of

differing response times in the case of the percentage bandwidth

analyser is eliminated since only one fixed bandwidth filter is

used and the spectral estimation is performed at higher frequencies.

1.3 Digital Spectrum Analysis

The introduction of the Fast Fourier Transform Algorithm

has made possible the computation of an N-length Discrete Fourier

Transform proportional to $N \log_2 N$ (when N is a power of 2)

mathematical operations instead of N^2 operations.   With the

development of faster computers and hardware this has lead to real

time digital spectrum analysers covering the low frequency end of

the spectrum.   Like the hybrid analyser described previously the

all-digital analyser samples the input signal at a rate f_s for T

seconds and stores $N = Tf_s$ samples in its memory.   The highest

frequency that can be resolved without ambiguity is $\frac{1}{2} f_s$, this is

the Nyquist folding frequency.   The frequency resolution is

determined by the record length T where $f = \frac{1}{T}$.   This resolution

is a result of the fact that for a record length of T seconds no

discrete Fourier transform can be defined with a resolution smaller

than $\frac{1}{T}$.   The digital analyser transforms the N samples in memory

to a set of $\frac{N}{2}$ Fourier coefficients spaced $\frac{1}{T}$ cycles per seconds apart

which may be processed to obtain the magnitude and phase of the

spectrum or the power spectrum.   In the case of random signals the

results of several sample records can be summed to form a smoothed
spectral estimate, this is an advantage of digital computation of
the spectrum which eliminates the use of analog integrators with
their attendant limited dynamic range and linearity.

1.4 Digital Spectrum Analysis Based on Walsh Functions

With the increasing use of digital techniques in signal
processing an interest has developed in representing signals by
means of orthogonal functions other than the traditional sine-
cosine functions. However for ease of implementation some useful
property is required before a system of functions merits use.
Two such properties are the existence of a fast transform algorithm
and the restriction of the function values to $[0, 1]$. One of the
best known examples of orthogonal functions having these properties
are the Walsh functions. As will be shown in Chapter 2 the
Discrete Walsh Transform performs additions and subtractions of the
input signal samples to generate the Walsh coefficients this feature
of the transform results in considerable computational savings in
the Fast Walsh Transform compared to the Fast Fourier Transform
and leads to a simpler hardware structure since there is no need
for a complex multiplier. An interesting feature of the Walsh
transform is that the description of a time and sequency[*] limited
signal in the Walsh transform domain is sequency limited, this is in
marked contrast to the description of the same type of signal
(time and frequency limited) in the Fourier transform domain.

[*]Sequency is a generalised frequency associated with Walsh functions,
see Chapter 2 for further details.

Due to the binary nature of Walsh functions they will be ideal for use in circuits based on binary digital components, as the sine-cosine functions are for circuits based on R.L.C. components. Simple filters based on Walsh functions have been realised and signals filtered by Walsh filters are indistinguishable from those that are filtered by ordinary filters. Experimental speech vocoders[5] have been built which give useful bandwidth reduction. The description of visual information in the Walsh transform domain leads to similar bandwidth reduction.[15]

1.5 Program of Work

The aim of the work undertaken was to investigate the advantages of a digital Walsh spectrum analyser for simplicity of construction and usefulness of the displayed spectrum. Specifically this analyser could be used to detect the dominant periodic component in an E.E.G. signal.[*] The major objectives of this were:-

(a) Study of the theory of Walsh functions.

(b) Study of the practical application of Walsh functions with particular reference to low cost hardware for spectrum analysis.

(c) Design and construction of an experimental system to enable spectrum analysis based on Walsh functions to be evaluated.

(d) Presentation of experimental evidence of the performance of the system.

[*]H.R.A. Townsend[27] has shown that knowledge of this frequency can be used to assess the progress of a deteriorating liver condition. A low cast dominant rhythm analyser for ward use would be a useful

(e) Devise a low cost system for detecting the dominant
periodic term in a noisy signal.

It was intended that the experimental analyser system should
cover as large a sequency bandwidth as possible as well as operating
in real time. Software methods were considered for use to
implement the transformation in conjunction with a P.D.P.-8 computer,
data being entered via the priority interrupt channel. This was
ruled out due to the relatively long time (5 micro-seconds) of the
basic addition/subtraction operation. The analyser constructed
(described in Chapter 3) computes 50 pairs of coefficients from a
possible 2048 pairs. The computed coefficients are transferred
to the computer through the data break channel of a P.D.P.- 8
computer where further processing is performed and the results
displayed on an oscilloscope. The analyser was designed to minimise
storage requirements and to reduce the overall system complexity
by utilising as far as possible commercially available M.S.I.
circuits. These design objectives and the condition that the
analyser should work in realtime resulted in the final design not
being based on the Fast Walsh Transform algorithm. The sequency
bandwidth of the proposed analysis system covers the major Walsh
coefficients of frequencies in the range 1 – 16 c/s. It is shown
in Chapter 5 that the proposed analyser design is optimum for
the previously mentioned frequency range although this might not be
the case for a wider frequency range.

A method for calculating the Walsh coefficients of sinusoidal
functions was developed and is presented in Appendix B. The nature
of the Walsh functions makes the sequency spectra of frequencies which

are not integer to the analyser time base phase dependant, this problem was overcome by means of multiple averaging of consecutive spectra to minimise the phase dependance. Walsh amplitude and power spectra of a range of signals were obtained from the experimental system. Finally an E.E.G. wave-form was simulated and its Walsh power spectrum recorded. The Walsh power spectrum of the simulated E.E.G. wave-form shows a peak at the sequency corresponding to the dominant frequency in the wave-form.

CHAPTER 2

WALSH FUNCTION THEORY

2.1 Introduction

Walsh functions were first introduced in a paper by J.L. Walsh[28] and the mathematical background of these functions is well documented.[10] With the introduction of economical digital components interest has increased in applying Walsh functions to communications and other electronics fields. Gibbs[3,4] has shown that Walsh functions can be obtained as solutions to a logical differential equation this leads to a useful tool for describing systems in the Walsh transform domain.[13] The theory of Walsh functions shows similarities to and divergences from the Fourier analysis. A brief outline of the theory of discrete Walsh functions is presented in this Chapter.

2.2 Discrete Walsh Functions and the Discrete Walsh Transform[3]

The discrete Walsh functions $\text{Wal}(k,t)$ may be defined in terms of the binary representation of the arguments (k,t) where the N length discrete Walsh function may be defined for $N = 2^n$ by

$$\text{Wal}_n(k,t) = \exp j\pi[k_{n-1}t_0 \oplus \bigoplus_{r=1}^{n-1} (k_{n-r} \oplus k_{n-r-1})t_r] \qquad (1)$$

for $K, t = 0,1,2\ldots,N-1$,

where $\oplus$ denotes modulo 2 addition

and $\displaystyle\bigoplus_{r=0}^{n-1}$ denotes modulo 2 summation over n terms.

The symbols K_r and t_r are the binary bits of K,t i.e. $t = \displaystyle\sum_{r=0}^{n-1} t_r \cdot 2^r$, $K = \displaystyle\sum_{r=0}^{n-1} K_r \cdot 2^r$.

The above definition generates the Walsh functions in increasing number of sign changes.   This is illustrated by the 8-length discrete Walsh functions which are represented as an 8 x 8 matrix in Figure 2-1.

Using equation (1) and from the matrix in Figure 2-1 the following properties can be deduced

(1) Walsh functions are symmetric

$$\text{Wal}_n(k,t) = \text{Wal}_n(t,k)$$

(2) The product of two Walsh functions is another Walsh function of the same set given by:-

$$\text{Wal}_n(\ell,t).\text{Wal}_n(k,t) = \text{Wal}_n(\ell \oplus k, t)$$

where $\ell \oplus k$ is the modulos-2 addition of the corresponding bits in the binary representation of ℓ and k respectively.

(3) A generalised frequency can be associated with the rows of the matrix in Figure 2-1, this is termed sequency (7) and is equal to $\frac{1}{2}$ (average number of sign changes of the periodic function).   In the case of equation 1 k is directly related to the number of sign changes of the corresponding Walsh function hence the corresponding sequency is $\frac{k}{2}$ for even k and $\frac{k+1}{2}$ for odd k.   For odd k the corresponding Walsh function is termed $\text{sal}(\frac{k+1}{2}, t)$ and for even k it is termed $\text{cal}(\frac{k}{2}, t)$.   It is seen from Figure 2-1 that there are three pairs of Walsh functions having the same sequency.   For a set of $N=2^n$ Walsh functions the number of pairs will be $\frac{N}{2} - 1$.

t =	0 1 2 3 4 5 6 7	sequency	gray code of k	Decimal bit reversed gray code
k				
0	1 1 1 1 1 1 1 1	0	000	0
1	1 1 1 1 − − − −	1	100	4
2	1 1 − − − − 1 1	1	110	6
3	1 1 − − 1 1 − −	2	010	2
4	1 − − 1 1 − − 1	2	011	3
5	1 − − 1 − 1 1 −	3	111	7
6	1 − 1 − − 1 − 1	3	101	5
7	1 − 1 − 1 − 1 −	4	001	1

Figure 2-1

8 x 8 Sequency Ordered Walsh Matrix

t =	0 1 2 3 4 5 6 7	sequency
ρ		
0	1 1 1 1 1 1 1 1	0
1	1 − 1 − 1 − 1 −	4
2	1 1 − − 1 1 − −	2
3	1 − − 1 1 − − 1	2
4	1 1 1 1 − − − −	1
5	1 − 1 − − 1 − 1	3
6	1 1 − − − − 1 1	1
7	1 − − 1 − 1 1 −	3

Figure 2-2

8 x 8 Hadamard Matrix

(4) $\text{Wal}_n(0,t) = 1$ for all t defined for equation 1.

(5) It follows from (2) and (3) that

$$(\text{Wal}_n(k,t))^2 = 1 \text{ for all t defined for equation 1.}$$

(6) From (2) and (3) the Walsh functions are orthogonal since

$$\sum_{t=0}^{N-1} \text{Wal}_n(k,t).\text{Wal}_n(\ell,t) = \sum_{t=0}^{N-1} \text{wal}(k\oplus\ell,t) = \begin{cases} N & k=\ell \\ 0 & k\neq\ell \end{cases}$$

(7) $\sum_{k=0}^{N-1} \text{Wal}_n(k,t) = \begin{cases} N & t=0 \\ 0 & t\neq0 \end{cases}$ this can be generalised to

$$\sum_{k=0}^{N-1} \text{Wal}_n(k,t\oplus t') = \begin{cases} N & t=t' \\ 0 & t\neq t' \end{cases}$$

(8) The set of Walsh functions $\text{Wal}_n(0,t),\ldots,\text{Wal}_n(N-1,t)$
generated by equation 1 is closed with respect to modulo-2
addition since for any k and ℓ in the set $0,1,2\ldots,N-1$
the summation $(K\oplus\ell)$ will generate another member of the
set, hence from (2) above the Walsh function $\text{Wal}_n(k\oplus\ell,t)$
is another member of the defined set of Walsh functions.

(9) It can be shown[6] that the following relationships hold
between Walsh functions defined over a finite normalised
interval θ and the Walsh functions defined over an infite
interval:-

$$\text{Wal}(\mu,\theta) = \text{Wal}(0,\theta) \quad 0 \leq \mu < 1$$

$$\text{cal}(\mu,\theta) = \text{cal}(i,\theta) = \text{Wal}(2i,\theta) \quad i \leq \mu < i+1$$

$$\text{sal}(\mu,\theta) = \text{sal}(i,\theta) = \text{Wal}(2i-1) \quad i-1 < \mu \leq i$$

where θ is the interval $-\tfrac{1}{2} \leq \theta \leq \tfrac{1}{2}$

μ is a real number, i is an integer.

From the properties stated the N-length $(N=2^n)$ discrete Walsh functions form a complete orthogonal set for the N-length real sequence $f(t)$ and the discrete Walsh transforms of this sequence is defined as

$$F(k) = \frac{1}{N} \sum_{t=0}^{N-1} f(t) \, \text{Wal}_n(k,t); \quad k = 0,1,2.....,N-1$$

where the inverse transform is

$$f(t) = \sum_{k=0}^{N-1} F(k) \, \text{Wal}_n(t,k) \; ; \quad t=0,1,2....,N-1$$

since $\displaystyle\sum_{k=0}^{N-1} F(k) \, \text{Wal}_n(t,k) = \sum_{k=0}^{N-1} (\frac{1}{N} \sum_{\rho=0}^{N-1} f(\rho) \, \text{Wal}_n(k,\rho) \cdot \text{Wal}_n(t,k)$

$$= \sum_{\rho=0}^{N-1} f(\rho) \, (\frac{1}{N} \sum_{K=0}^{N-1} \text{Wal}_n(t,k) \cdot \text{Wal}_n(k,\rho))$$

$$= \sum_{\rho=0}^{N-1} f(\rho) (\frac{1}{N} \sum_{K=0}^{N-1} \text{Wal}_n(k,t \oplus \rho))$$

the summation in brackets will equal one only when $t=\rho$ as stated in property (6), therefore

$$F(k) = \frac{1}{N} \sum_{k=0}^{N-1} f(t) \, \text{Wal}(k,t) \tag{2}$$

$$f(t) = \sum_{k=0}^{N-1} F(k) \, \text{Wal}(t,k) \tag{3}$$

are transform pairs.

Referring to Figure 2-1 it is seen that for an even K the Walsh function in the corresponding row is symmetrical about the mid-point of $t = 0,1,...7$, and for odd k the corresponding Walsh function is

skew symmetric, this applies also to even and odd t.   Therefore a
sequence f(t) will have a transform which is composed of the
coefficients of skew symmetric Walsh functions if it is skew
symmetric about its mid-point and vice-versa.

The transform pair in equations (2) and (3) can be represented
in matrix form as:

$$[f_n(t)] = [W] [F(k)] \tag{4}$$

$$[F_n(k)] = \frac{1}{N} [W] [f_n(t)] \tag{5}$$

where $[F(k)]$ and $[f(t)]$ are column matrices and $[W]$ is a square
matrix whose rows or columns are the N-length discrete Walsh functions
ordered in increasing number of sign changes.   The rows of the matrix
$[W]$ can be rearranged to form the Hadamard matrix $[H(n)]$ which has
a simple recursion structure given by

$$[H(n+1)] = \begin{bmatrix} H(n) & H(n) \\ H(n) & -H(n) \end{bmatrix} = [H(n)] \otimes [H(1)]$$

where $[H(1)] = \begin{bmatrix} 1 & 1 \\ 1 & - \end{bmatrix}$, $\otimes$ denotes kronecker multiplication and $[H(o)]=1$.
Figure 2-2 shows an 8 x 8 Hadamard matrix $[H(3)]$ and it can be shown
that the bit reversed grey code of the numbers indicating the rows
of the matrix $[W]$ indicate the same Walsh functions in the rows of
the matrix $[H(n)]$ as shown in Figure 2-2.

By analogy with linear systems theory a convolution type of
operation(3) can be defined for two N-length ($N=2^n$) sequences f(t) and
g(t) as follows:-

$$z(t) = f_n(t) \circledast g_n(t) = \frac{1}{N} \sum_{y=0}^{N-1} f_n(y) \, \dot{g}_n(t \oplus y) \tag{4}$$

this type of operation has the same property in Walsh transform theory as does convolution in Fourier transform theory since taking the Walsh transform of both sides of equation 4 we will have

$$Z(k) = \frac{1}{N} \sum_{t=0}^{N-1} z(t) Wal_n(k,t) = \frac{1}{N} \sum_{t=0}^{N-1} \left(\frac{1}{N} \sum_{y=0}^{N-1} f(y) \, g(t \oplus y) \right) Wal_n(k,t)$$

$$= \frac{1}{N} \sum_{y=0}^{N-1} f(y) \left(\frac{1}{N} \sum_{t=0}^{N-1} g(t \oplus y) Wal_n(k,t) \right) \tag{6}$$

now for any member x of the set $[0,1,\ldots N-1]$ where $N=2^n$, the operation $x \oplus y$ for y a constant in the set will generate another member of the set here the condition $\sum_{x=0}^{N-1} f(x \oplus y) = \sum_{x=0}^{N-1} f(x)$ for an N-length sequence $f(t)$ is valid. Using the above condition, the bracketed term in equation 6 can be rearranged to give

$$Z(k) = \frac{1}{N} \sum_{y=0}^{N-1} f(y) \left(\frac{1}{N} \sum_{t=0}^{N-1} g(t) Wal_n(k, t \oplus y) \right)$$

$$= \left(\frac{1}{N} \sum_{y=0}^{N-1} f(y) . Wal_n(k,y) \right) \left(\frac{1}{N} \sum_{t=0}^{N-1} g(t) Wal_n(k,t) \right) \tag{7}$$

since $Wal_n(k,t) . Wal_n(k,y) = Wal_n(k, y \oplus t)$

hence equation 6 gives

$$Z(k) = G(k) . F(k)$$

where $G(K)$, $F(K)$ and $Z(K)$ are the Walsh transforms of $g(t)$, $f(t)$ and $z(t)$ respectively. This operation is termed dyadic or logical convolution.

Following similar procedures a dyadic auto correlation
function $\ell_n(t)$ can be defined for an N-length sequence $f_n(t)$
(3)   given by

$$\ell_n(t) = f_n(t) \otimes f_n(t) = \frac{1}{N} \sum_{\gamma=0}^{N-1} f_n(t) \cdot f_n(t\oplus\gamma) \qquad (8)$$

and taking the Walsh transform of both sides of equation 8 we will
have

$$L_n(k) = (F_n(k))^2 \qquad (6)$$

where the right side of equation 9 is the discrete Walsh power
density spectrum. This relationship is the analogue of the Wiener-
Khinchin relationship in the Walsh transform domain. Hence the
Walsh power spectrum of a sequence f(t) is the Walsh transform of
its dyadic auto correlation function.

A discrete delta function $\delta_n(t,\gamma)$ can be defined for Walsh
functions since the Walsh transform of an N-length sequence N,0,0....0
is 1. Therefore dyadically convolving a sequence f(t) with the Walsh
delta function will produce the shifted sequence $f(t\oplus\gamma)$ i.e.

$$f(t) \otimes \delta_n(t,\gamma) = \frac{1}{N} \sum_{p=0}^{N-1} f(t\oplus p)\, \delta_n(\gamma,p)$$

$$= \frac{1}{N} \sum_{p=0}^{N-1} \delta_n(\gamma\oplus p, t)\, f(p)$$

$$= f(t\oplus\gamma)$$

$$(1,6,15,21,23)$$

2.3 Fast Walsh Transform

To compute all the Walsh coefficients of an N-length sequence ($N=2^n$) it usually takes $N(N-1)$ additions and subtractions, but due to the fact that the Walsh or Hadamard matrices can be decomposed into a product of sparser matrices, the number of operations needed is $N \log_2 N$ as will be shown later. One of the decompositions which can be simply implemented in terms of hardware[11] is due to C.K. Rushforth[21] and is an adaption of a Fast Fourier Transform algorithm due to Peace. The method proposed by Rushforth is based on the decomposition of the Hadamard matrix $[H(n)]$ and he has shown that $[H(n)]$ can be represented as a product

$$[H(n)] = ([H(1)] \otimes [I(1)] \otimes [I(1)] \ldots \otimes [I(1)])([I(1)] \otimes [H(1)] \otimes [I(1) \ldots$$

$$[I(1)])([I(1)] \otimes \ldots \otimes [H](1)) \qquad (10)$$

where $[H(1)] = \begin{bmatrix} 1 & 1 \\ 1 & - \end{bmatrix}$, $\otimes$ denotes kronecker multiplication, and $[I(1)]$ is the identity matrix $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Each bracket in the above expression for $[H(n)]$ contains n matrices and there are n brackets. The expression for $[H(n)]$ can be reduced further by using a perfect shuffle matrix P which on premultiplying a matrix with an even number of rows will perfectly inter-leave the rows of its two halves, an example of this for the case of a four element column matrix is

$$[\rho] \cdot \begin{vmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{vmatrix} = \begin{vmatrix} x_0 \\ x_2 \\ x_1 \\ x_3 \end{vmatrix}$$

where in this instance the matrix $[P]$ will be equal to

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using the shuffle matrix $[P(n)]$ and denoting the first factor of equation 10 by the matrix $[C(n)]$ it can be shown that the successive factors of equation 9 are:-

$$([I(1)] \otimes [H(1)] \otimes [I(1)] \ldots \otimes [I(1)]) = [P(n)][C(n)][P(n)]^{-1}$$

$$([I(1)] \otimes [I(1)] \ldots \ldots \otimes [H(1)]) = [P(n)]^{n-1}[C(n)][P(n)]^{-(n-1)}$$

substituting the above factors in equation 9 and noting that $[P(n)]^n = [I(n)]$ = identity matrix, equation 10 will be

$$[H(n)] = [C(n)].[[P(n)].[Cn][P(n)]]^{-1}.[[P(n)]^2[C(n)][P(n)]]^{-2} \ldots$$

$$.([P(n)]^{n-1}[C(n)][P(n)]]^{-(n-1)}$$

which reduces to $[H(n)] = ([C(n)].[P(n)])^n$ on making the substitution $[P(n)]^{-n} = [I(n)]$. The factored form of $[H(n)]$ can be substituted for the matrix form of the Walsh transform in equation 5 which will be

$$[F(k)] = \frac{1}{N}([C(n)][P(n)])^n.[f(t)] \quad t,K = 0,1,\ldots N-1; \quad N=2^n$$

In the case of an 8-length sequence the matrix $[C(n)] = [C(3)] =$

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & - & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & - & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & - & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & - \end{bmatrix}$$

The product $[C(2)][P(3)]$ will shuffle the matrix $[f(t)]$ and perform 8 subtract-add operations on the shuffled elements, this operation is performed three times hence the total number of operations is 24 as opposed to $8 \times 7 = 56$. In the general case of an N-length sequence $(N=2^n)$ the number of operations performed will be $N \log_2 N = n.2^n$.

2.4 Discrete Walsh Power Spectra

It has been shown in section 2 (equations 8 and 9) that the Walsh transform of the dyadic auto-correlation function of an N-length real sequence $f_n(t)$ is its power density spectrum. Using the notation of equation 5 where

$$[F_n(k)] = \frac{1}{N} [W_n] [f_n(t)] \qquad (5)$$

the Walsh power density function will be $B_n(k) = (F_n(k))^2$ (11)

$K = 0,1,...N-1$; $N=2^n$. Denoting the coefficients of even Walsh functions by $C(s)$ and the coefficients of the odd Walsh functions by $S(s)$ where s denotes sequency and $s = \begin{cases} \frac{k+1}{2}, & k \text{ odd} \\ \frac{k}{2}, & k \text{ even} \end{cases}$

There will be $\frac{N}{2} - 1$ spectral points $P_n(s)$ where

$$P_n(s) = C^2(s) + S^2(s) \text{ for } 0 < S < \frac{N}{2}$$

The zero sequency term is $C^2(o)$ and the highest sequency term is $S^2(\frac{N}{2})$ for $S = \frac{(N-1) + 1}{2}$. The average energy of the sequence is preserved in the transform domain since by transposing equation 5 and multiplying by 5 we will have

$$[F_n(k)]^T \cdot [F_n(k)] = \frac{1}{N^2} [f_n(t)]^T \cdot [W] \cdot [W_n][f_n(t)]$$

$$= \frac{1}{N} [f_n(t)]^T \cdot [f_n(t)]$$

since $[W]$ is its own transpose and $[W] \cdot [W] = N[I(n)]$

therefore $N \cdot \sum_{K=0}^{N-1} (F_n(k))^2 = \sum_{t=0}^{N-1} (f_n(t))^2$ (12)

The Fourier series describing a continuous functions $x(\theta)$
in the interval $\frac{-T}{2} \leqslant \theta \leqslant \frac{T}{2}$ can be represented as

$$x(\theta) = \sum_{i=-\infty}^{+\infty} C_i e^{\frac{j2\pi i\theta}{T}}$$

where $C_i = (\sqrt{A_i^2 + B_i^2}) \cdot e^{j\emptyset}$; .... (13)

$\emptyset = \tan^{-1} \frac{Ai}{Bi}$, Ai = coefficient of $\sin \frac{2\pi i\theta}{T}$; B_i = coefficient
of $\cos \frac{2\pi i}{T} \theta$ and the frequency power density of $x(\theta)$ will be given
by $C_i \overset{*}{C_i}$ ($\overset{*}{C_i}$ denotes complex conjugate). The conditions stated above
will hold for the discrete Fourier transform hence the frequency
power spectral density computed by the discrete Fourier transform
method will be invariant to cyclic shifts of the input data samples.
In the case of the Walsh transform the above property does not hold
since there is no comparable addition formula [6,7] to that of equation 13
in Walsh transform theory. Therefore the power density spectral
points will vary with circular shifts of the input sequence i.e. the
Walsh power spectrum of periodic signals is phase shift dependent.
A dyadic shift of the input sequence $f(t)$ to $f(t \oplus t')$ will affect the
sign of the coefficients but not their magnitude, hence the power
density spectral points will be invariant to dyadic shifts.[14] A Walsh

power spectrum which is invariant to circular shifts of the input
data samples has been developed[12,1] and is called the odd harmonic
spectrum since it sums the power density spectral points of fundamental
sequencies and all their odd harmonics which results in n + 1
invariant spectral sums. It has been shown[1] that the circular
shift invariant sums are directly related to the structure of the
matrix yielding these circular shifts and the spectral sums are given
by the equation (using the Hadamard matrix in equation 5)

$$P(q) = \sum_{k=2^{q-1}}^{2^q - 1} (F_n(k))^2 \qquad q = 0, 1, \ldots, n$$

for an N-length input sequence $(N=2^n)$. Therefore although this
power spectrum is shift invariant its sequency resolution is not
complete. A further development of the odd harmonic spectrum[12] yields
$\frac{N+1}{2}$ invariant power density spectral points but would be too complex
for a simple hardware realisation due to the increased number of
cross products as the input sequence length increases.

For the case of a signal mixed with noise it is necessary to
average the power spectrum over a number of sequence lengths to smooth
out the effects of the added noise. In Chapter 4 and Appendix B
it is shown that this averaging process leads to a minimisation of the
effects of phase shift.

CHAPTER 3

EXPERIMENTAL WALSH FUNCTION SPECTRUM ANALYSER

3.1 Introduction

There are various algorithms for computing the Fast Walsh Transform in sequency or binary order.   All these algorithms assume the simultaneous availability of the input data samples and this implies the use of at least double the storage space if the system was to be used in real time.   Apart from the foregoing these algorithms are not economical in terms of the use of computing elements which could equal the number of samples used in the computation if each iteration was to be carried out in parallel.

An economical method of implementing a hardware processor is one which is based on a decomposition algorithm utilising a perfect shuffle concept.[11]   If this processor is used in real time it will need three stores, two of which will take in the input data samples alternately with the third used as a partial sums store. To cut down on the number of adder/subtractor units needed a serial method of computation is adopted;   one A/S unit is used in conjunction with shift registers for the storage medium as shown in Figure 3.1.   It should be noted that the detailed control logic for connecting the various shift registers to the A/S has not been shown.   The data is entered in either shift register one or two.   After N samples have been loaded in say shift register 1, the multiplexer starts loading shift register 2.   When shift register 1 has been loaded with N data samples the algorithm is carried out between it and the partial sums shift register, the ideal shuffle is realised by taking the data samples from the centre of the shift

```
┌─────────────────────────────────┐
│       shift register-2          │◄─────┐
└─────────────────────────────────┘      │
                                    ┌──────────┐   I/P
                                    │  Multi-  │◄─────
                                    │  plexer  │
┌─────────────────────────────────┐└──────────┘
│       shift register-1          │◄─────┘
└─────────────────────────────────┘
   │                        │
   │                        ▼
   │                  ┌──────────┐
   └─────────────────►│  Adder-  │
                      │Subtractor│
                      └──────────┘
                           │
                           ▼
                      ┌──────────┐
┌─────────────────────────────────┐
│ Partial sums shift register     │
└─────────────────────────────────┘
```
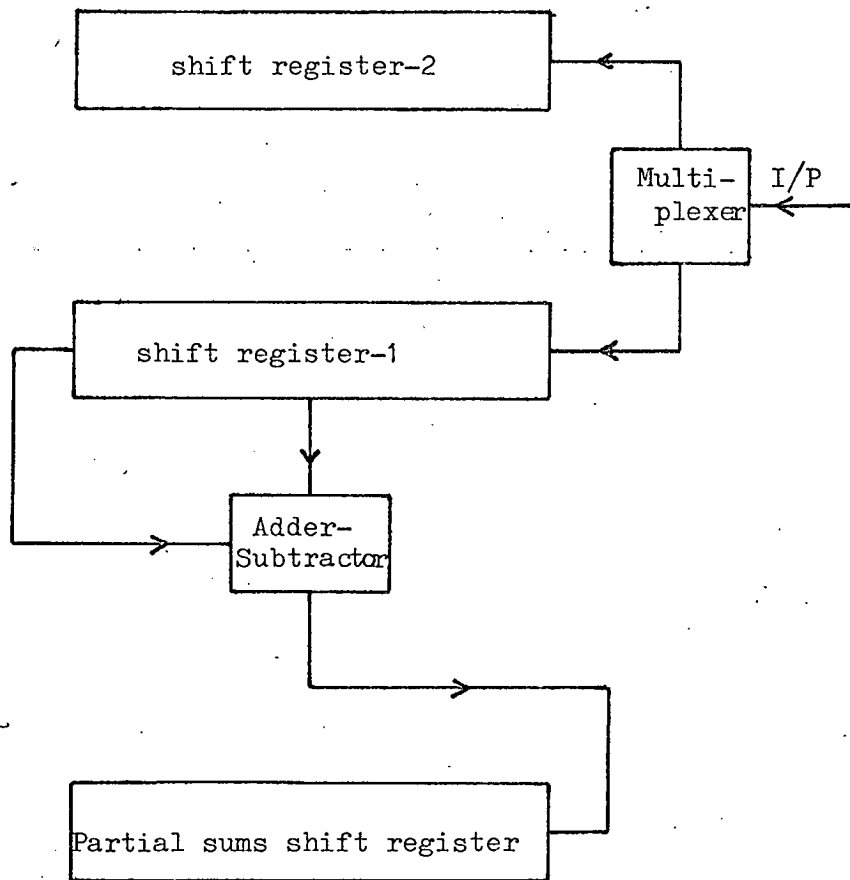
Figure 3-1

Hardware realisation of F.W.T. algorithm

register and the end.   For each pair of data words entering the A/S an addition and subtraction is performed in sequence and the results entered in the partial sums shift register.   At the end of N mathematical operations 1 iteration cycle would have been completed.   During the next iteration cycle the partial sums shift register is connected to the A/S and the stored data will be processed in the manner described above;   the results of the second iteration are stored in shift register 1.   After $n = \log_2 N$ iterations the Walsh coefficients will have been computed.   From the foregoing brief description of a representative system it can be seen that it is inefficient in terms of storage space since two thirds of it is idle between computation cycles.

3.2 Alternative Realisation of a Walsh Transform Processor

The equation describing a Walsh coefficient is a sum of the product of the data samples and the value of the relevant Walsh function at the instant the sample is taken

$$F(K) = \frac{1}{2^n} \sum_{t=0}^{2^n-1} f(t) \, \text{Wal}(K,t) \tag{1}$$

F(K) can be computed as a partial sum of the samples multiplied by the value of the Walsh function at each instant of sampling.   At the end of the period of definition of the Walsh function the final sum is equal to the relevant coefficient as selected by the value of K.   A hardware realisation of equation 1 would be optimum in terms of storage space and the speed of calculation of each partial sum.

A transform processor based on equation 1 was designed and built. This includes a Walsh function generator which can be scanned through a fixed sequency band with a preselectable starting sequency. The sequency band was chosen to cover the significant harmonics in the sequency spectrum generated by sinosoidal frequencies in the range one to sixteen cycles per second.

To make use of the processing capability of a digital computer the system was connected to a P.D.P.8 through the data break channel. The choice of the sequency band and the timing requirements of the transfer operation sets an upper limit on the sequency that can be processed in the system by limiting the sub-cycle of the highest sequency Walsh function.

A block diagram of the complete data processing system is shown in Figure 3.2. The analogue to digital converter is preceeded by a low pass sequency filter that forms a step approximation of the input signal thus limiting the highest sequency presented to the system. The conversion command to the A.D.C. occurs in the middle of each output step from the low pass sequency filter. At the end of the conversion cycle the data word is presented to one set of inputs of the arithmetic unit, the other set of arithmetic unit inputs are connected to the shift register stack forming the serial word store. By clocking the sequency register of the Walsh function generator and the shift register stack in synchronism a new set of partial sums will be formed. At the end of the period of definition of the Walsh functions a data break request signal is generated signalling the computer to enter the break state. The shift
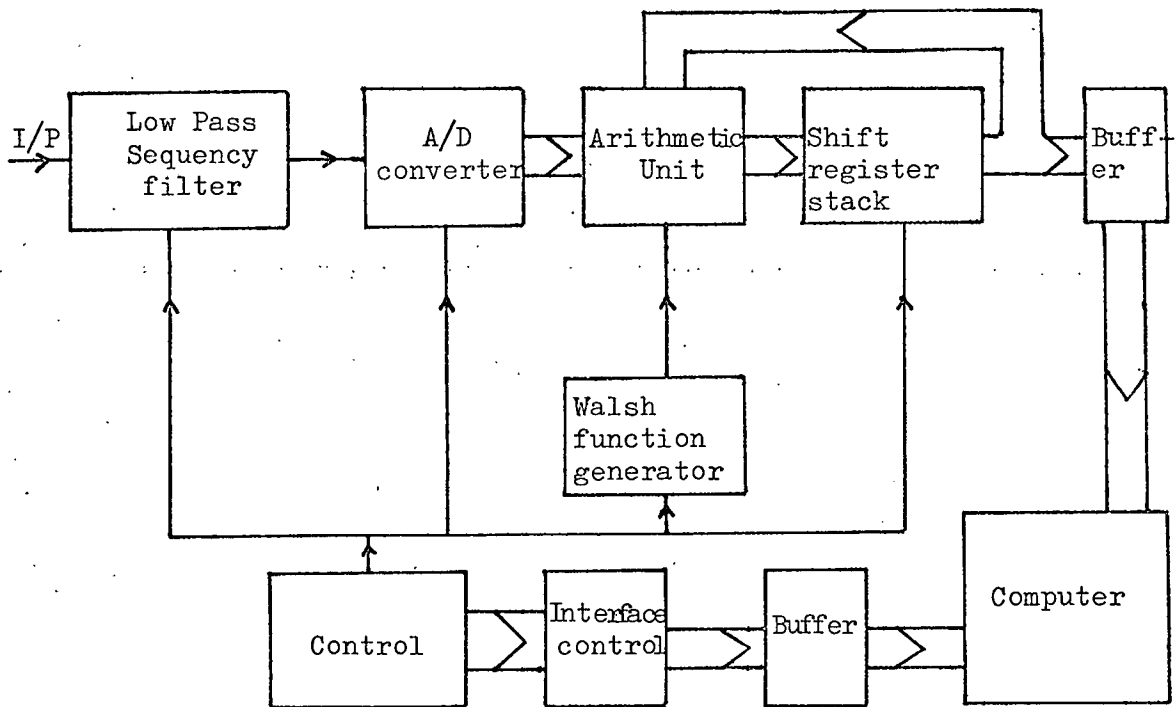
Figure 3-2

System block diagram

register stack will be cleared as the coefficients are transferred

to the computer and the new cycle starts at the end of the

transfer operation.

3.3 Low Pass Sequency Filter (L.P.S.F.)

The Walsh series representation of a function defined over

a specified interval is a step wise approximation of that function.

The width of the step defines the highest sequency Walsh function

present in the series. Harmuth[6,8] has proposed the simple circuit

shown in Figure 3.3 using one integrator and a sample and hold

amplifier to realise a low pass sequency filter. In this circuit

the period of integration is defined by the cut-off sequency of the

filter. Sampling occurs at the end of the integration period after

which the integrator is reset and the cycle repeats. Figure 3.4

shows a similar circuit which was constructed to use two integrators

since this relaxes the requirement for a short reset period and

fast slew rate amplifiers for the integrators. The cut-off

sequency of the filter is 2048 C/S which corresponds to a sub-

cycle of 244 micro-seconds. The integration time constants of the

integrators were chosen to be greater than the sub-cycle period

with variable gain introduced in the summing amplifier to set the

gain of the filter to unity. Output voltage offsets due to input

offset currents were adjusted with the offset voltage controls.

A standard configuration was chosen for the sample and hold circuit,

the hold capacitor was chosen to be as large as possible to minimise

the sample to hold error which is a function of the interelectrode

capacitance of the switch, the pinch off voltage and the peak to

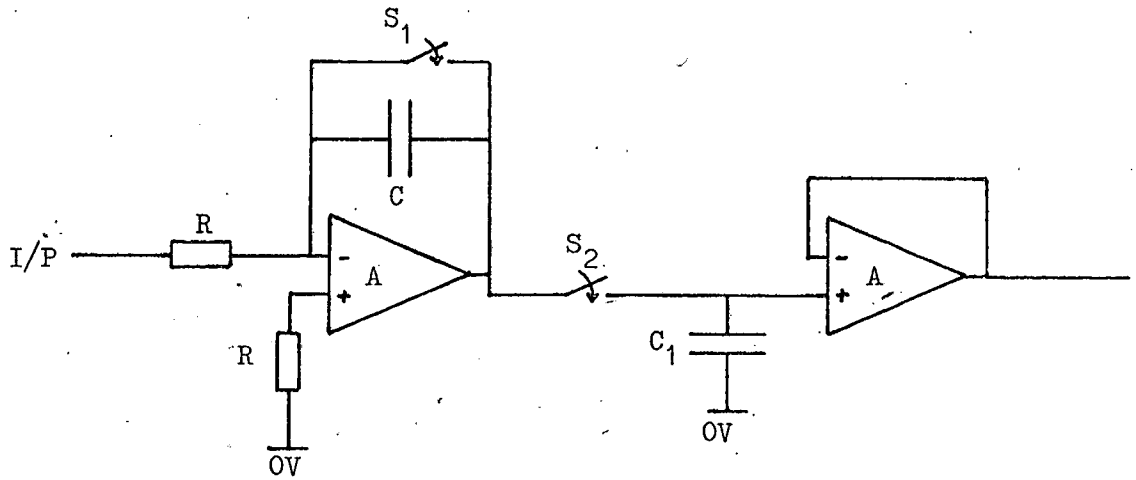peak voltage of the waveform driving the gate of the switch. This
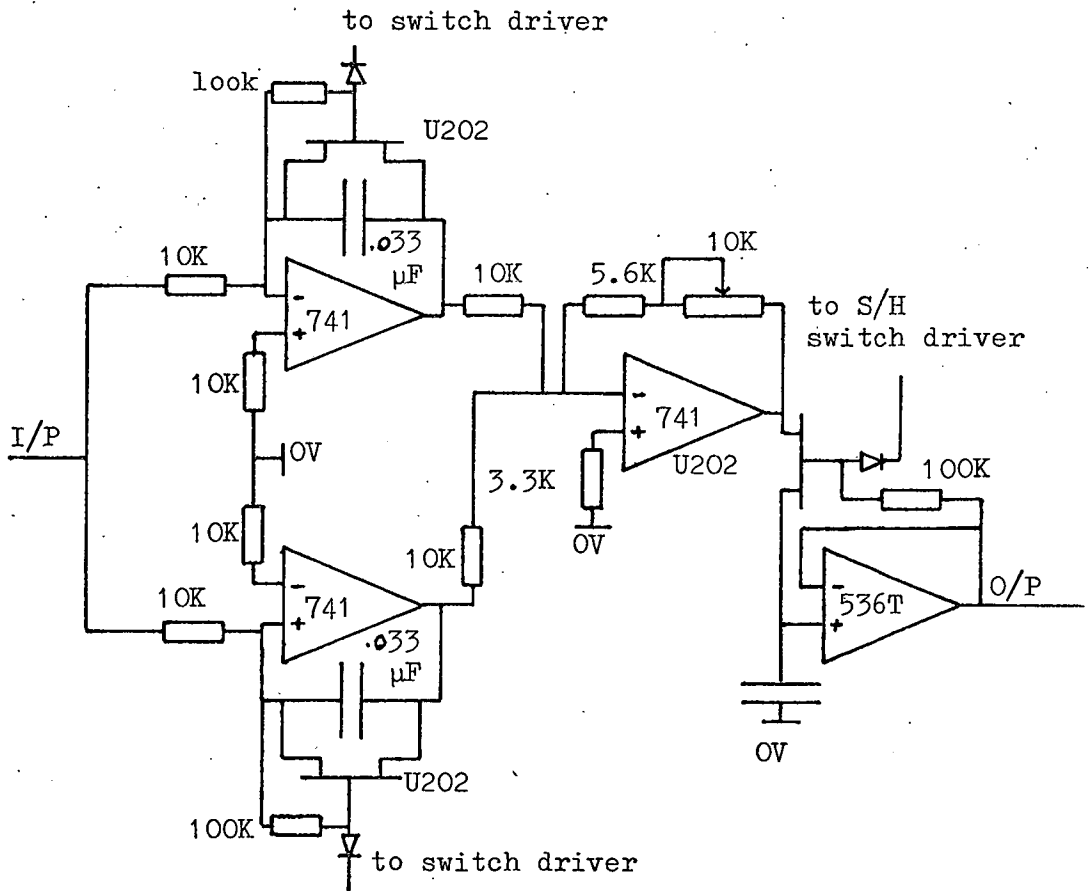
Figure 3-3

Low pass sequency filter

Figure 3-4 Practical Low pass sequency filter

error is given by the expression

$$\text{sample/hold error } e = \frac{(Vd - Vp).Cqd}{Cqd + C_H} \; ;$$

Cqd = gate to drain capacitance

C_H = Hold capacitor

Vp = pinch off voltage

Vd = peak to peak voltage of drive signal

which for the components and drive voltage used will be equal
to 31 milivolts.   This error was also adjusted for by the offset
voltage controls on the operational amplifiers since it acts as
an offset voltage and is independent of the voltage on the hold
capacitor.

3.4 Analog-Digital Converter (A.D.C.)

The A.D.C. used was a bipolar 12 bits successive approximation
type (Analog Devices A.D.C. 12QM)[24] with a total conversion time of
twenty five micro-seconds including a settling time of two and
half micro-seconds.   A block diagram of the A.D.C. is shown in
Figure 3.5.   The A.D.C. is reset on the positive going edge of the
convert command pulse and conversion begins on the negative edge.
A logical signal (STATUS) is generated internally to indicate the
state of the converter (logic 1 when the A.D.C. is in the conversion
mode and logic 0 when in the rest mode).   The STATUS signal can be
used to synchronise other devices to the converter.   The input
voltage (bipolar-unipolar) and the output code type (offset binary
or two's complement representation) are selectable by connections
made between specified pins on the converter.   Full scale voltage
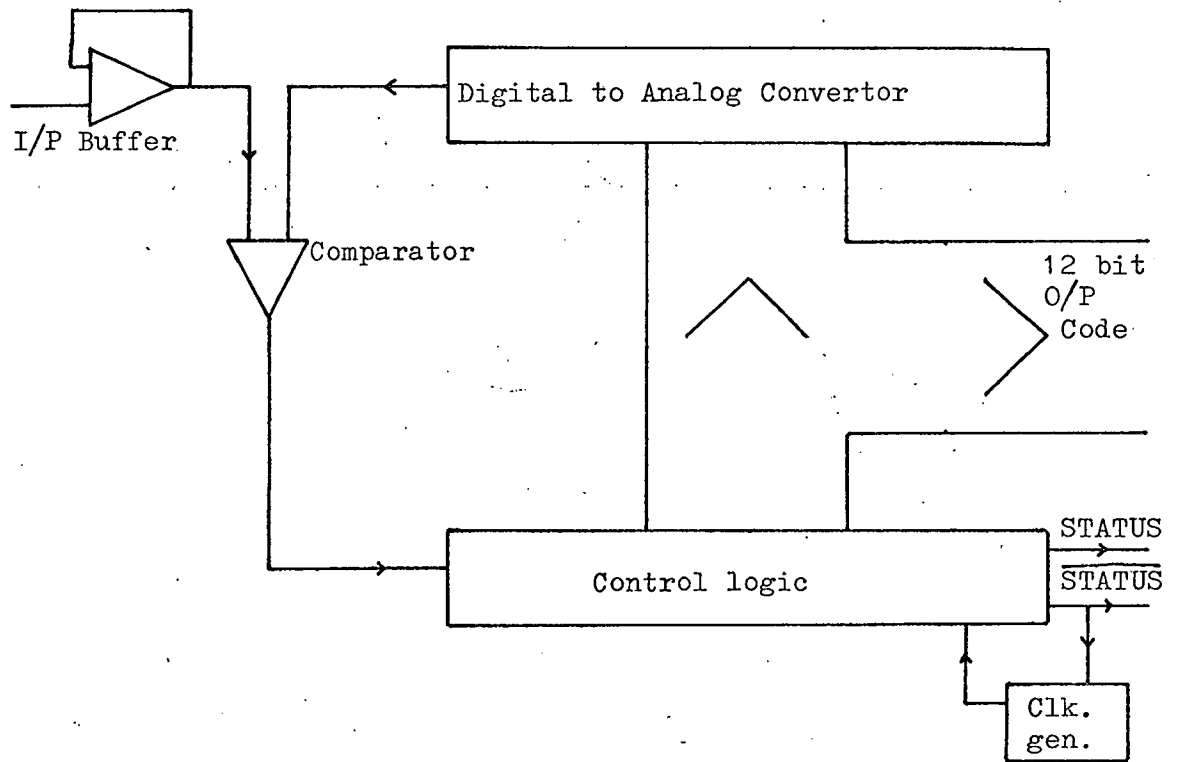input in the unipolar mode is 10 volts and $\pm$ 5 volts in the bipolar mode.

Figure 3-5

Analog to Digital converter block diagram

The A.D.C. was set up for bipolar input two's complement output operation. During the data transfer between the processor and the computer the A.D.C. is forced into the reset state by holding the convert command input in the logical one state, by this means an all zero word is presented to one set of inputs of the arithmetic unit.

3.5 Arithmetic Unit

Due to the normalisation requirements of the Walsh coefficients and the data word length presented by the A.D.C., the arithmetic unit word length was chosen to be twenty-four bits. A block diagram of the arithmetic unit is shown in Figure 3.6. Six four bit full adders (signetics 8260)[(25)] connected as a ripple carry adder formed the major part of this unit. The worst case settling time of the adder based on the propagation delays of the carry input and the data inputs to the carry outputs is 133 nano seconds for the twenty-four bits.

The gated output of the Walsh function generator is Exclusive-ORed with each bit of the A.D.C. data word and connected to the least significant twelve bits of the adder inputs, while the twelve most significant inputs are connected to the Exclusive-OR of the A.D.C. sign bit and the gated Walsh function generator output, the latter being connected to the carry input of the adder thus forming a twenty-four bit two's complement representation of the data word.

When the control inputs of the Full adder $(C_{in}h, E_{in}h)$ are held in the logical one state each output of the adder will be the
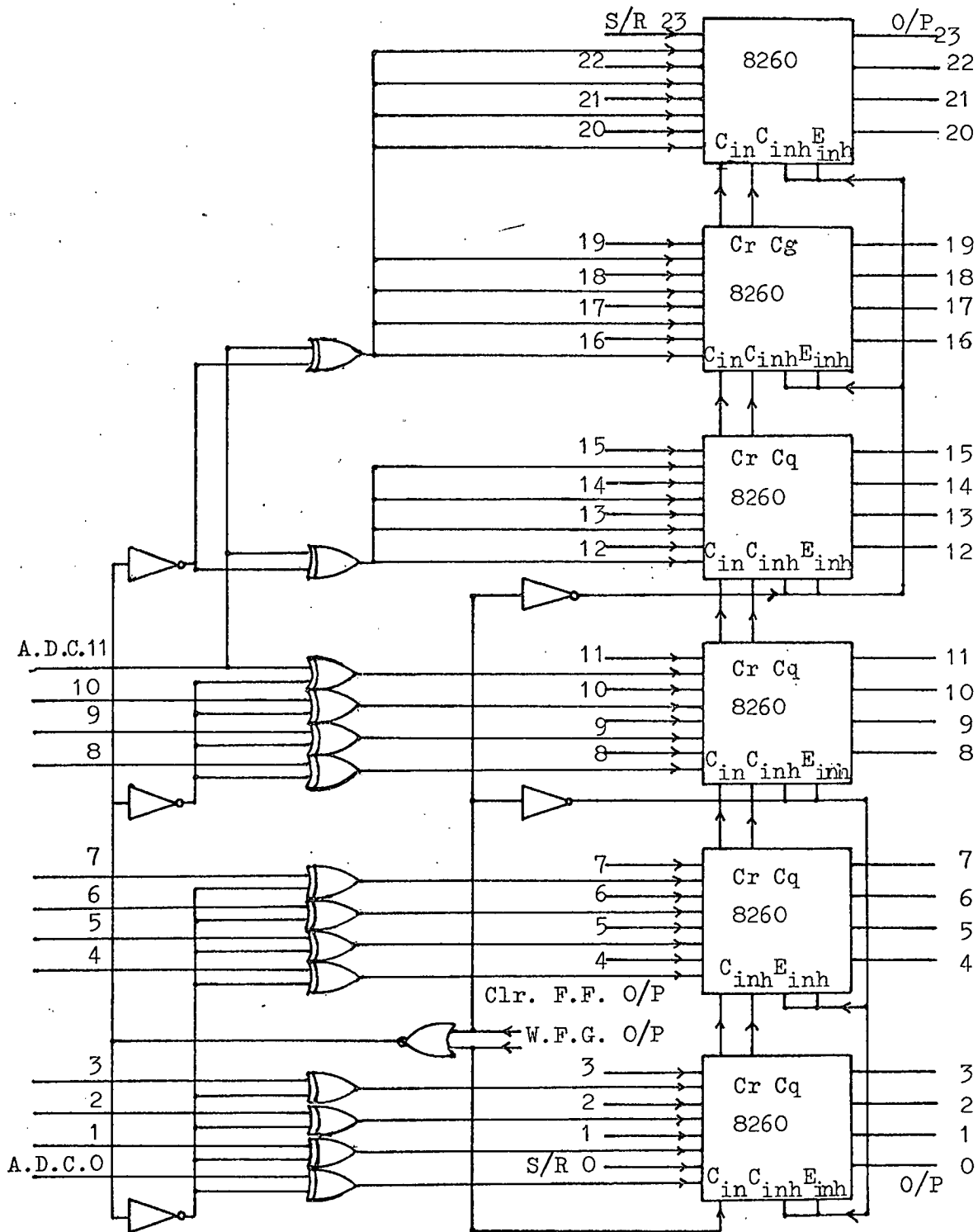
Figure 3-6

Arithmetic Unit

AND function of the corresponding pair of inputs.   This feature of the adder is used during the processor data transfer operation to form the AND function of the A.D.C. data word (in the logical "0" state) and the data words from the shift register stack so that the latter will be cleared at the end of the transfer operation.

3.6 Shift Register Stack

The number of coefficients to be computed was chosen to cover the highest harmonic generated by a sinosoidal frequency of 16 cycles which occurs at a sequency of 48 cycles per second corresponding to a Walsh function having 95 or 96 zero crossing per second.   Consequently the shift register stack was built from 12 "Signetics 2500"[26] dual hundred bit shift registers to make up a 24 bit x 100 word serial store.   A block diagram of this part of the system is shown in Figure 3.7.

The inputs of the stack were driven by open collector Hex buffers (SN7404) with pull-up resistors to minimise the logical "0" state voltage level input, and the outputs were buffered by hex inverters due to the limited drive capability of the shift registers.   This arrangement increases the clock to data output delay and raises it from 250 nsec. to a minimum of 280 nsec.   The recirculate and output enable controls of the shift registers were inhibited to allow the data to circulate through the arithmetic unit.   The twelve most significant bits of the S/R stack output are connected through voltage level translators to the computer data word inputs thus performing a division by 4096.
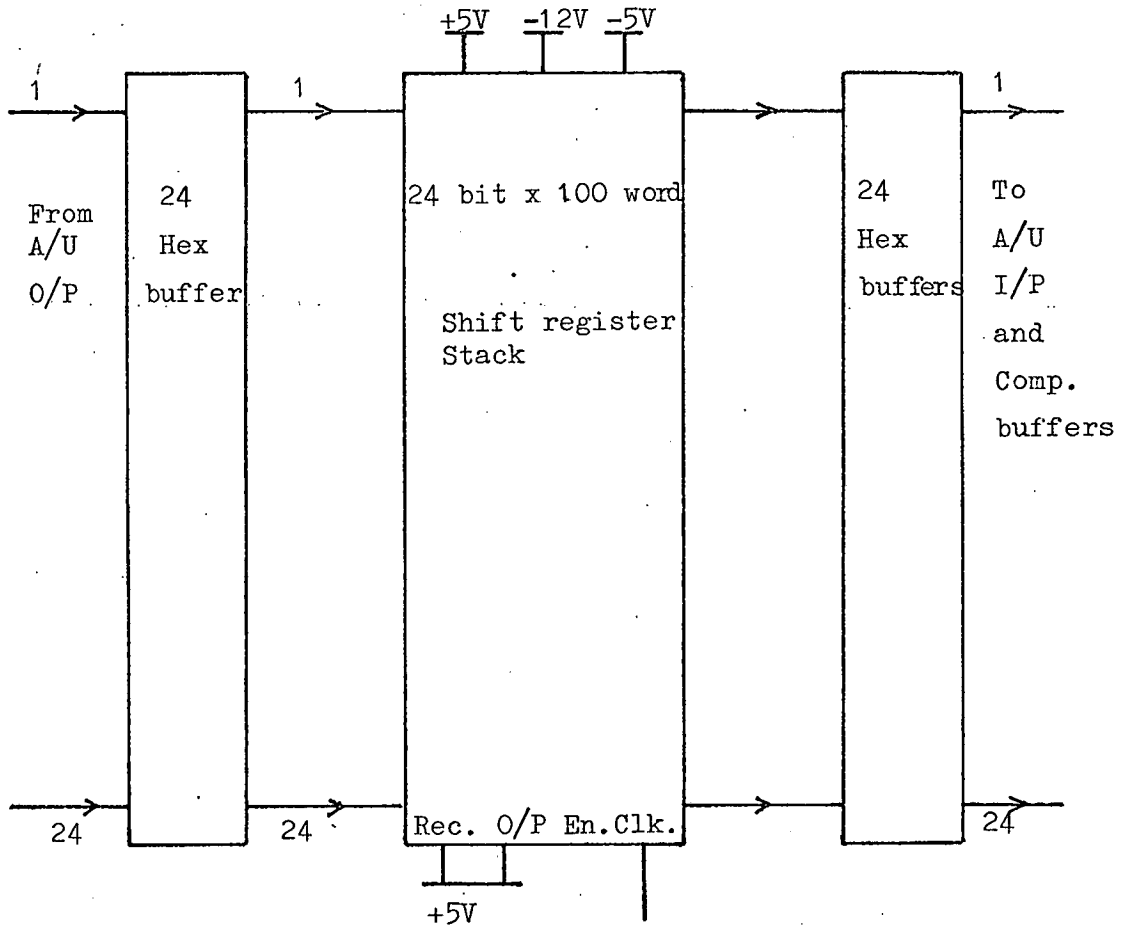
Figure 3-7

Shift register stack

3.7 Walsh Function Generator (W.F.G.)

The requirements for a Walsh Function Generator that can be scanned in sequence for every time increment is met in a generator proposed by Stafford and Durrani.[22] This generator is the hardware realisation of the equation

$$Wal(K,t) = \exp . j\pi[K_{n-1}t_o \oplus (\overset{n-1}{\underset{r=1}{\bigoplus}} (K_{n-r} \oplus K_{n-r-1})t_r)] \qquad (2)$$

this equation generates Walsh functions in increasing number of zero crossings in the interval of definition (sequence order) as "K" takes the values $0,1,2...2^n-1$. For the case of n=2 substituted in equation 2

$$Wal(K,t) = \exp . j\pi(K_1 t_o \oplus (K_1 \oplus K_o)t_1) \qquad (3)$$

this will generate the Walsh function's for the indicated values of K as shown below

$$
\begin{array}{c|cccc}
 t & 0 & 1 & 2 & 3 \\
K & & & & \\
\hline
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 \\
2 & 0 & 1 & 1 & 0 \\
3 & 0 & 1 & 0 & 1 \\
\end{array}
$$

where 1 is equivalent to (-1) and 0 to +1. Figure 3.8 shows a hardware realisation of equation 3. If the "K" register is initially reset and for every C_1 pulse the register is clocked four times then the output of the generator will scan all the Walsh functions in the set defined by equation 3 as outlined in the timing diagram.
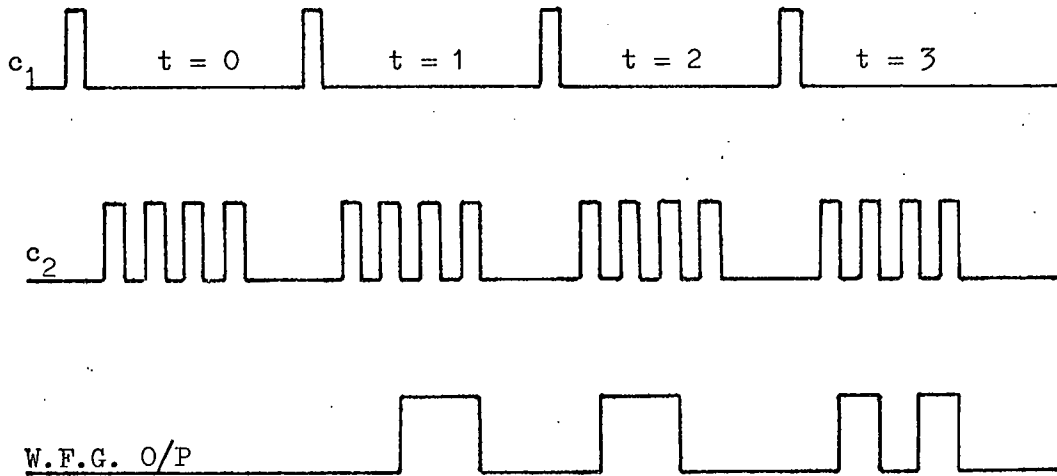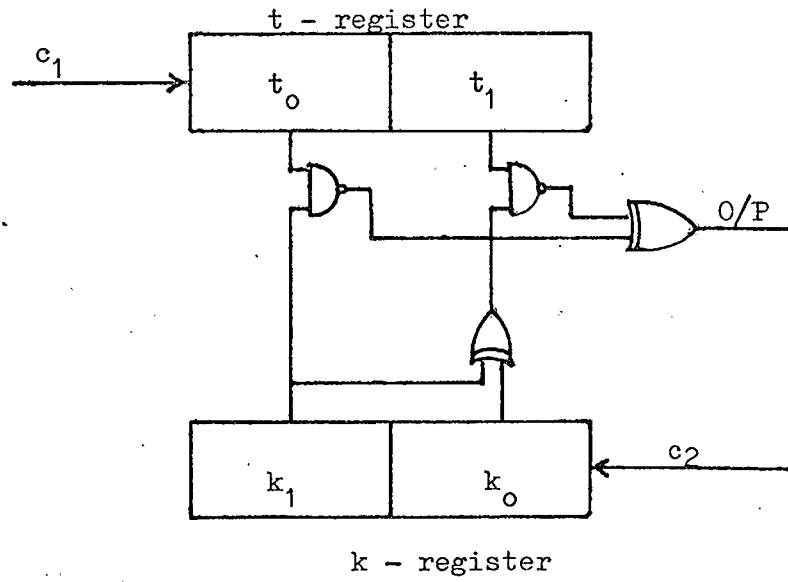
Figure 3-8

Walsh function generator based on

equation 3
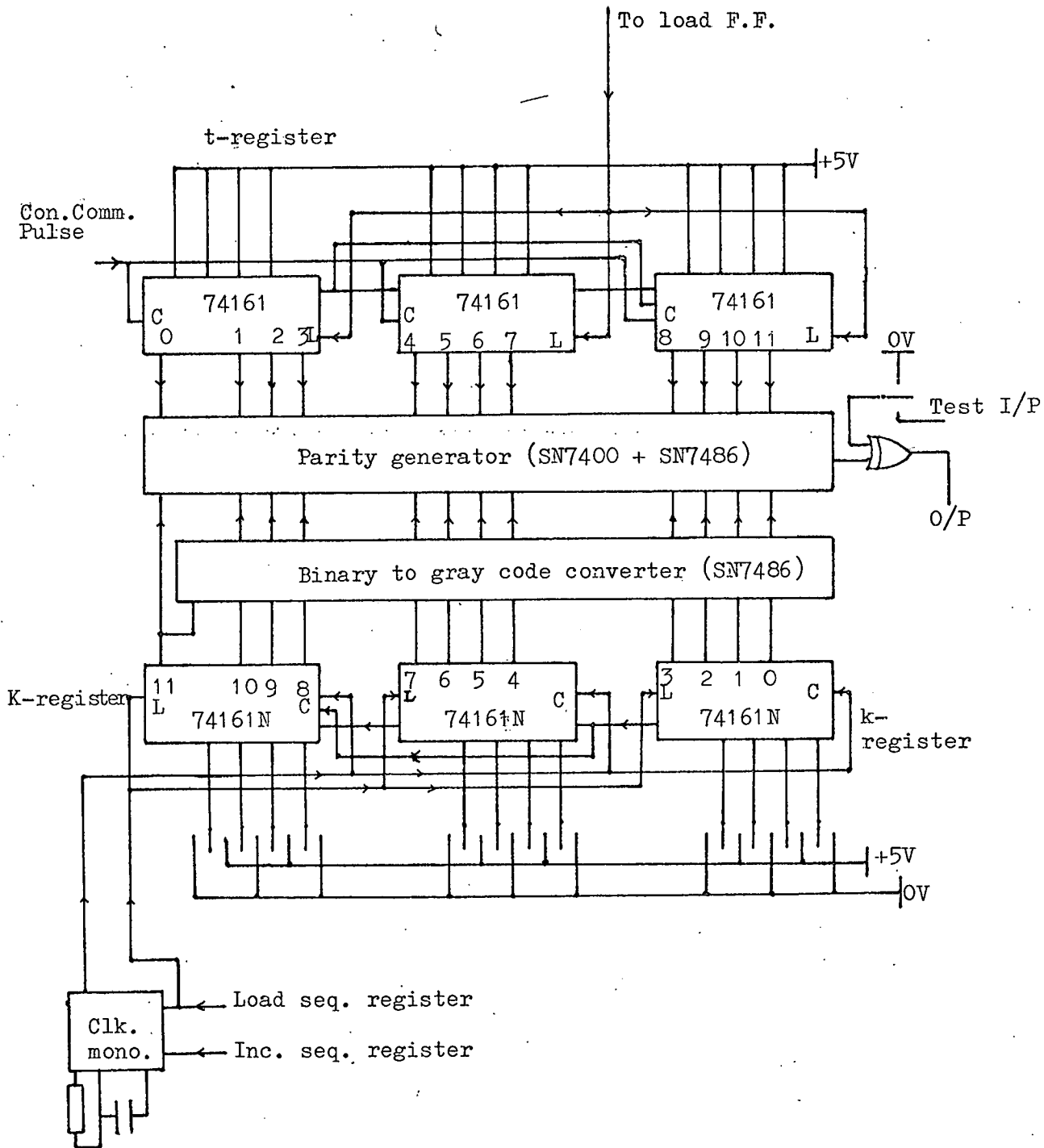
Figure 3-9

Walsh function generator

A twelve bit Walsh function generator based on equation 2 was built, Figure 3.9 shows a block diagram of this generator. The "K" register was constructed from synchronous presetable binary counters (SN74161) with switches on the data inputs so that the generator can start the sequency scan from a preselected sequency after which the register is reloaded with the initial number through the clock mono-stable and the reload control of the register. Due to the presence of a dummy time pulse when the processor is first started, a flip-flop "L" was used to load the time register with all ones as well as to inhibit the all ones decoder. The worst case settling time for the generator output from the application of a clock pulse to the sequency register is approximately 200 nsec.

3.8 Processor Control Logic

All the control levels and timing pulses in the processor are synchronised by a master clock which defines the integration times of the low-pass filter integrators thereby defining the basic processor computation sub-cycle.   The processor control logic can be divided into three main control blocks:-

(1) master clock and low-pass filter control

(2) clear and convert command pulse control

(3) shift register stack clock control

(1) Master Clock and Low-pass Filter Control

The master clock illustrated by Figure 3.10 and 3.11 is a gated oscillator constructed from two cross coupled mono-stables (SN74121), the clock frequency (8192 C/S) was set to be four times the cut-off sequency of the low-pass filter (2048 C/S).   Prior to

Figure 3-10

Master oscillator and L.P.S.F. Control
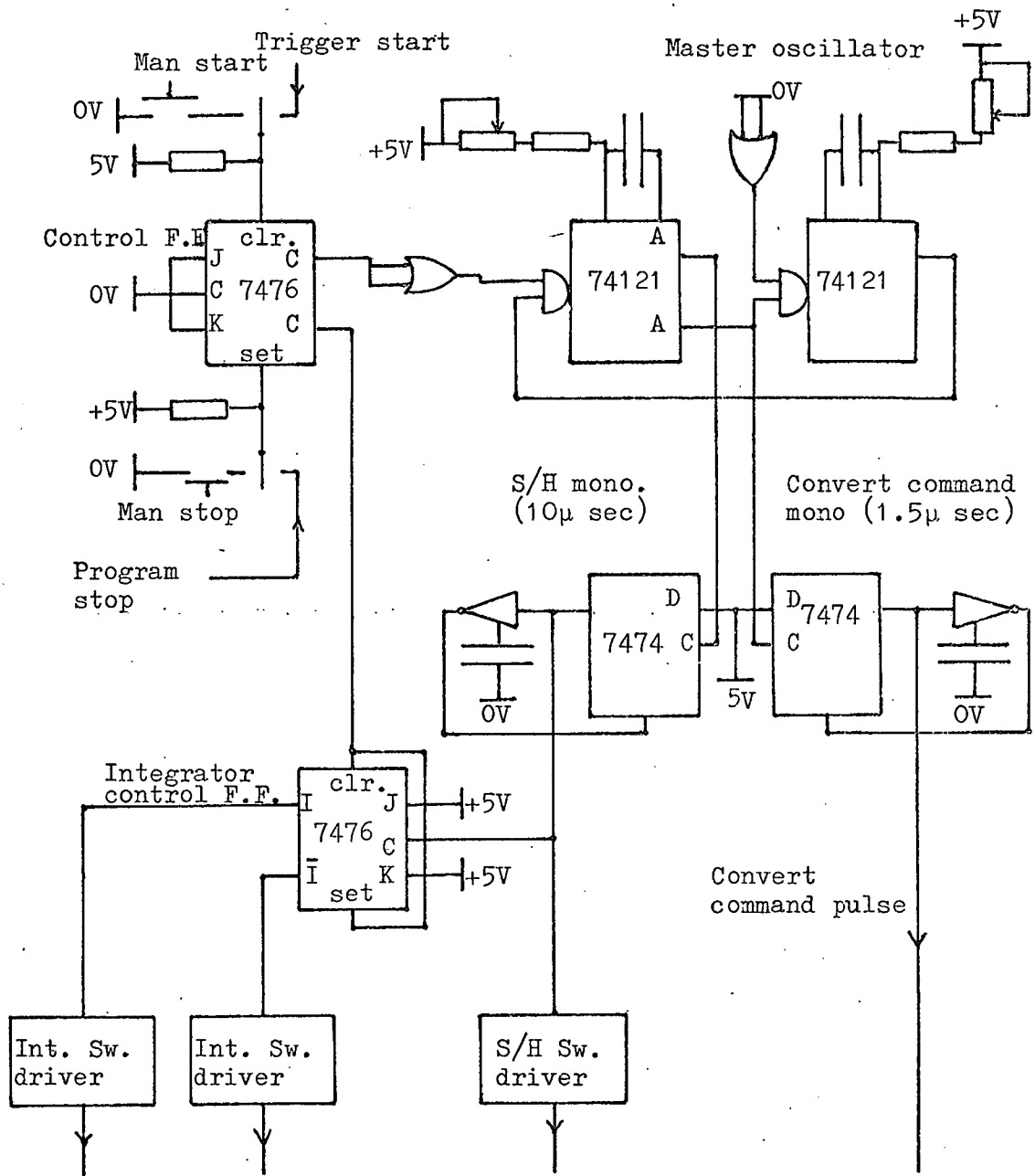
control F.F.C

Master osc.

A

|<122μsec>|<122μsec>|

A

sampling mono.

|>|< 10μsec

Integ. cont.

I or Ī

|< 244μsec >|

I or Ī
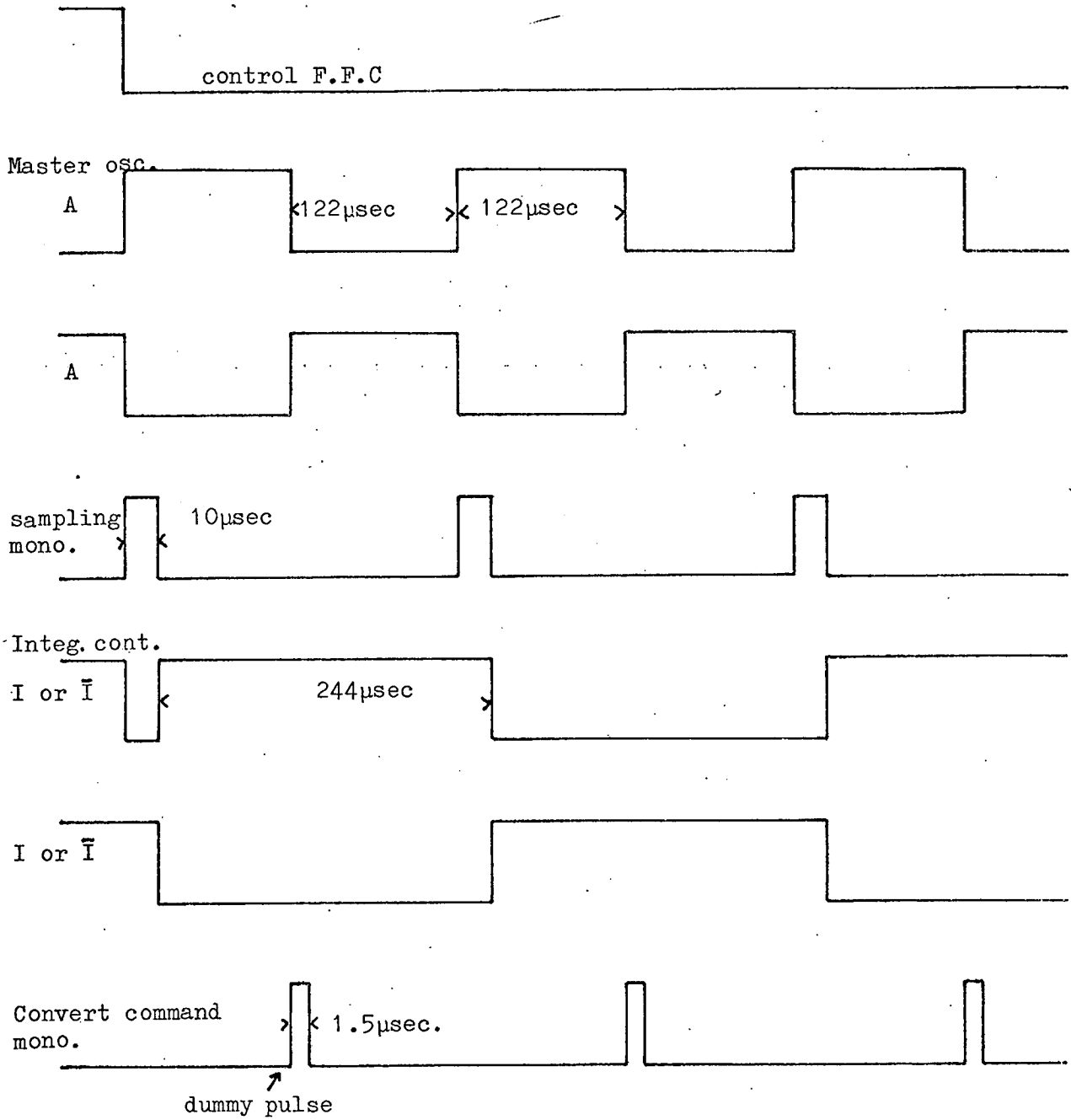
Convert command mono.

|>|< 1.5μsec.

dummy pulse

Figure 3-11

Timing diagram for Figure 3-10

starting the master clock the integrators are held reset by forcing the outputs of the integrator control flip-flop to the logical "1" state. The sampling mono-stable triggers off the positive edge (output A) of the master clock, clocks the integrator control flip-flop and therefore allows the sample and hold unit to be updated prior to the reset of either integrator. The convert command mono-stable triggers off the positive edge (output A) of the master clock. Due to this arrangement a dummy pulse is generated which is used in loading the Walsh function generator time register with all ones so that the time register is clocked to the zero state by the next pulse.

(2) <u>Clear and Convert Command Pulse Control</u>

When the processor is in the stop mode and the manual reset button is depressed the load, clear and break request B, flip-flops will be set. In this state the analog to digital converter will be held in the reset state and the arithmetic unit is modified to perform the AND function of the data word and the shift register stack output. The processor will run through one dummy sub-cycle thus clearing the shift register stack. If the clear button is held depressed the same previous conditions will hold until the button is released.

In the last computation sub-cycle the all ones condition of the Walsh function generator time register is decoded and conditions the K input of the clear flip-flop as shown in Figure 3.12. At the end of the computation the 100 counts overflow pulse (generated in the S/R stack clock circuit) will set the break request flip-flop B, and clock the clear flip-flop to logic state 0. The clear flip-flop
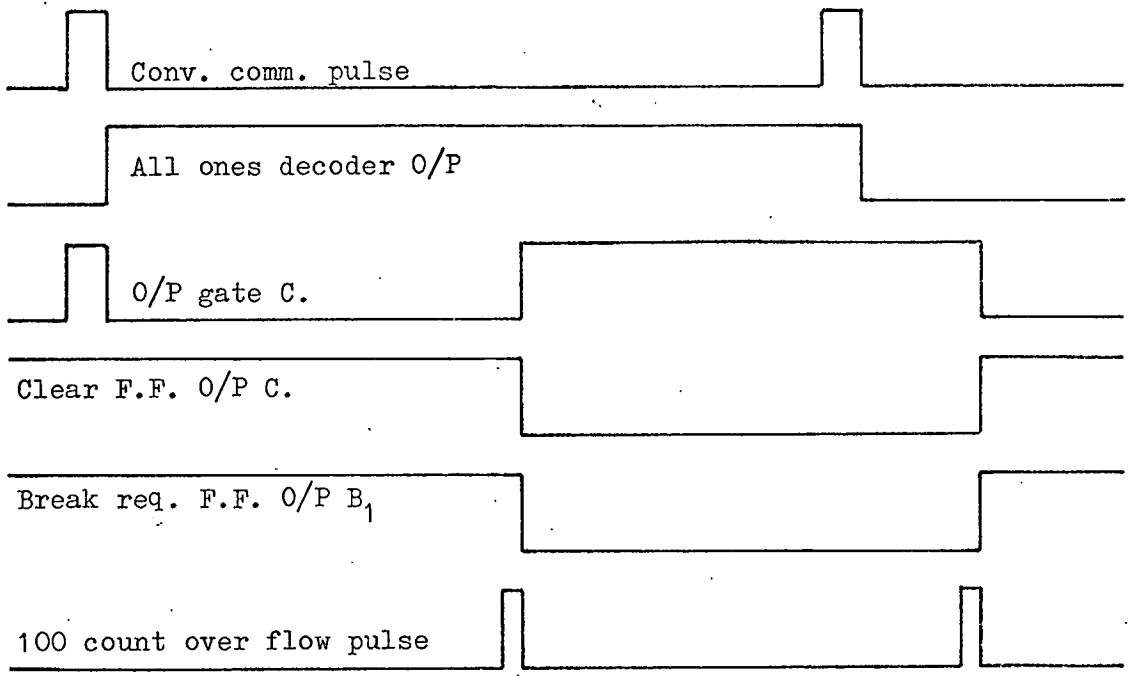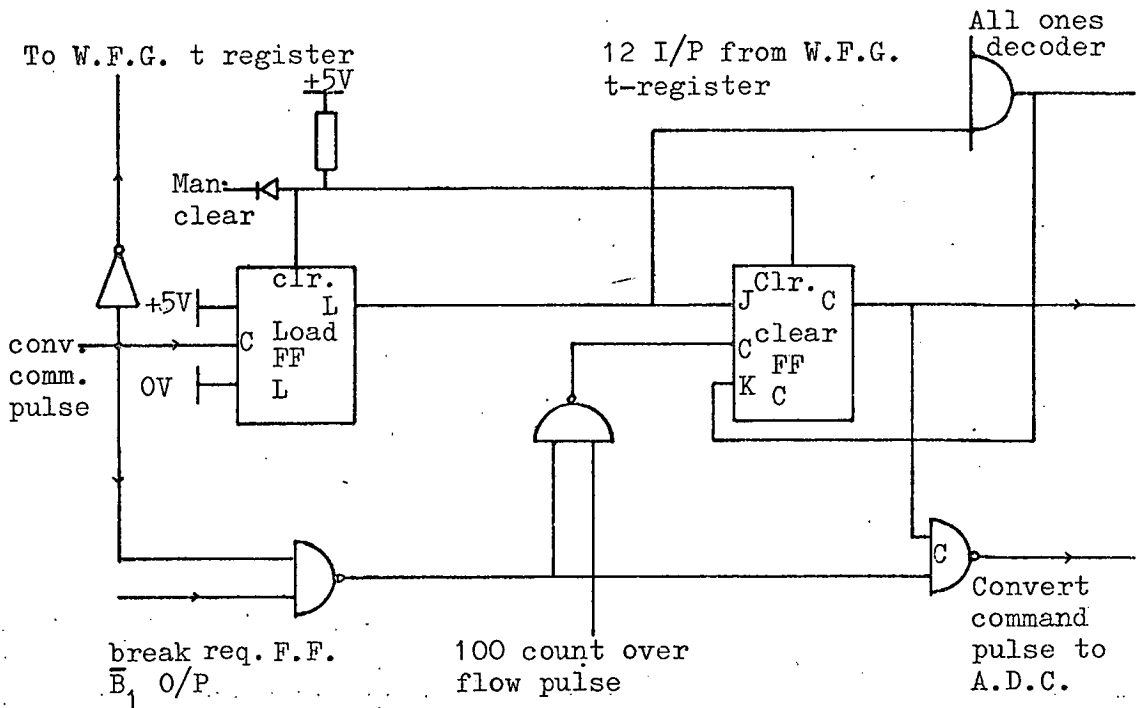
Figure 3-12

Convert command pulse and clear control
circuit

output will hold the A.D.C. in the reset state by forcing gate C output to logic state 1 and at the same time modifies the arithmetic unit to simulate a set of AND gates. The break request B_1 signal inhibits the next convert command pulse from falsely clocking the clear flip-flop to the one state thus allowing the computer data transfer operation to be completed. The convert command pulse will clock the Walsh function generator time register to the "0" state thus lifting the all ones condition from the K input of the clear flip-flop. At the end of the computer data transfer operation the hundred counts overflow pulse will clock the clear flip-flop to the "1" state to initiate the next A.D.C. conversion cycle.

(3) Shift Register Stack and Sequency Register Clock Control (Figures 3-13)

At the termination of the A.D.C. conversion cycle the status output clocks flip-flop S.R.C. to the 1 state thereby removing the stop condition from the clock generator. Clock pulses are routed to the shift register stack, the 100 counts register and the sequency register of the Walsh function generator thus synchronising the W.F.G. output and the shift register stack.

The clock pulse width of the oscillator was set to 200 nano seconds which is the minimum for reliable clocking of the series 2500 shift registers. The time between clock pulses was set according to the worst case delays of the arithmetic unit, the shift register stack and the Walsh function generator output plus a safety margin, this comes to a total of 650 nano seconds and results in a total computation time of 85 micro seconds for the 100 partial sums.
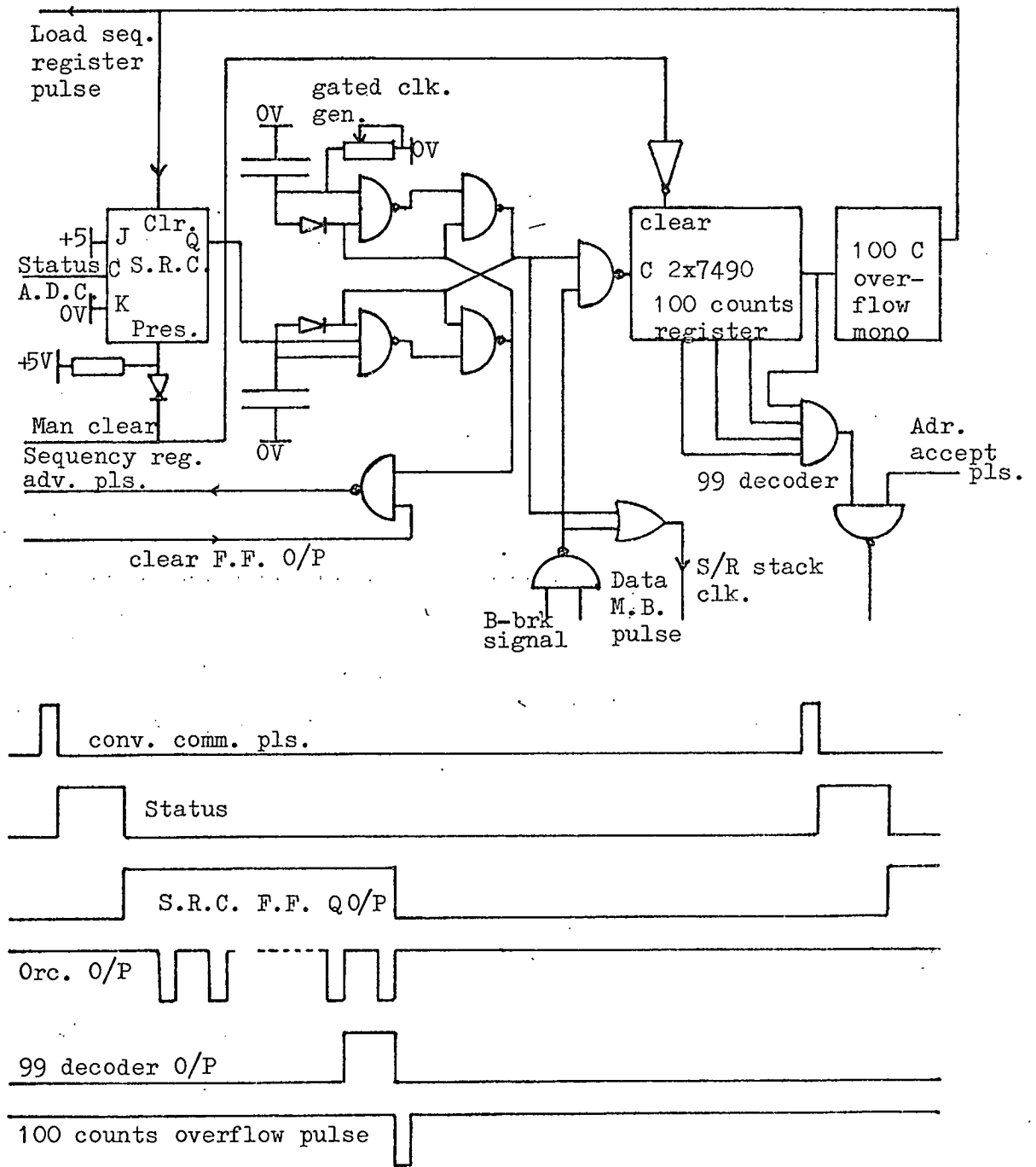
Figure 3-13

Shift register stack and sequency register
clock control

In the clear mode of the processor clock pulses are routed to the shift register stack and the hundred counts reigster only. At the end of the clear operation the S.R.C. flip-flop will be reset by the 100 counts overflow pulse. In the computer data transfer mode the computer B-break signal gates the (Data Memory buffer) pulses to the hundred counts register, in this case the 99th state of the 100 counts register is decoded and used to gate the hundredth computer address accept pulse for clearing the break request condition.

3.9 Computer Interface Control Circuit

Due to the timing requirements of the analyser and the amount of data to be transferred to the computer, the single cycle data break facility was chosen since it needs the least amount of time for the transfer operation (1.5 micro second per word transfer). To initiate a data break transfer of information, the interface control must provide the following signals.[16]

(1) the absolute address in core memory

(2) the data word

(3) direction of transfer indication

(4) data break request signal

(5) single cycle request signal

Signals in (3) and (5) above are D.C. levels which can be fixed in this instance since information is transferred into the computer. Referring to the computer interface control circuit shown in Figure 3-14 and the timing diagram shown in Figure 3-15 it will be seen that at the beginning of the last sub-cycle of a computation cycle the all ones decoder output conditions the J input of the break request flip-flop

B_1 and at the end of the computation this flip-flop is clocked to the one state by the 100 counts overflow pulse signalling the computer to enter the break state after the completion of the current instruction.

In the break state the computer generates command pulses and control levels. Three are used in the interface circuitry; these are the address accept pulse, the B-Break signal and the Data to Memory Buffer pulse. The address accept pulse is used in incrementing the interface address register while the B-Break signal is used to gate the data to M.B. pulse (these pulses are continuously generated when the computer is in the run mode) for shifting the data out of the shift register stack and incrementing the 100 counts register. The Data to M.B. pulse is generated after the address accept pulse, therefore the 99th state of the 100 counts register is decoded and is used to gate the 100th address accept pulse to reset flip-flop B_2 which signals the end of the break request condition.

At the end of the data transfer the 100 counts overflow pulse is gated by B_1 flip-flop to reload the interface address register with the starting address and to trigger the interrupt request mono stable (for the computer program recognition of a data transfer completed condition). The ungated 100 counts overflow pulse clocks B_1 flip-flop to the logical "0" state and in the next sub-cycle clocks the B_2 flip-flop to the logical 1 state thus setting the conditions for the next data transfer operation. The data from the S/R stack and the absolute address from the interface address register are inverted and level translated to present the correct logic levels to the computer (logical $1 \equiv 0V$; logical $0 \equiv -3$).

Break request F.F.S.

99 decoder

Man. clear

+5V

All ones
decoder

clr.

J B$_1$

C B$_1$

K B̄$_1$

100 C
overflow
pl.

J Clr B$_2$

C

OV K B̄$_2$

Address accept
pulse

Brk. request
signal

OV

+5V

8281

Load C

7493

clr. C

7493

Clr. C

Man. clear

B$_1$

100 C over
flow pulse

Int. I
request
mono.

Address accept
pls.

Interupt request
pls.

+5V
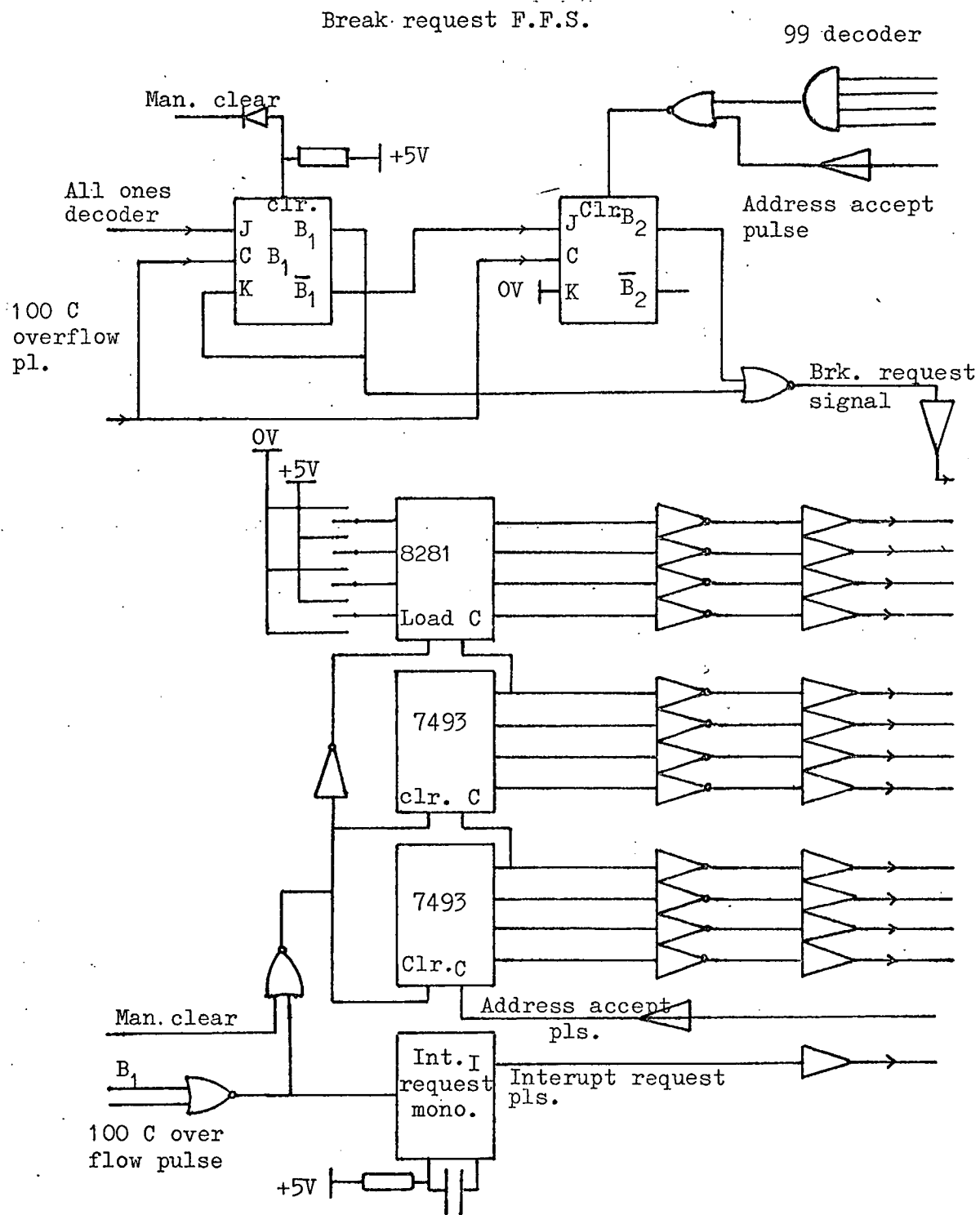
Figure 3-14

Computer interface circuit
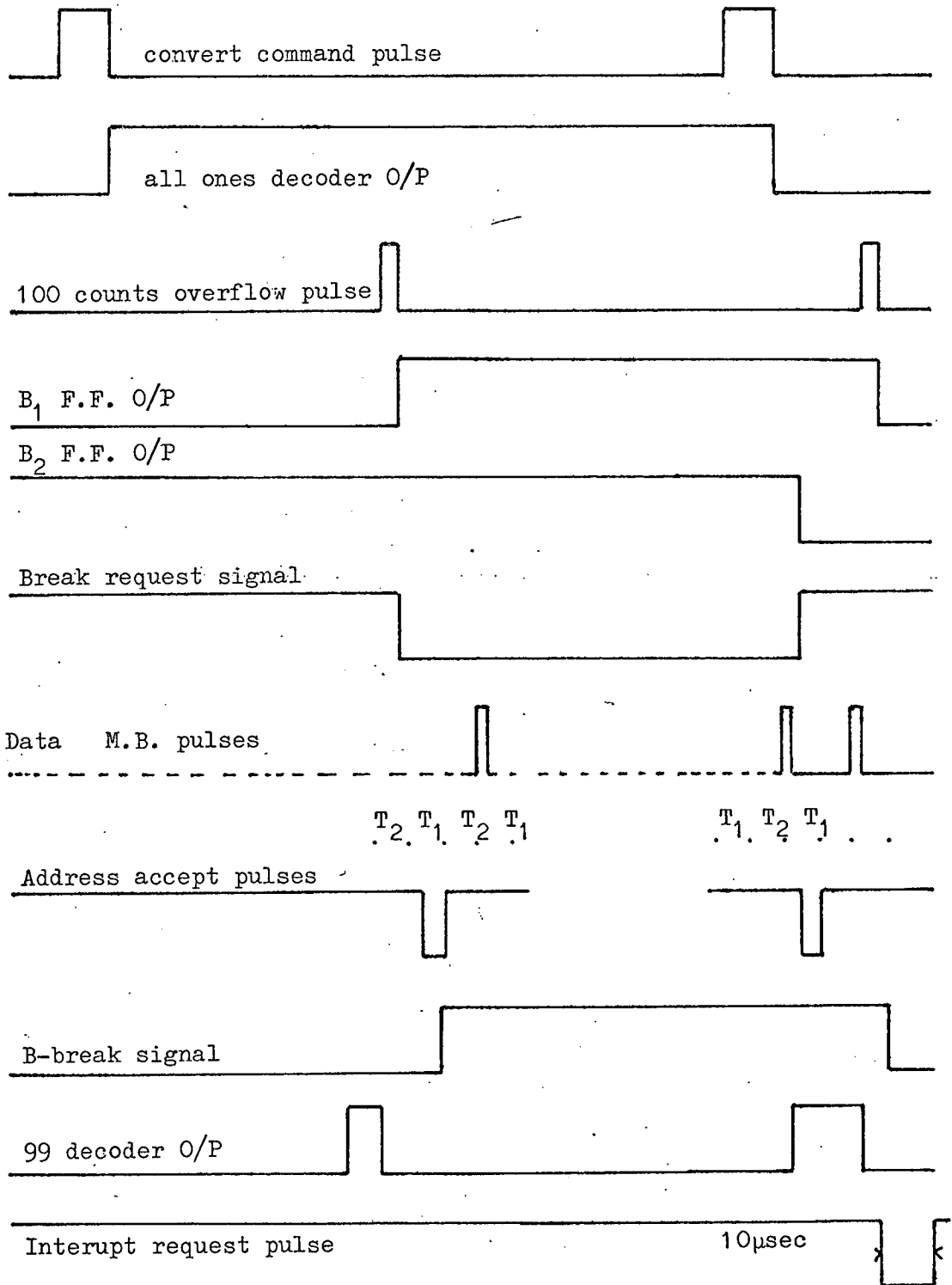
Figure 3-15

Timing diagram for Figure 3-14

3.10 Detailed Block Diagram

Figure 3-16 shows a detailed block diagram of the processor with the data flow and control signals included. Figure 3-17 shows the major control signals of the last computation sub-cycle and the first computation sub-cycle of the next period. All the circuits apart from the computer level translators were built on single sided vero-boards. The computer level translators were built from modified D.E.C. inverters and connected to the computer through single screened leads to minimise noise pickup and cross talk. The processor was tested by modifying it to run through one sub-cycle with a D.C. voltage connected to the A.D.C. and then checking the contents of the S/R stack by manually shifting the data out. The same procedure was used to check the processor-computer interface to make sure that no loss of data was incurred by the transfer operation.
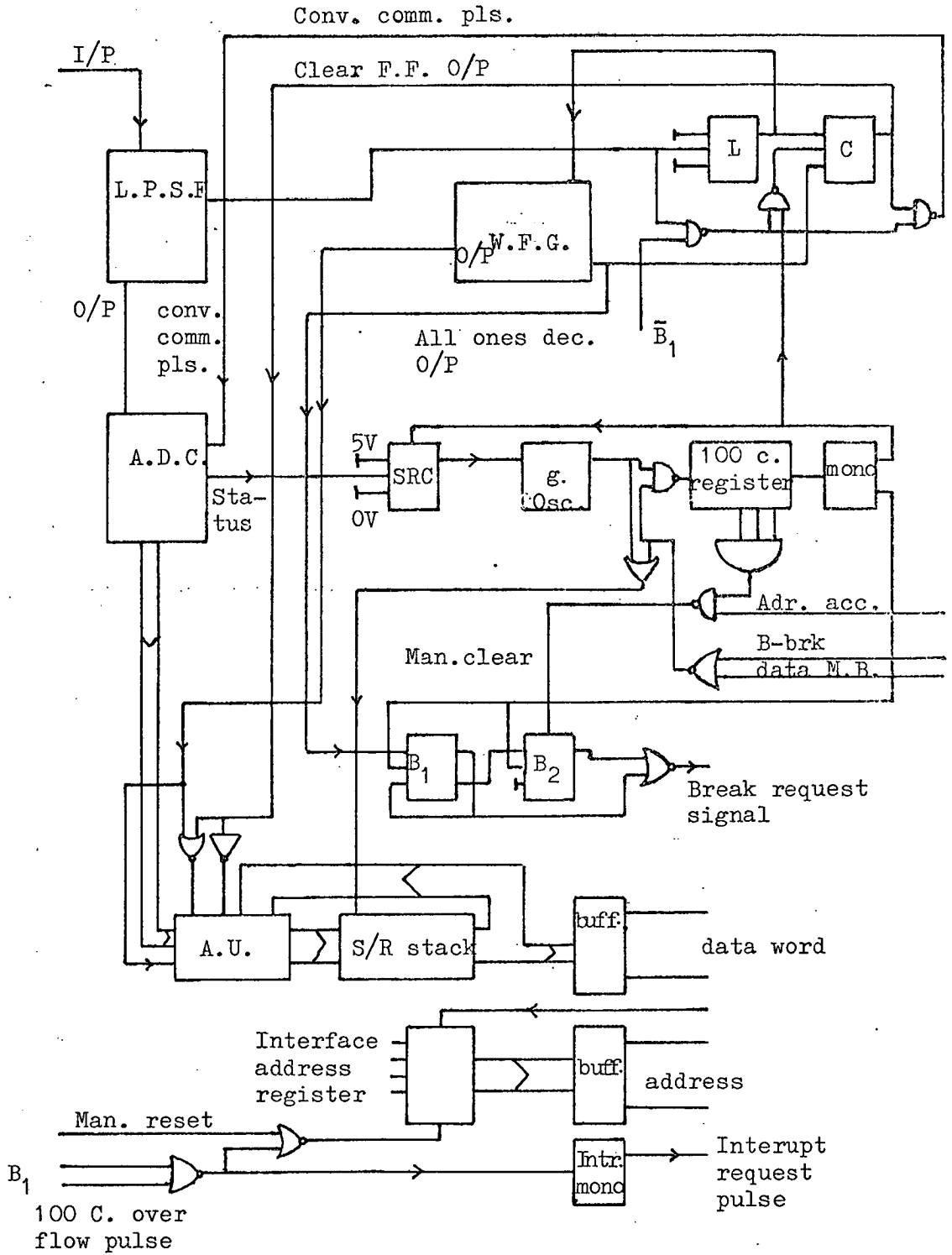
Figure 3-16

Detailed block diagram
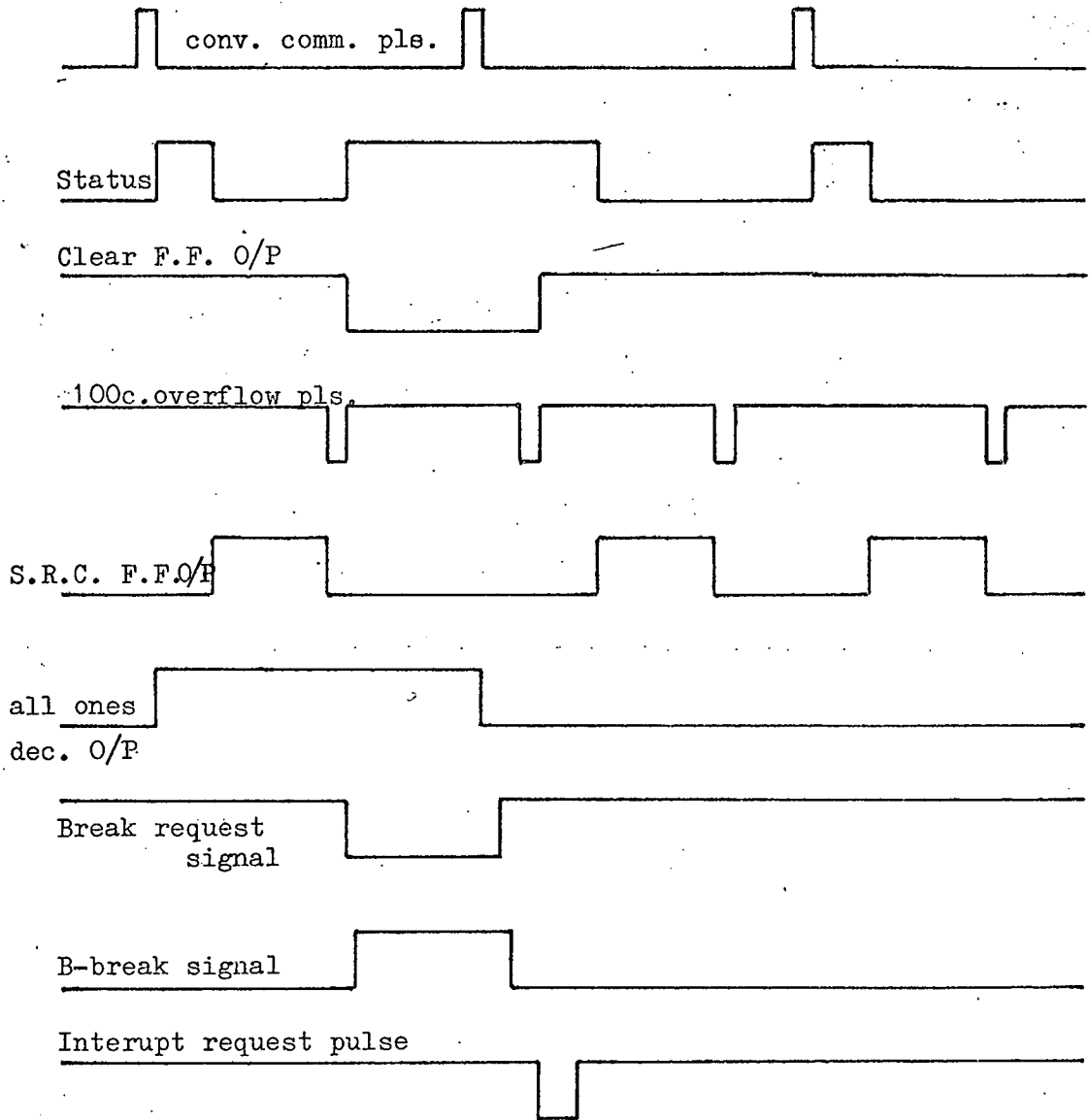
Figure 3-17

Major signals of Figure 3-16

CHAPTER 4

PERFORMANCE OF EXPERIMENTAL SYSTEM

4.1 Introduction

The performance of the analyser system was tested by using two known signals. The first of these signals were the set of Walsh functions, to generate these functions a generator was built which is synchronised to the analyser system. The second known signals used were sinusoidal functions of various frequencies. A trigger generator was built which starts the analyser system at various phase angles relative to the sinusoids. A method of predicting the coefficients of sinusoides in the Walsh transform domain was developed and is outlined in Appendix B. In Appendix C is a listing of the coefficients of various sinusoids. in the range 1 to 16 cycles and 1 volt amplitude.

4.2 Trigger Circuit and Test Generator

The test generator shown in Figure 4-1 was built of three four-bit binary counts (SN7493) and is clocked by the convert command pulse. The reset of the generator is connected to the manual reset of the analyser system, by this means the outputs of the generator are synchronised to the Walsh function generator. The highest sequency Walsh function generated (sequency 2048) corresponds to the lowest bit of the generator and the lowest sequency Walsh function (sequency 1) corresponds to the highest bit. A total of twelve Walsh functions are obtained from the generator without Exclusive-OR gates.
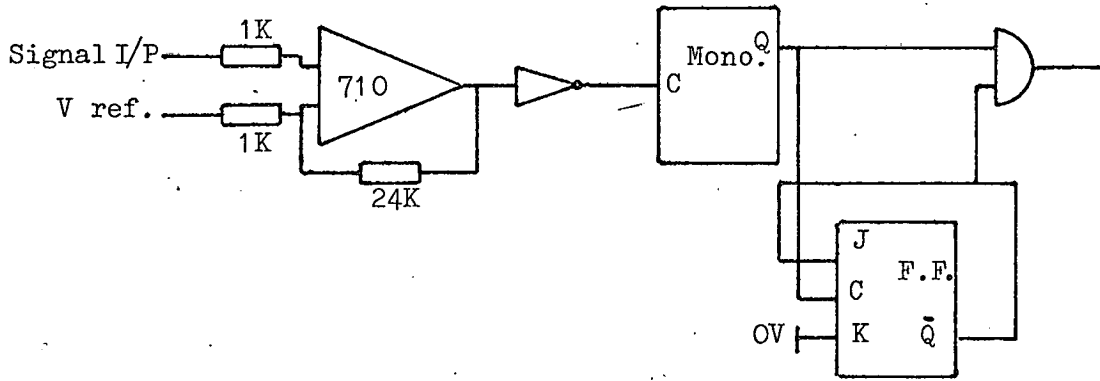
Figure 4-1

Trigger Circuit
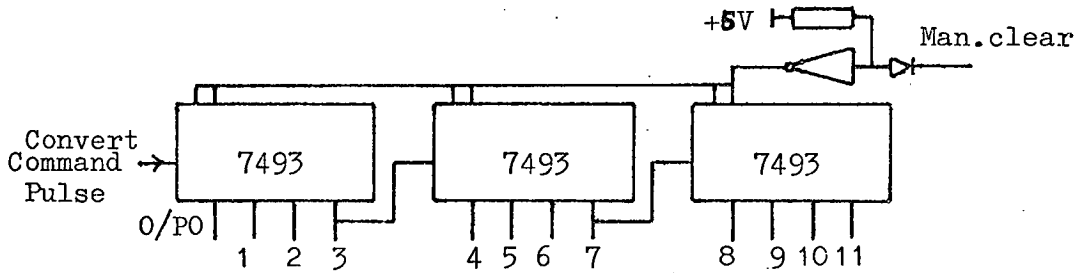
Figure 4-2

Square wave generator

Trigger pulses at various points on a sinewave were generated by means of the circuit shown in Figure 4-2. This circuit consists of an analogue comparator (μA710) and a gating system made up of a mono-stable and a bistable arranged so that when in the reset condition it allows only one pulse to trigger the analyser system. Various starting·points on the sinusoide were obtained by varying a D.C. reference input to the comparator. To aleviate the problem of false triggering (due to the high gain of the comparator), the comparator was modified to act as a Schmit trigger with a hysteresis of .194 volts.

4.3 Response of the System to Various Test Generator Inputs

According to the multiplication equation of Walsh functions

$$\sum_{t=0}^{2^n-1} A\,\text{Wal}(k,t).\text{Wal}(m,t) = \sum_{t=0}^{2^n-1} A\,\text{Wal}(k\oplus m,t) = \begin{cases} 0, m \neq k \\ 2^n A, m=k \end{cases} \tag{1}$$

it can easily be arranged for the analyser system to be tested by feeding a D.C. voltage equivalent to a binary number A to the lowpass sequency filter and connecting any output of the test generator to the multiplying input of the Walsh function generator. An inherent division by 2^{12} in the analyser will eliminate the 2^n factor in equation 1.

In this instance the computer was instructed to loop through the display program, continuously displaying the analyser data break transfer locations in the computer memory. The system successfully picked out the input sequency due to the test generator, the error at other sequencies was plus or minus one bit. This was due to an

unavoidable 3 to 4 milivolt noise at the input of the A.D.C.
(coupling through from the computer) which randomly effected the
least significant bit.   An error in the computed coefficient of
1-3% was observed which was due to the difficulty of setting the
gain of the summing amplifier of the lowpass filter and the gain
and zero offset of the A.D.C.

4.4 Computer Programs

Two simple programs (as listed in Appendix A) were developed
for the computer analyser system using the P.D.P.-8 PAL-III
assembler language.[18]   The first of these programs, as shown in
flow chart form in Figure 4-3, is a display loop which can display
any preselectable number of locations in memory with a selectable
display time for each memory location. The behaviour of the sequency
spectrum between integer sequencys* eliminates the need to display
the spectra as intensified points;   the method adopted was to display
each memory content as a horizontal line whose width is determined
by the program dwell time and the sweep time setting of the
oscilloscope.   A division by four of the displayed information is
inherent in the display hardware because the D.A.C. is a 10 bits
offset binary converter, this latter fact necessitates conversion
from 2's complement to offset binary code.   It should be noted
that the tally and delay locations in memory hold the complements of
the number of locations to be displayed and the program dwell time
for each location to be displayed respectively, the auto index
register is a location in memory which is automatically incremented
if addressed by an indirect memory reference instruction.

* See Chapter 2
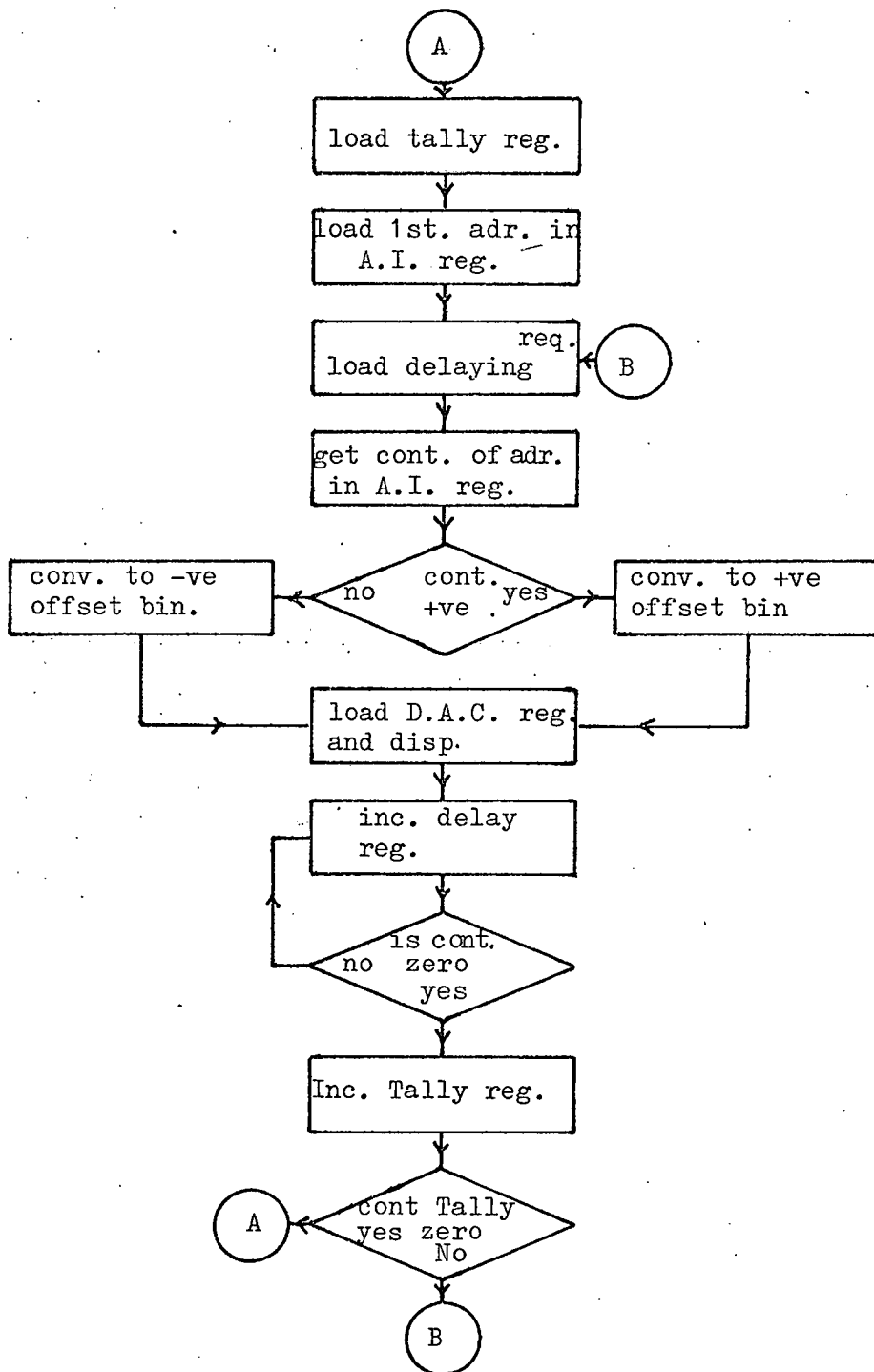
Figure 4-3

Display loop flow chart

Due to the large numbers anticipated when averaging over long periods of time and the dynamic range of these numbers, a program based on the P.D.P.-8 extended floating point package was developed.[17] Essentially this program activates the computer interrupt circuitry and puts the computer in a waiting mode. On the arrival of an interrupt signal the program transforms the new set of coefficients to floating point format squares them and adds them to a running sum location and then reverts to the waiting mode. At the end of a preselectable number of interrupts the program deactivates the interrupt circuitry then forms the sum of the spectral pairs in the running sum location and their scaled logarithms. At the end of the preceeding operations the program reverts to the display loop routine.

The above program is made up of three routines; the first of these as shown in flow chart form in Figure 4-4 is a house keeping routine which clears the running sum locations, loads the number of interrupts location and holds the computer in the waiting mode. On receiving an interrupt request signal control is passed to the second routine through memory page zero by a jump to subroutine instruction. The second routine, as shown in flow chart form in Figure 4-5, converts the new set of coefficients to floating point format, squares each coefficient and adds it to a specified running sum location. At the end of the foregoing set of computations the routine checks whether the preselected number of interrupts has been reached and accordingly returns control to the first routine or deactivates the interrupt circuitry by modifying the interrupt instructions in memory page zero and hands over control to the third routine.

```
          ( A )
            │
            ▼
┌───────────────────────┐
│ res. running sum      │
│ loc.                  │
└───────────────────────┘
            │
            ▼
┌───────────────────────┐
│set no of int.         │
│location               │
│                       │
└───────────────────────┘
            │
            ▼
┌───────────────────────┐
│                       │
│activate int.          │
│cct.                   │
└───────────────────────┘
            │
            ▼
        int.
      request  yes ──────►
        no              ( B )
            │
            ▼
        int.
     no request yes ────►
```

```
          ( B )
            │
            ▼
┌───────────────────────┐
│save current adr.      │
│+1 in loc. zero        │
└───────────────────────┘
            │
            ▼
          ( C )
```
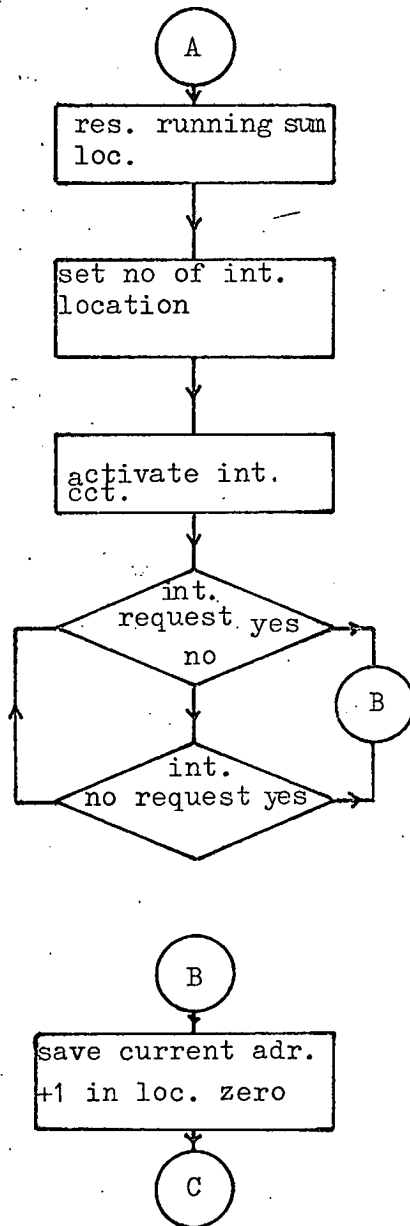
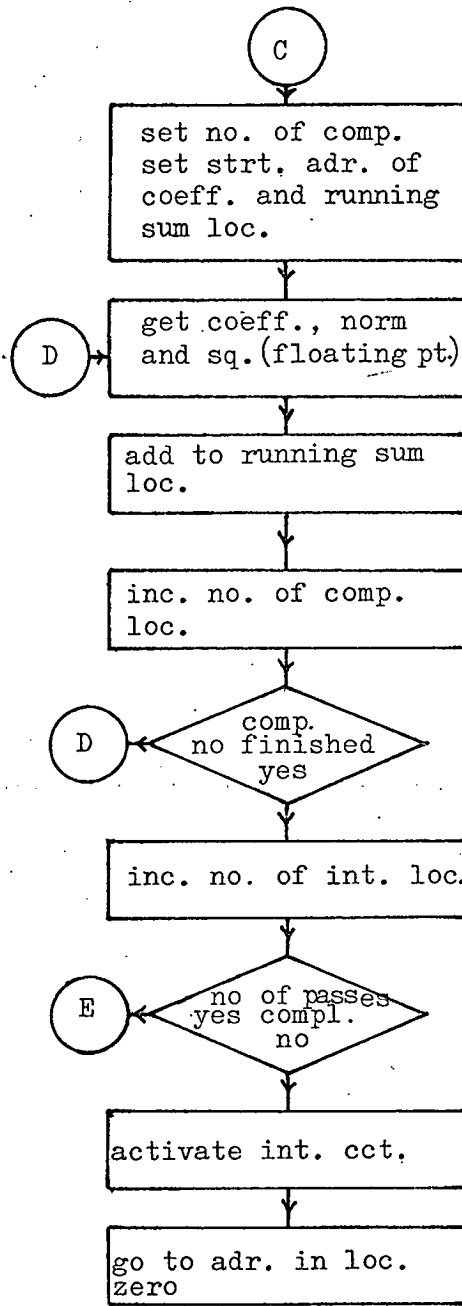Figure 4-4

Initialise routine flow chart

Figure 4-5

Square and sum coefficients routine

flow chart

After the modification of the interrupt instructions the third routine as shown in Figure 4-6 is entered through a subroutine which sums the sequency pairs in the running sum location. The natural logarithm of the sequency pairs is formed and the logarithm of the number of interrupts is subtracted thus forming a smoothed spectral point. The foregoing result is multiplied by a scaling factor $(100 \log_{10} e)$ and converted to fixed point format then deposited in the location to be displayed. The results are multiplied by a scaling factor to make use of the dynamic range of the display D.A.C. At the end of these computations control is passed over to the display loop program.

4.5 Sequency Amplitude Spectra of Sinusoids.

The response of the analyser to various sinusoidal frequencies in the range of 1-16 cycles/sec was investigated. This necessitated the calculation of individual coefficients of the relevant Walsh functions. Bobwetter[2] has shown that the coefficients of the Walsh functions in the Walsh series representing a sinusoidal waveform can be easily calculated from a recursive digital pattern. Another method for calculating the Walsh coefficients is due to G.S. Robinson and R. Granger[19] where the coefficients are derived from the Z-transform representation of the Walsh functions, but this fails at higher frequencies due to the absence of a $\frac{\sin x}{x}$ multiplier. A simpler method for calculating the relevant coefficients is outlined in Appendix B where it is shown that only the number of zero crossings of a Walsh function need be known to calculate the corresponding coefficient.
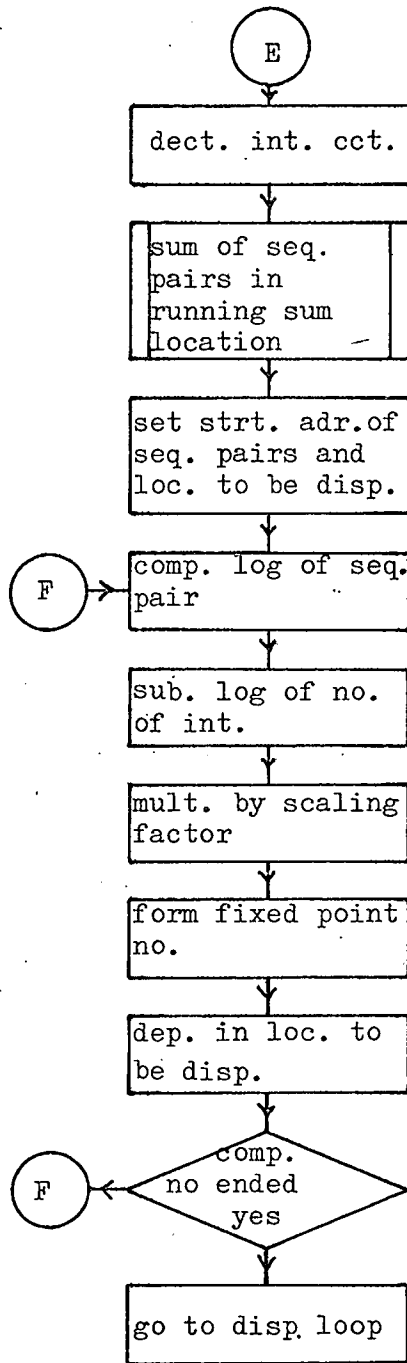
Figure 4-6

Spectral points routine

flow chart

Figures 4-7, 4-8, 4-9 and 4-10 show the amplitude sequency spectra of 5, 6, 11 and 15 cycle/sec. sinusoids of 2.82 volts r.m.s. amplitude and zero phase shift (sine-wave) and one second computation time. The spectrum was arranged as alternating pairs of odd and even Walsh coefficients of sequencies 1 - 50. Due to the difficulty of setting the gain of the low pass filter and A.D.C. all the computed coefficients of the sine-waves shown were within 5% of the predicted values as given in Table 4-1. The ratios of the computed coefficients of each frequency were calculated and they are in good agreement with the values calculated (within 1.2%) by the method outlined in Appendix B.

To show the effect of phase shift on the amplitude sequency spectrum of a sine wave, a frequency of 16 cycles/sec was chosen since this has only two coefficients in the sequency band 1 - 50. Figure 4-11 shows the amplitude spectrum for various phase shifts. Table 4-2 lists the various phase shifts as set on the trigger circuit versus the phase shifts as calculated from the ratios of the computed coefficients. For these measurements great care was taken in setting the gain of the low pass filter and A.D.C. which resulted in an error of a maximum of 1.4% in the amplitudes as calculated from the coefficients of the sequency of 16 at various phase shifts.

4.6 Sequency Power Spectra of Integer and Fractional Sinusoidal Frequencies

As outlined in Appendix B, equation 11 is an expression for the averaged power density at each sequency due to a fractional frequency input

$$\left[\frac{S^2(\omega,m) + C^2(\omega,m)}{2}\right] + \frac{1}{2p}\left[C^2(\omega,m) - S^2(\omega,m)\right]\cos(2\alpha + (\rho-1)\theta)\frac{\sin\rho\theta}{\sin\theta} \qquad (2)$$
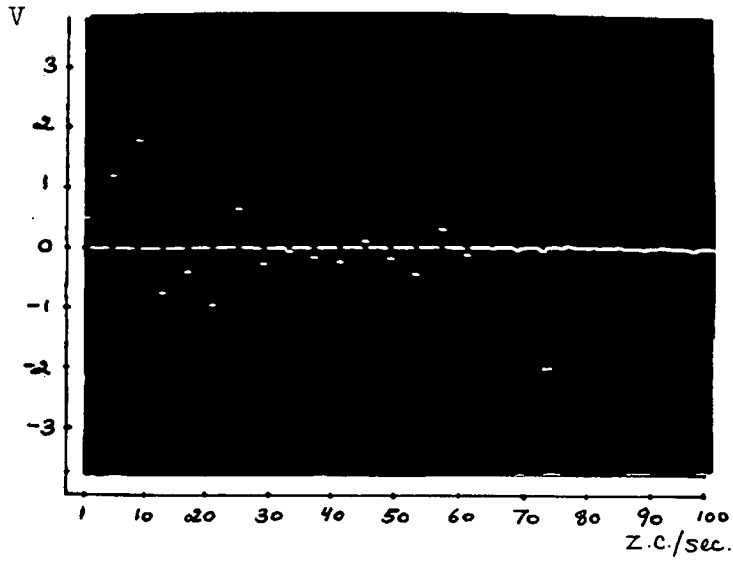
Fig.   4-7
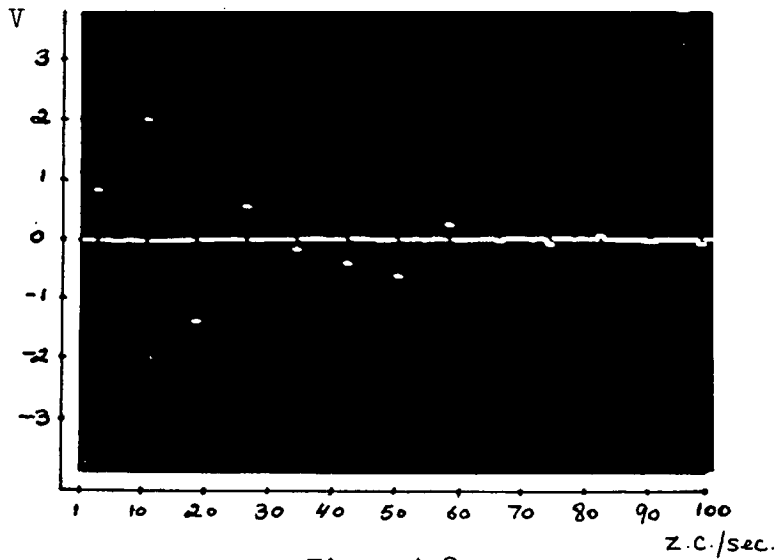
f=5 c/s

Fig.   4-8

f=6 c/s

Walsh coefficients of 5c/s and 6c/s sine waves
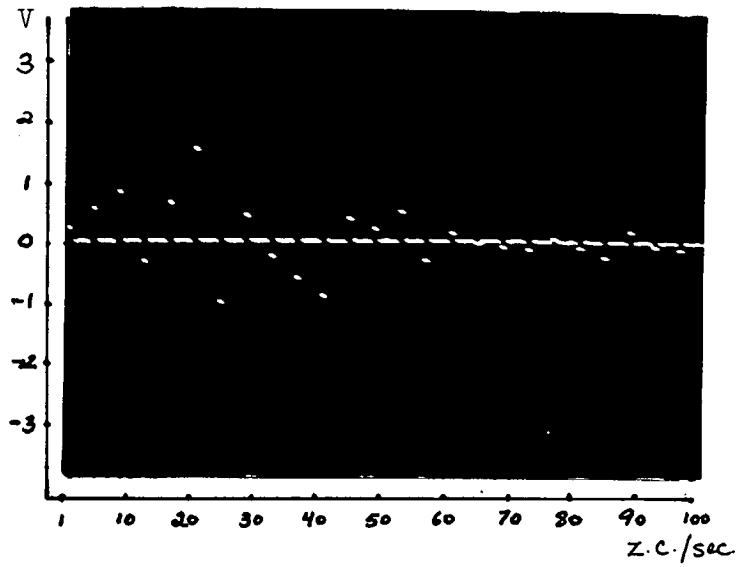
Fig. 4-9

f=11c/s
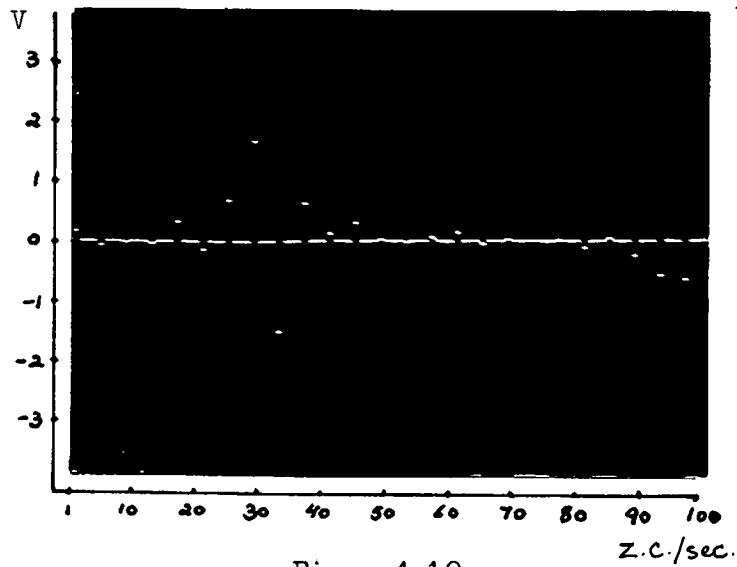
Fig. 4-10

f=15 c/s

Walsh coefficients of 11c/s and 15c/s sine waves

(in the sequency range 1-50)

f = 5 c/s

sal(n)	coeff.*	ratio of coeff.	ratio calc.	% error
1	200	2.415	2.4156	-.02
3	483	1.494	1.49	+.26
5	722	2.407	2.403	+.16
7	-300	1.863	1.872	-.48
9	-161	2.403	2.392	-.45
11	-387			
13	258	1.5	1.497	+.2
sal(5)coeff.	sal(5)coeff.			% error
750	722			-3.4

a – see Figure 4-7

f = 6 c/s

sal(n)	coeff.*	ratio of coeff	ratio calc.	% error
2	340	2.41	2.39	+.8
6	820	1.4936	1.4978	-.2
10	-549	2.418	2.415	+.12
14	227	3.29	3.33	-1.2
18	-69	2.42	2.44	-.8
22	-167			
26	-250	1.497	1.5	-.2
.sal(6)coeff	sal(6)coeff			% error
836	820			-1.9

b – see Figure 4-8

f = 11 c/s

sal(n)	coeff*	ratio of coeff	ratio calc.	% error
7	-136	1.86	1.87	+.5
9	253	2.411	2.413	-.1
11	610	1.484	1.496	-.8
13	-411	2.446	2.43	+.65
15	168			
17	-102	1.647	1.659	.75
sal(11)coeff	sal(11)coeff			error
640	610			-4.6

c – see Figure 4-9

f = 15 c/s

sal(n)	coeff*	ratio of coeff	ratio calc.	% error
7	-14	9.5	10.15	-6.4
9	133	2.375	2.414	-1.2
11	-56	4.984	5.028	-.9
13	279	2.408	2.414	-.3
15	672			
17	-611	1.1	1.103	-.27
sal(15)coeff	sal(15)coeff			% error
704	672			-4.5

d – see Figure 4-10

Table 4-1

Measured and calculated Walsh coefficients

*multiply by 2.44 x 10^{-3} volts

(a) - θ = 0°
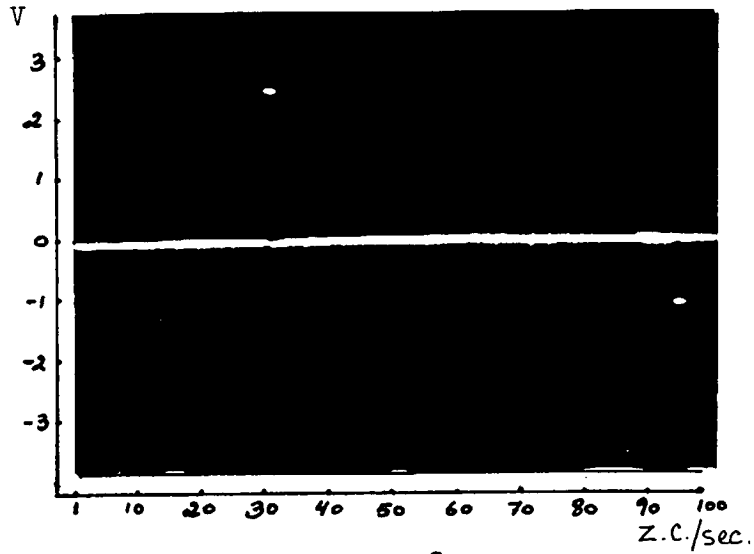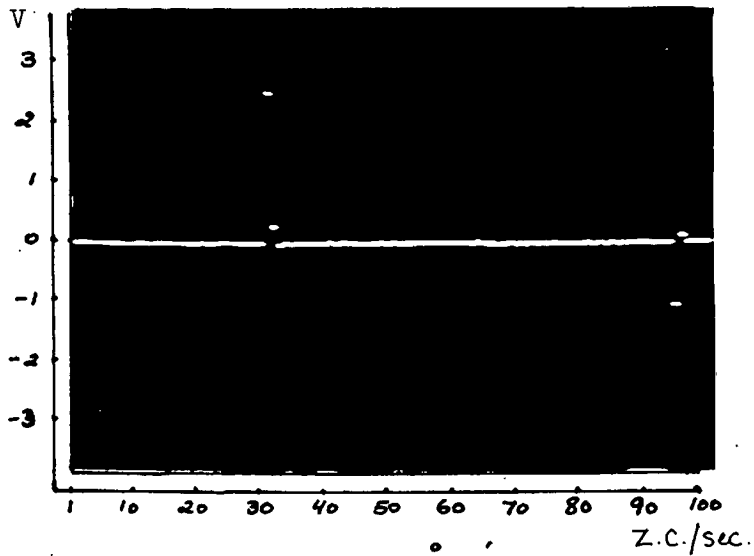
(b) - θ = 7° 8'

Fig. 4-11

f = 16-(a,b)

Effect of phase shift on the Walsh coefficients
of a 16 c/s sine wave

(c) – θ = 30° 15′

(d) – θ = 74° 14′
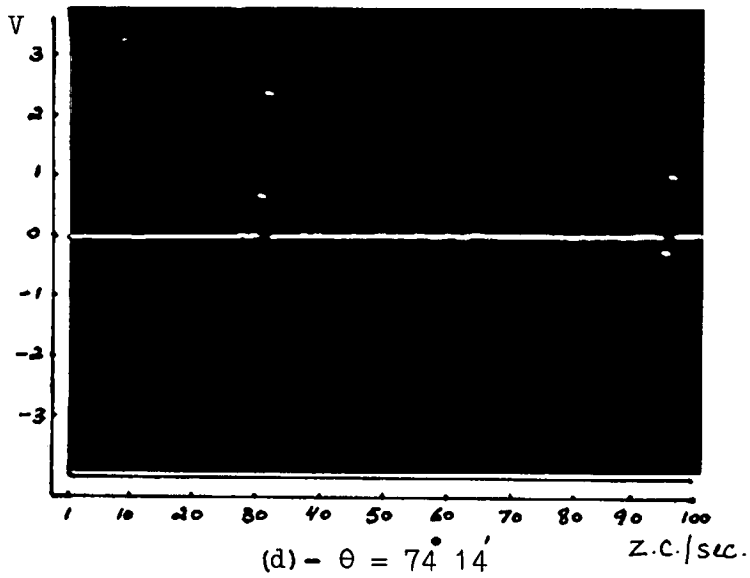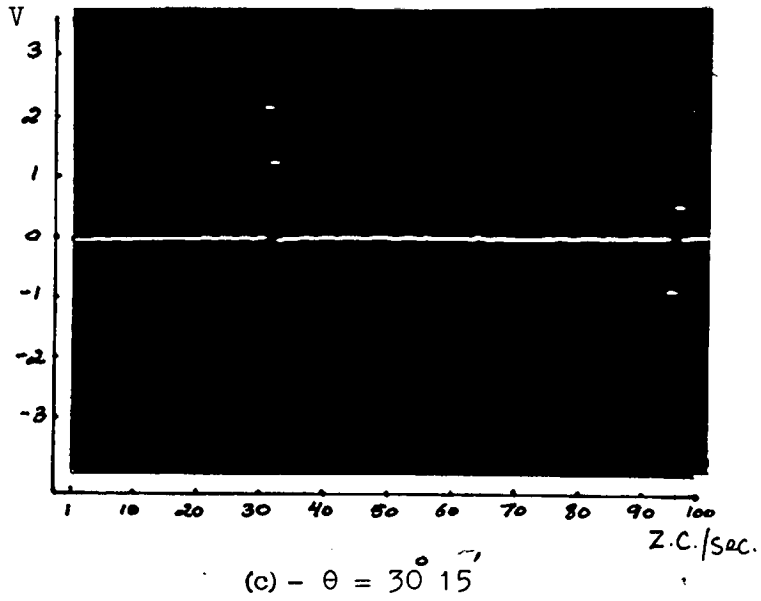
Fig. 4-11

f = 16-(c,d)

Effect of a phase shift on the Walsh coefficients
of a 16 c/s sine wave

(e) - θ = 83° 15'

(f) - θ = 89° 15'
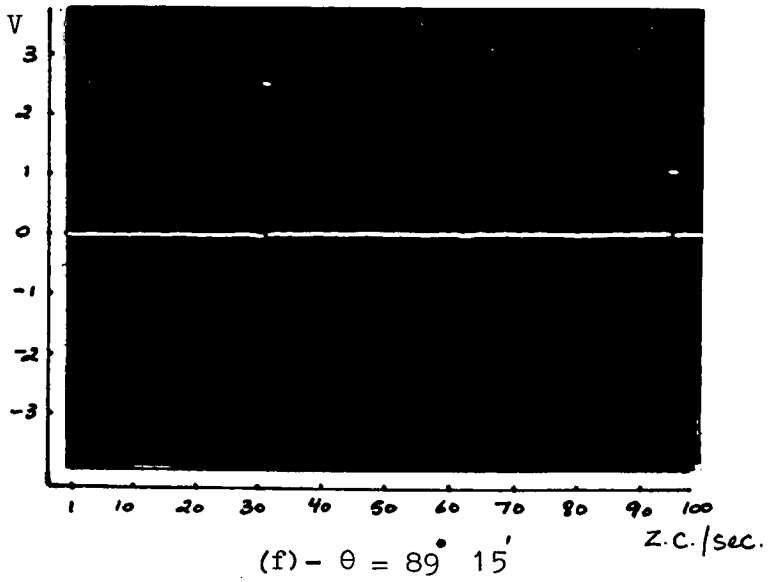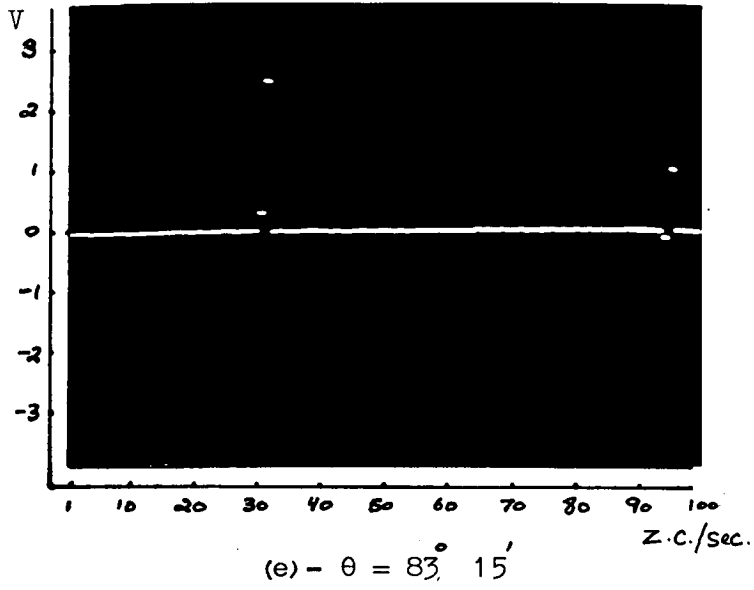
Fig.   4-11

f = 16-(e,f)

Effect of phase shift on the Walsh coefficients
of a 16 c/s sine wave

$\alpha°$ set	coeff sal 16 *	coeff cal 16 *	coeff sal 48 *	coeff cal 48 *	$\dfrac{\text{cal}}{\text{sal}}$ 16	$\dfrac{\text{cal}}{\text{sal}}$ 48	$\tan^{-1}\dfrac{C_{16}}{S_{16}}$	$\tan^{-1}\dfrac{C_{48}}{S_{48}}$	S or C $\sin$ or $\cos$ $\tan^{-1} C_{16}/S_{16}$	%* error of calc. coeff
0°	1029	0	-429	0	0	0	0	0	1029	-1.1
7°	1021	128	-427	53	.1253	.1241	7°8'	7°4'	1029	-1.1
29°	888	518	-372	214	.5833	.5752	30°15'	29°55'	1026	-1.4
74°36'	280	992	-114	409	3.5428	3.5877	74°14'	74°25'	1033	-.7
83°35'	124	1023	-51	423	8.25	8.294	83°6'	83°7'	1032	-.8
89°15'	-13	1030	5	428	79.23	85.6	90°44'	90°40'	1031	-1

Table 4-2

Sequency coefficients of sinusoidal frequency 16 c/s
for various phase shifts

* (1) multiply by 2.44 x 10^{-3}V

   (2) 2.82 volts R.M.S. = 3.987 volts = decimal 1634

   (3) absolute value of coeff. at sequency 16 = 1041

where S and C are the sal and cal coefficients of Walsh functions of sequency m, α is an arbitrary starting phase angle, θ and ρ are the incremental phase shift and the number of power densities at a sequency of m that have been averaged. For integer frequencies the phase dependant part of equation 2 will be zero since S = C . Table 4-3 lists the peak power densities of various integer frequencies and the second largest power densities at the sequency harmonics of these frequencies for an averaging time of sixty seconds and input amplitude of two volts. The error in terms of amplitude and coefficient ratios for the listed frequencies were within 2.7% and 3.4% respectively thus showing good agreement with the calculated values.

Sixty second runs of two volt amplitude fractional frequencies in the range 2 cycles/sec to 3 cycles/sec at $\frac{1}{4}$ cycle/sec intervals and 13 cycles/sec to 14 cycles/sec., with the same frequency intervals were recorded. The choice of the latter range of frequencies was made since the peaks of the sequency power spectra for the 13 cycles/ sec and 14 cycles/sec input frequencies differed by less than 3db from the second largest peaks in the respective spectra. The sequency power spectra for frequencies of 2.25, 2.5 and 2.75 are shown in Figure 4-12. Table 4-4a lists the power densities of sequences 2 and 3 for two sets of runs. The corresponding errors were computed relative to the phase independant term of equation 2. The possible variation that might be introduced by the phase dependant term of equation 2 is listed as a percentage of the phase independant term. Figure 4-13 shows the sequency power spectra for frequencies 13.25, 13.5 and 13.75 and a corresponding listing of the power densities at sequences of 13 and 14 is shown in Table 4-4b. The phase dependant term of equation 2 is multiplied by the expression

freq. c/s	peak power density at seq.	power density db*	harmonic with highest power density	power density db*	true ratio	error from true ratio %	error in abs. value of coeff.
1	1	54.16	3	46.82	2.412	−3.4	−2.1
2	2	54.25	6	46.47	2.412	+1.6	−1.1
5	5	51.41	3	48.02	1.49	−.8	−1
8	8	54.24	24	46.56	2.412	+.4	−1.1
12	12	52.26	20	48.71	1.497	+.5	−2.7
13	13	49.92	19	47.21	1.38	−.87	−2.1
14	14	51.26	18	49.51	1.221	+.33	−2.6
16	16	54.18	48	46.58	2.412	−.58	−1.9
20	20	51.41	12	47.91	1.49	+.4	−1

Table 4-3

Sequency power densities for frequencies $1 < f \leqslant 20$
and 60 second computation times

*add $10 \log_{10}(2.44 \times 10^{-3})^2$

(a) f = 2.25 c/s

(b) f = 2.5 c/s

(c) f = 2.75 c/s
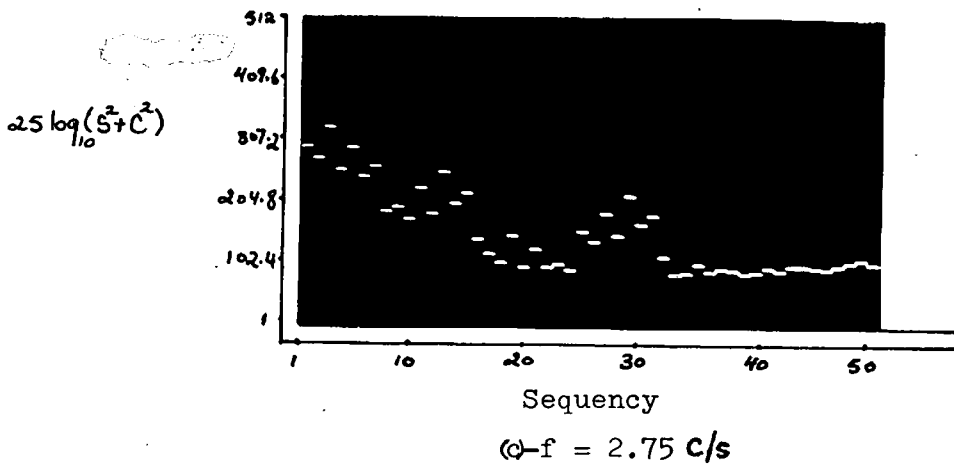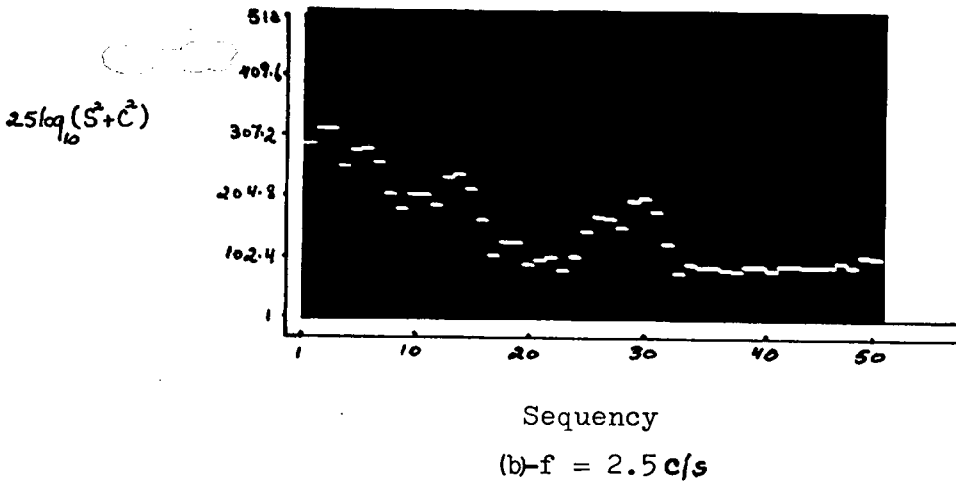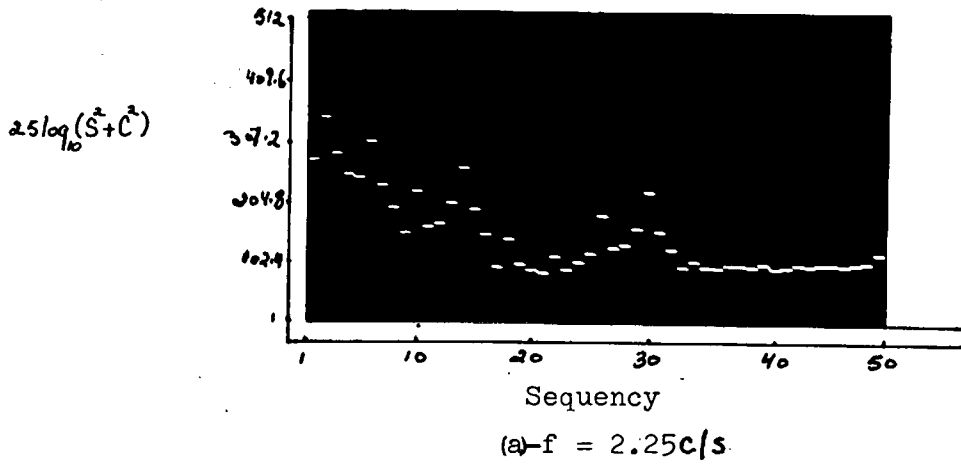
S ≡ coeff. of odd Walsh Function ; C ≡ coeff. of even Walsh Function

Fig. 4.12

Sequency power spectra of 2.25 c/s, 2.5 c/s, 2.75 c/s sine waves for 60 sec. run

(a)-f = 13.25 c/s

(b)-f = 13.5 c/s
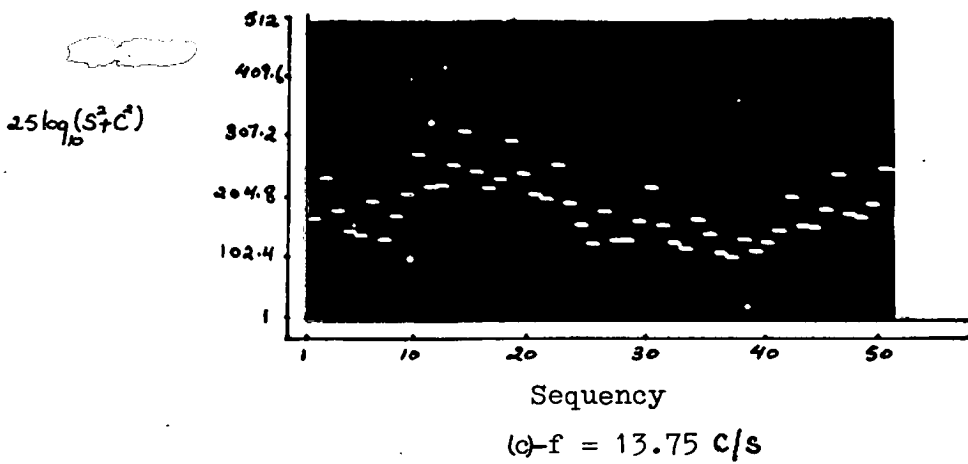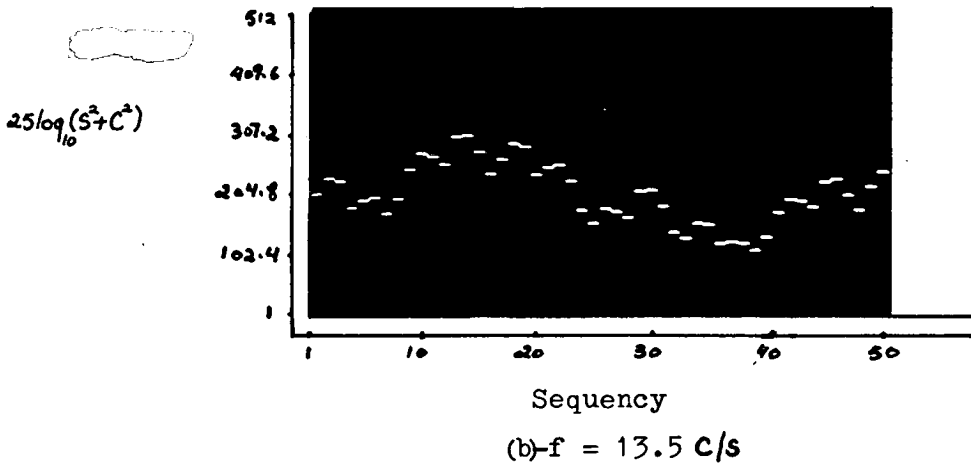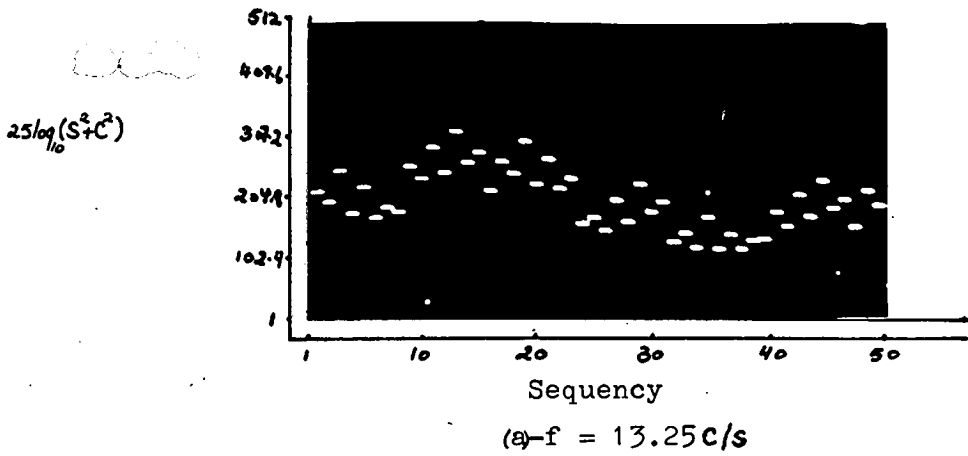
(c)-f = 13.75 c/s

$S \equiv$ coeff. of odd Walsh Function ; $C \equiv$ coeff. of even Walsh Function

Fig. 4-13

Sequency power spectra of 13.25 c/s, 13.5 c/s, 13.75 c/s sine waves for 60 sec. run

f c/s	power seq. 2 db*		calc power	% error	power at seq. 3 db*		calc power	% error	%phase dep. power to indep. power	
	run 1	run 2			run 1	run 2			seq.2	seq.3
2.25	53	53	53.23	-5.5	43.3	43.3	43.48	-4.4	±.32	±.76
2.5	49.5	49.7	50.11	-15.2	49.4	49.1	49.2	+4.8	±38.7	±71
2.75	43.2	—	43.57	-9.1	51.4	—	51.74	-8.1	±1	±.64

(a)

f c/s	power seq. 13 db*		calc. power	% error	power seq. 14 db*		calc. power	% error	% phase dep. power to indep. power	
	run 1	run 2			run 1	run 2			seq. 13	seq. 14
13.25	48.5	48.7	49.25	-18	40.3	39.7	41.08	-20	±.63	±.93
13.5	46.4	46.5	46.63	-5.4	46.9	46.8	47.54	-15.9	±70.7	±38.3
13.75	40.4	40.8	40.8	-10	49.7	49.5	50.52	-20	±1.5	±.33

(b)

Table 4-4

Sequency power densities for frequencies

(a) $2 < f < 3$

(b) $13 < f < 14$

*add $10 \log_{10}(2.44 \times 10^{-3})^2$

$(\cos(2\alpha + (\rho-1)\theta).\frac{\sin\rho\theta}{\sin\theta})$ which has an envelope as shown in Figure

4-14. When the sum of the power spectral densities is divided by ρ

the maximum of the envelope will be 1. In the region of $\theta = 90^\circ$,

270° corresponding to a frequency which is an integer plus a $\frac{1}{4}$ cycle

or $\frac{3}{4}$ cycle the possible variation introduced by the phase dependant

term is very small as listed in Tables 4-7a and b. For the case of

$\theta = 180^\circ$ corresponding to a frequency which is integer plus a $\frac{1}{2}$ cycle

the phase dependant term is modified by $\cos 2\alpha$ only and is large

compared with the other terms in the expression as shown in Tables

4-4a and b. At the extreme of θ, that is in the region of $0^\circ < \theta < 10^\circ$

and $350^\circ < \theta < 360^\circ$ corresponding to a frequency which is integer $\pm .027$

cycle it can be shown that the absolute value of the phase dependant

term is very small since the terms S and C of equation 2 are nearly

equal. In the case of frequencies 2.25 cycles/sec and 2.75 cycles/sec

the calculated errors of 4.4% and 8.1% represent an amplitude

error of approximately 2.2% and 4% which is expected due to the gain

adjustment difficulties. The error for the 13.25 cycles/sec

frequency is approximately 20% corresponding to an amplitude error of

approximately 10%; the error in the case of a frequency of 13.75

cycles/sec cannot be satisfactorily explained.

4.7 Sequency Power Spectra of Sinusoids with Additive Noise

Sequency power spectra (in the sequency range 1 - 50) of

Gaussian noise averaged over sixty seconds were recorded. The

spectra for noise of 3db cut-off frequencies 5 cycles/sec, 15 cycles/

sec and 150 cycles/sec are shown in Figure 4-14, and Table 4-5 shows

a list of the 3db point and the power spectral density of the secondary

peaks. The shape of the noise sequency power spectrum can be
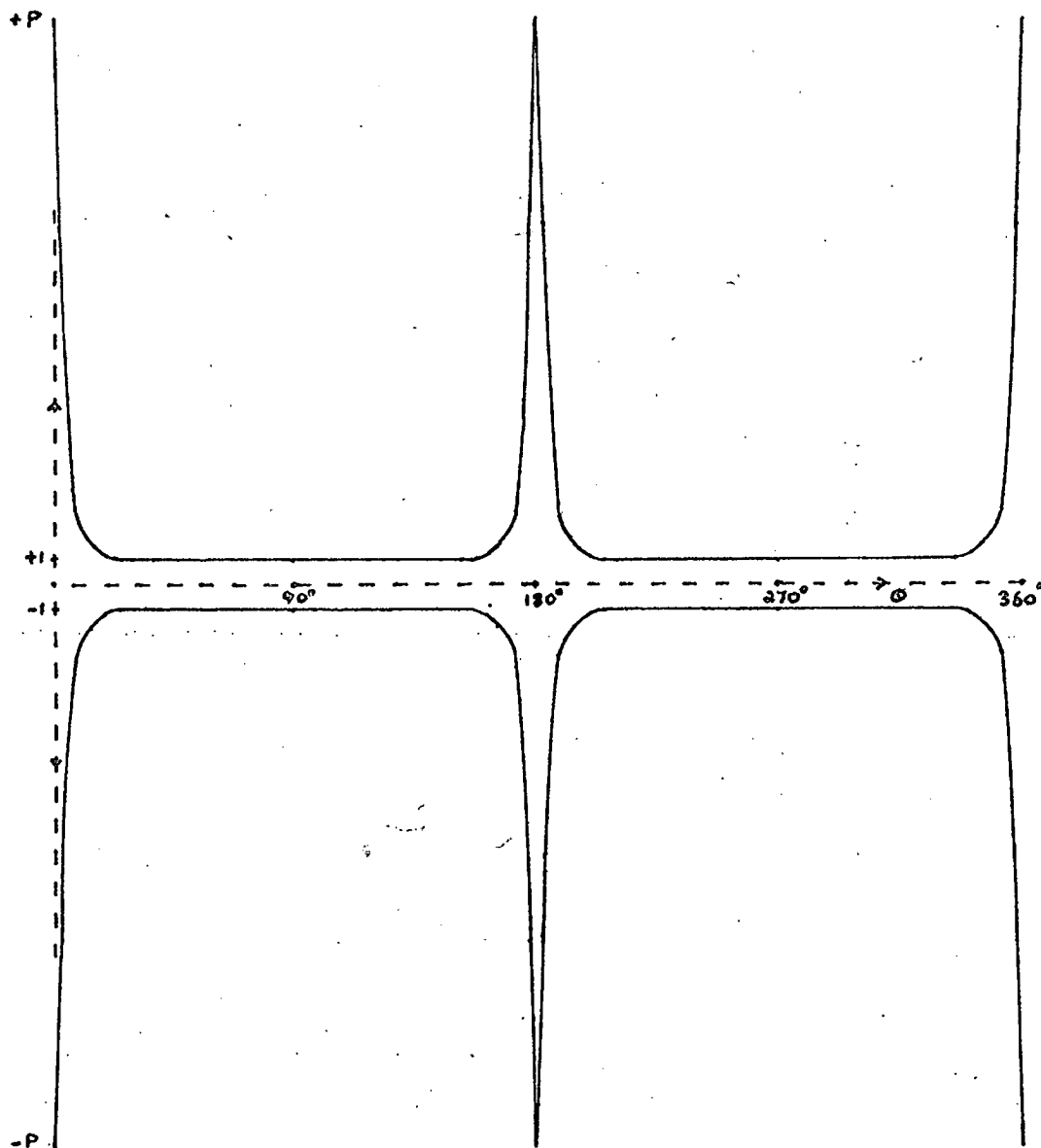
Figure 4-14

Envelope of the Function

$$\cos(2\alpha + (\rho-1)\theta) \; \frac{\sin\rho\theta}{\sin\theta}$$

$25 \log_{10}(S^2 + C^2)$

Sequency

(a)-5 c/s

$25 \log_{10}(S^2 + C^2)$

Sequency

(b)-15 c/s

$25 \log_{10}(S^2 + C^2)$

Sequency

(c)-150 c/s
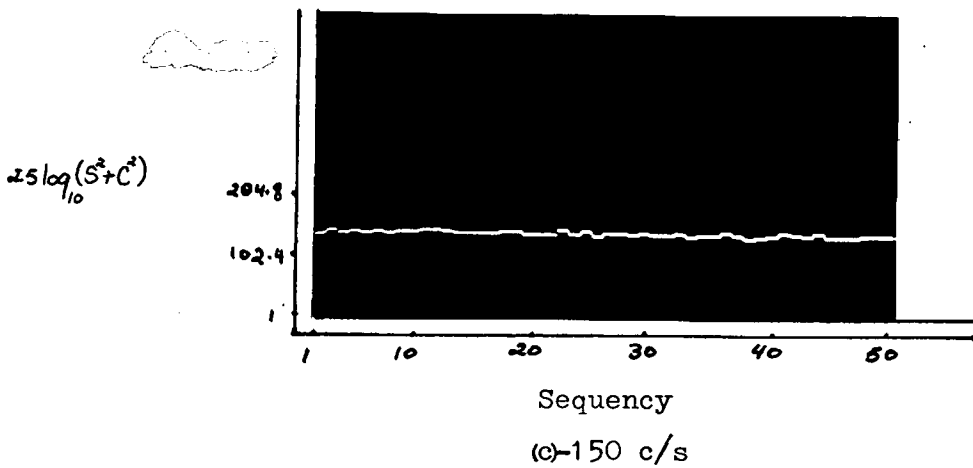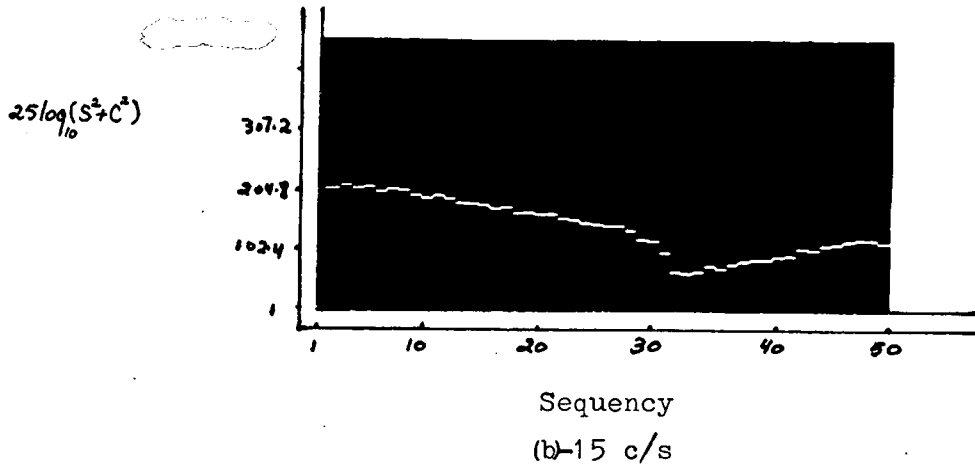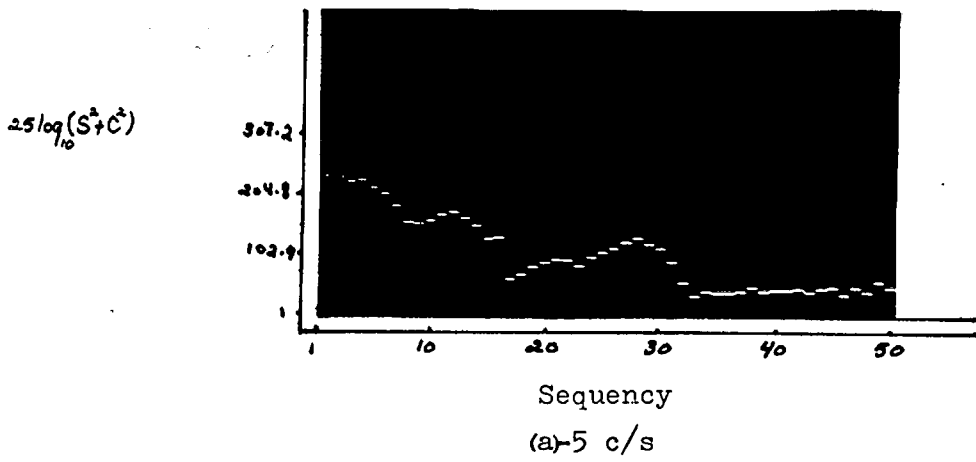
$S \equiv$ coeff. of odd Walsh Function; $C \equiv$ coeff. of even Walsh Function

Fig. 4-15

Sequency power spectra of gaussian noise of various cut off frequencies

Cut off frequency c/s	approximate 3db sequency	sequency of first peak	power density db*	sequency of second peak	power density db*
5	5	12	26.2	28	19.3
15	15	—	—	—	—

Table 4-5

Sequency power densities of Gaussian noise of various cut off frequencies

(see Figure 4-15)

*add $10 \log_{10}(2.44 \times 10^{-3})^2$

predicted by transforming the arithmetical auto-correlation function

of the input noise to the logical auto-correlation function and then
(20)
Walsh transforming the latter to obtain the sequency power spectrum.

In the case of white noise the arithmetical auto correlation

function is an impulse at the origin and zero elsewhere, thus the

logical auto-correlation function is an impulse at the origin as

well and Walsh transforming the latter will result in a constant

sequency power spectral density.   Calculation of the noise power

spectra was not attempted due to the large number of samples of the

auto-correlation function needed.

Power spectra of 150 cycles/sec noise mixed with a 13.5 cycle/

sec sin wave for increasing r.m.s. noise amplitude were recorded

for averaging times of 4 minutes.   Figure 4-16a shows the power

spectrum of the signal without noise, Figures 4-16b, c, d are the

power spectra of the signal mixed with successively increasing noise

r.m.s. amplitude.   The effects of noise on the power spectra of

Figures 4-16c and d are clearly visible, Figure 4-17 shows the input

signal corresponding to Figure 4-16d.

4.8 Sequency Power Spectra of Simulated E.E.G. Signal

An E.E.G. signal was simulated by randomly exciting a band-

pass filter centred at 13.88 cycles/sec.   Sequency power spectra

of this signal for averaging times of 4 minutes and 1 minute are

shown in Figures 4-18a and b, Table 4-6 is a listing of the power

densities of sequency 13, 14 and 18, 19.   The peaks of the power

spectrum corresponds to those for a frequency in the range of 13 to

14 cycles/sec.

$25 \log_{10}(S^2+C^2)$

Sequency

(a) - noise R.M.S. Amp. **0V**

$25 \log_{10}(S^2+C^2)$

Sequency
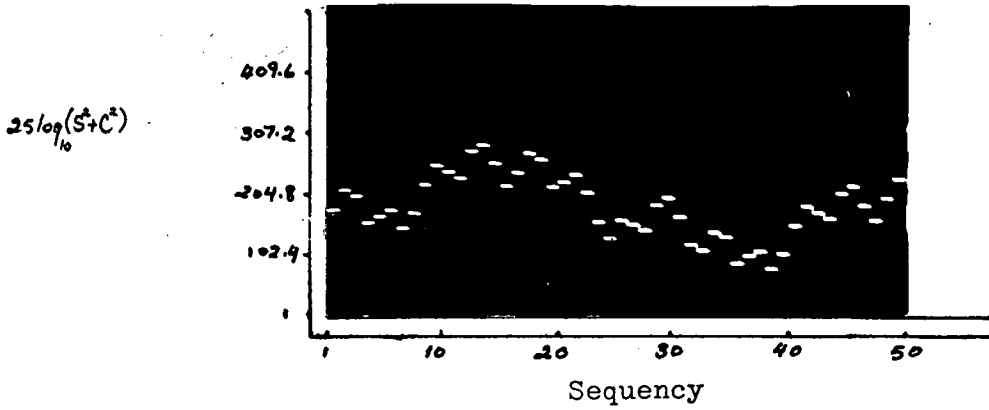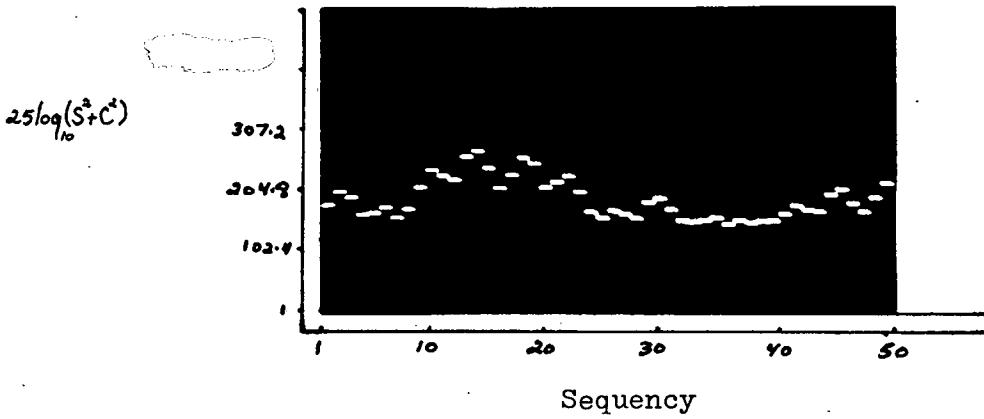
(b) - noise R.M.S. Amp. **·508V**

$25 \log_{10}(S^2+C^2)$

Sequency
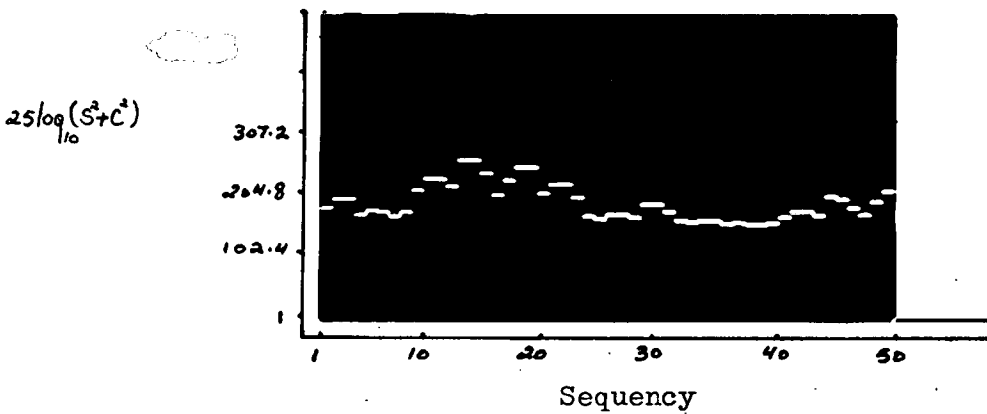
(c) - noise R.M.S. Amp. **·653V**

$S \equiv$ coeff of odd Walsh Function; $C \equiv$ coeff of even Walsh Function

Fig. 4-16

13.5 c/s plus noise

Amp of sinusoid 2 volts
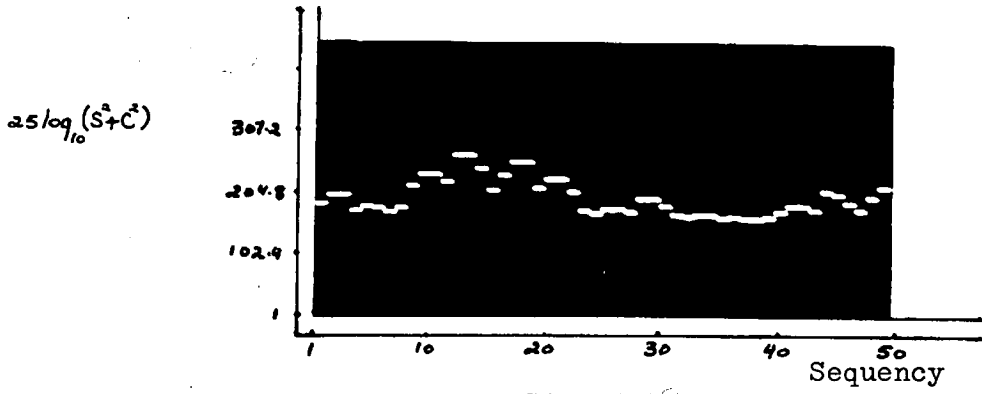
Fig. 4-16d

13.5 c/s plus noise R.M.S. amp. **.653**V
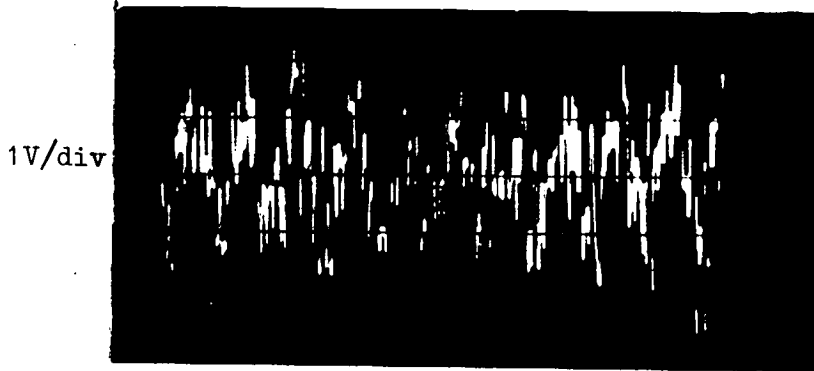
100 msec/div

Fig. 4-17

input signal of 4-16d

$25 \log_{10}(S^2+C^2)$

Sequency

(a) – 1 minute run

$25 \log_{10}(S^2+C^2)$
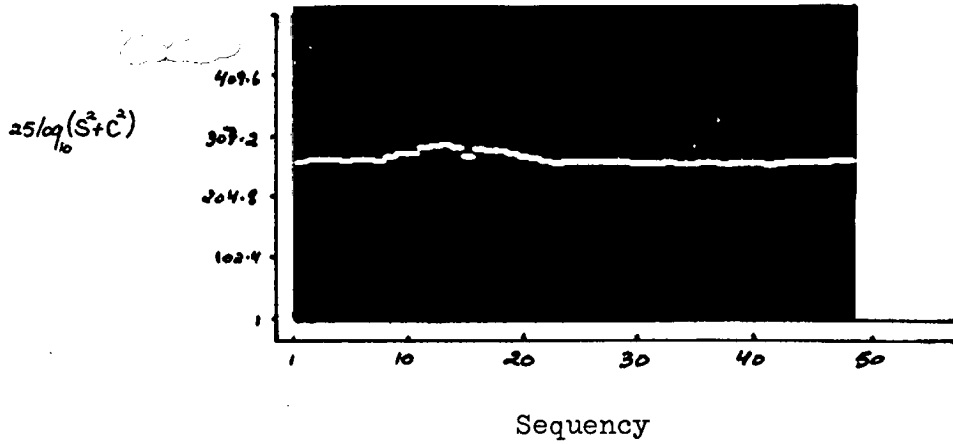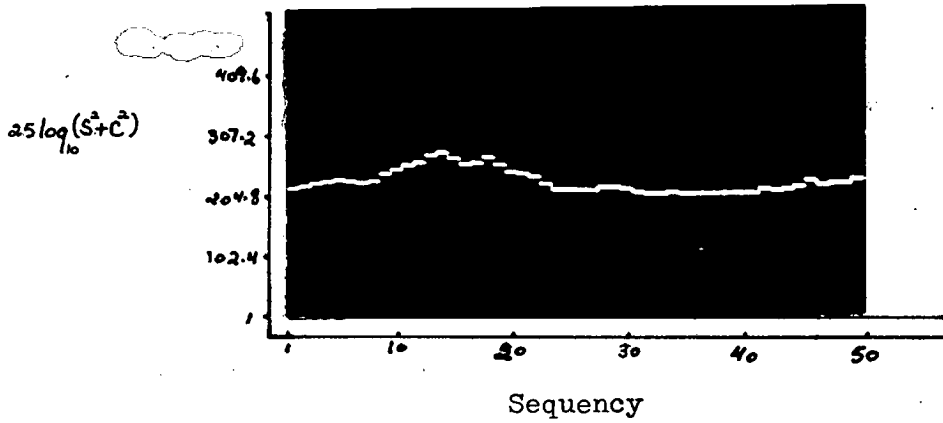
Sequency

(b) – 4 minutes run

$S \equiv coeff. of odd Walsh Function ; C \equiv coeff. of even Walsh Function$
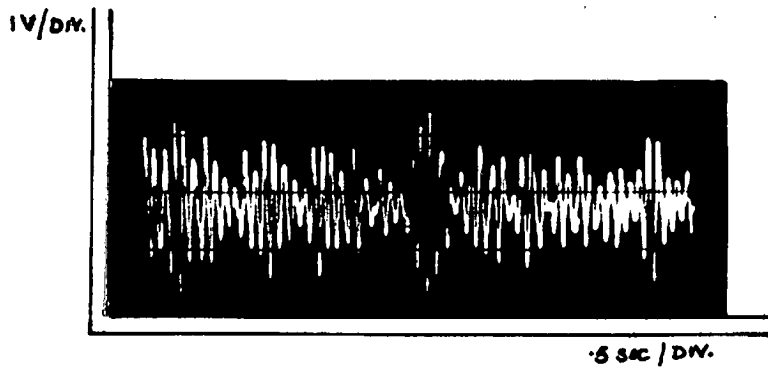
(c) – input signal

Fig. 4-18

Input signal of dominant frequency 13.88 c/s

averaging time minutes	power density at seq. 13 db*	power density at seq. 14 db*	power density at seq. 18 db*	power density at seq. 19 db*
1	43.4	43.8	42.2	42.3
4	41.6	42.7	41.4	39.5

Table 4-5

Sequency power densities of simulated E.E.G.

signal (see Figure 4-18)

*add $10 \log_{10}(2.44 \times 10^{-3})^2$

CHAPTER 5

CONCLUSIONS AND FURTHER WORK

5.1 Discussion and Conclusions

The starting phase dependance of Walsh coefficient spectra
has been shown to be a major difficulty with this method of signal
analysis.   However, it has been found that the effect of starting
phase can be reduced sufficiently to allow the dominant sequency
term to be detected.   To achieve this result a Walsh power spectrum
is formed and successively averaged.   A digital computer was
interfaced with a Walsh coefficient spectrum analyser and used to
mechanise the formation of the power spectrum.   Section 5.2 will
describe a proposal for a special purpose sequency power spectrum
analyser for low frequency signals.

It has been shown in Chapter 4 that the coefficient spectra
of waveforms having constant amplitude and integer frequencies have
maxima at sequencies corresponding to the input frequencies and all
odd harmonics of those sequencies.   Values of these maxima are
frequency dependant.   For the system constructed the maximum value
of all the maxima occurs for frequencies which are power of two
multiples of 1c/s corresponding to Walsh functions that are hard
limited sinusoids.   In general it can be shown from the coefficient
generation law* that frequencies which are power of two multiples of
an odd frequency will have the same value for their coefficients
but these will occur at sequencies which are the power of two multiples
of those for the odd frequency.   For example frequencies 2c/s, 4c/s,
8c/s, 16c/s or 6c/s, 12c/s, 24c/s can be considered to be power of
two multiples of 1c/s, 3c/s and 5c/s respectively.

* See Appendix B

Although the coefficient spectra of integer sinusoids in

the sequency domain are diffuse it is possible to extract the input

amplitude and phase information for a noise free sine wave from

one computation cycle since the Walsh coefficients can be calculated.

For frequencies that are integer plus a fraction, multiple averaging

of consecutive computations of the power densities will reduce the

phase dependence. Although, as has been shown, this averaging

scheme breaks down when the input frequency is integer plus half

a cycle; it should be noted that the power spectrum will still have

a maximum which occur either at the lower or upper sequency bounding

the input frequency as the starting phase angle varies. This is

shown in Table 5-1 which lists the calculated power densities at

sequencies 12, 13, 14, 15, 18 and 19 Z.P.S. for an input frequency

of 13.5 c/s and various starting phase angles. Equation 11 of Appendix

B was used to produce this table.

The indeterminacy of frequencies which are fractional can be

restricted to a smaller sequency increment by increasing the

resolution of the system. This implies a larger shift register

stack to cater for the same frequency band of interest as well as

increasing the total computation time for each spectrum.

The above difficulties associated with the phase angle will

not occur in the case of a sinusoide of the form $A \cos(+t + \theta(t))$

where $\theta(t)$ is a random phase angle. This follows because the phase

angle which occurs in the power density expression will be the mean

of $\theta(t)$. This is the case for the simulated E.E.G. signal which

is a sum of exponentially decaying sinusoids generated by a

randomly excited band pass filter.

Sequency	power density at angle α $(2.44 \times 10^{-3})^2 \times 10^4 \ v^2$				
	$\alpha=0°$	$\alpha=22.5°$	$\alpha=30°$	$\alpha=45°$	$\alpha=90°$
12	1.33	1.194	1.096	.859	.384
13	7.8	6.85	6.18	4.57	1.34
14	3.45	4.09	4.538	5.625	7.8
15	.593	1.013	1.319	2.026	3.46
18	.93	.852	.799	.6685	.407
19	.157	.27	.350	.5435	.93

Table 5-1

Calculated sequency power densities for a
frequency of 13.5 c/s at various starting
phase angles.

5.2 A Sequency Power Spectrum Analyser for Low Frequency Signals

It has been shown that the phase dependance of Walsh function spectrum analysis can be minimised by forming the running average of the sequency power spectrum. Hence a low cost all digital signal analyser for low frequency signals can be constructed. Tests on a simulated E.E.G. signal indicates that such a system could be used to detect the mean dominant frequency of E.E.G. signals. Townsend has shown that this frequency can be used to monitor the condition of patients having diseased livers.

The frequency range of interest in an E.E.G. signal (1c/s-16c/s) has most of its major coefficients occurring in the sequency band 1-16 Z.P.S. Therefore the storage and speed requirements of a spectral analyser covering the sequency band mentioned can be relaxed. A possible system based on the experimental sequency analyser constructed is shown in block diagram form in Figure 5-1. The eight bit A.D.C. (seven bits plus sign) was chosen since Townsend[27] has shown that this is adequate for the amplitude resolution of the analogue signal. For the computation of sixteen pairs of coefficients the first thirty-two Walsh functions (excluding the zero sequency W.F.) of the set of sixty four are used since the set of thirty-two does not include the cal(16,t) function to make up the highest sequency pair in the band of interest. To reduce the component count the Walsh functions can be stored in a read only memory as (1,0) patterns of thirty-two rows by sixty four columns as shown in Figure 5-2. Each row will be one Walsh function and any column will represent a sequency scan of thirty-two Walsh functions for the appropriate time increment. The read only memory shown has a capacity of 512, eight bit, words and the columns of the sixty-four by thirty-two Walsh matrix are assigned four words each. The least significant three bits
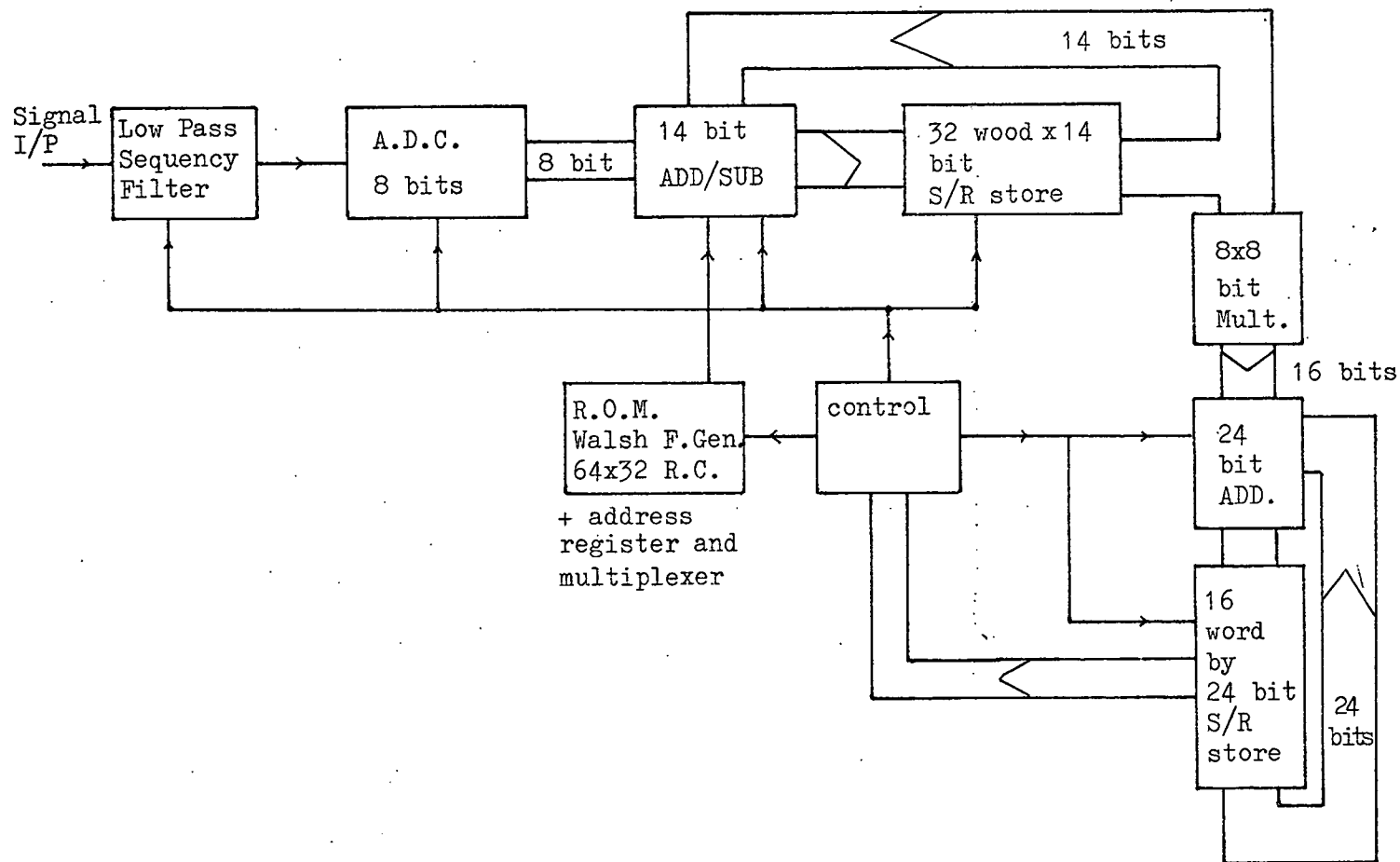
Figure 5-1

Block diagram of proposed sequency power spactrum
analyser.
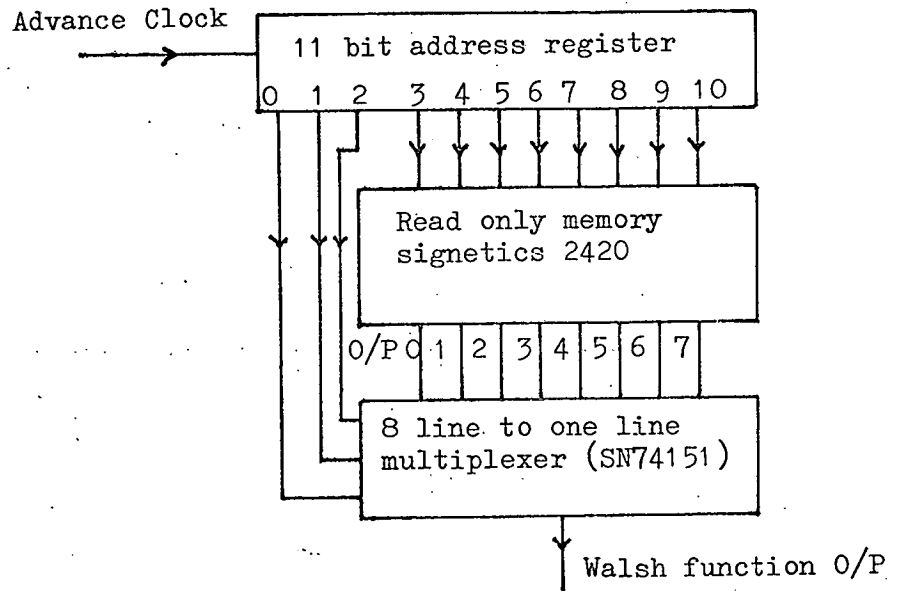
Figure 5-2

Read only memory Walsh function generator.

of the address register serialise the word presented to the

multiplexer. After the eighth clock pulse a new word is presented

so that in effect every thirty-two clock pulses will scan one

column of the sixty-four stored words. An alternative to the R.O.M.

shown would be two programable R.O.M.S. with a capacity of 256, four

bit words connected in parallel to the address lines.

The shift register stack word length of fourteen bits was chosen to

accommodate a full scale input of eight bits occurring in sixty-four

consecutive sub-intervals of the sequency low pass filter. This

store can be built of the signetics N2518 M.O.S.I.C. which is made

up of six, thirty-two bit shift registers and is T.T.L. compatible.

Taking the top most eight significant bits of the shift register

stack output will perform the required normalisation of the Walsh

coefficients. Choice of the Walsh function sequence length leads

to a low pass sequency filter sub-interval of 15.625 miliseconds.

This sub-interval is long enough for calculation of the partial sums

and the squaring and adding of the computed coefficients to the

running sum store. The sum of the squared coefficients of Walsh

functions of the same sequency requires a maximum of seventeen bits

therefore the word length of the running sum store was chosen to be

twenty-four bits long to give a maximum averaging time of 128 seconds.

The eight by eight bit multiplier and twenty-four bit adder

squares and adds two consecutive coefficients from the shift register

stack to a single location in the running sum store thus forming the

averaged spectral estimate at the end of the chosen averaging time.

The controller provides the control signals for the low pass filter

integrators and synchronises the clocking of the Walsh function

generator and the shift register stack in each computation sub-cycle.

At the end of each computation the controller generates the control signals that initiate the shifting of the coefficients out of the shift register stack and the squaring and adding of each sequency pair to a single location in the running sum store.   At the end of the averaging time the controller detects the largest number present in the running sum store, notes its sequency and stores it so that it might be displayed.   The detection scheme could utilise the first to overload principle.(9)

APPENDIX - A

PROCESSOR DISPLAY AND COMPUTATION PROGRAMS

(1) Display Loop Program

```
*200

SET,    CLA  CLL

        TAD  X      /set starting address in auto

        DCA  10     /index register.

        TAD  M      /set number of locations to

        DCA  N      /be displayed.

BEGN,   CLA  CLL

        TAD  Z      /set duration of display.

        DCA  Q

        TAD  I10    /get coefficient.

        SMA         /skip on negative A.C.

        JMP  PLS

        RAL         /form offset binary of negative number

        CLL         /in A.C.

        RTR

        RAR

        DYS         /load display register and display.

        ISZ  Q

        JMP  -1

        ISZ  N      /test if all coefficients have been

        JMP  BEGN   /displayed.

        JMP  SET

PLS,    CLL         /form offset binary of positive

        RAL         /number in A.C.

        GTL

        RAR
```

```
        RTR

        DYS          /load display register and display.

        ISZ  Q

        JMP  -1

        ISZ  N       /test if all coefficients have been

        JMP  BEGN    /displayed.

        JMP  SET

X,                   /starting address of coefficients.

M,                   /number of locations to be displayed.

N,           0000

Z,                   /duration of display of each location.

Q,           0000
```

(2) Computation Program

Initialisation Routine

*2200

```
        CLA  CLL

        TAD  TALY    /set number of locations to be

        DCA  INC     /cleared.

        TAD  ADRS    /set starting address of running sum

        DCA  11      /locations.

STRT,   CLA  CLL     /clear location whose address is

        DCA  I 11    /in auto index register 11.

        ISZ  INC

        JMP  STRT

        TAD  ION     /set up interupt instructions

        DCA  2       /in page zero.

        TAD  GOTTO

        DCA  1
```

```
                TAD   I PASS    /set number of interupts to

                CIA             /be processed.

                DCA   I NUMB

                ION             /activate interupt circuit

                JMP   + 1       /hold computer in waiting

                JMP   - 1       /mode.

TALY,   7324

INC,    0000

ADRS,   0715

ION,    6001

GOTO,   4404

PASS,   2062

NUMB,   2057
```

Floating Point Routine

```
*2000

COMP,   0000

                CLA   CLL

                TAD   COEF    /set starting address break request

                DCA   11      /locations.

                TAD   TALY    /set number of computations

                DCA   INC     /to be performed.

                TAD   ADRS    /set starting address of running sum

                DCA   12      /locations.

                TAD   PNTR    /set starting address of running sum

                DCA   LOC     /locations for floating point package.

STRT,           TAD   EXP1    /form floating point equivalent

                DCA   EXP     /of coefficient specified by address

                TADI  11      /in location 11.

                DCA   HORD

                DCA   LORD
```

```
        TAD    LOC      /set address of location to which
        TAD    3        /squared coefficient will be
        DCA    LOC      /added.
        JMS    I7       /call floating point package.
        FNOR            /normalise coefficient.
        FSQR            /square coefficient.
        FADDI  LOC      /add result to address in LOC.
        EXIT            /exit from floating point package.
        TAD    EXP      /put result in address
        DCA I  12       /specified in location 12.
        TAD    HORD
        DCA I  12
        TAD    LORD
        DCA I  12
        ISZ    INC      /check if all coefficients have
        JMP    STRT     /been squared and added.
        ISZ    NUMB     /check if preset number of interupts
        JMP I  COMP     /has been reached.
        JMP I  LOGI     /go to sum of squares sub-routine.
COEF,   0377            /starting address of coefficients-1.
TALY,   7634
INC,    0000
ADRS,   0715            /starting address of running sum locations-1.
PNTR,   0713
3,      0003
LOC,    0000
EXPI,   0013
NUMB,   0000
PASS,   nnnn            /preset numbers of interupts.
LOGI,   3400            /address of sum squares sub-routine.
```

Logarithm Routine

```
LOG,    CLA     CLL

        TAD     ADRSI       /set starting address of sum of

        DCA     12          /squares locations.

        TAD     TALY 1      /set number of computations.

        DCA     INC

        TAD     EXP1        /form logarithm of number

        DCA     EXP         /of interupts.

        TAD     PASS

        DCA     HORD

        DCA     LORD

        TAD     DISP        /set starting address of locations

        DCA     11          /to be displayed.

        JMS     I7          /call floating point package.

        FNOR                /normalise

        FLOG                /form natural logarithm

        F PUT B

        EXIT                /exit from floating point package.

STRT,   CLA     CLL

        TAD I   12          /get spectral sum and put

        DCA     EXP         /in floating A.C.

        TAD I   12

        DCA     HORD

        TAD I   12

        DCA     LORD

        JMS I 7             /call floating package.

        FLOG                /form natural logarithm.

        FSUB B              /smooth spectral point.

        FMPY A              /change to log.to base 10 and scale.
```

```
              EXIT           /exit from floating point package.
FIX,    CLA    CLL
        TAD    EXP    /form fixed point formate of number in
        SZA    SMA    /floating A.C.
        JMP.   +3
        CLA    CLL
        JMP    DONE+1
        TAD    -EXP1
        SNA
        JMP    DONE
        SMA
        JMP    ERROR
        DCA    EXP
GO,     CLL
        TAD    HORD
        SPA
        CML
        RAR
        DCA    HORD
        ISZ    EXP
        JMP    GO
DONE,   TAD    HORD   /get fixed point number and deposit
        DCA I  11     /in location to be displayed.
        ISZ    INC    /finished?
        JMP    STRT
        JMP I  DSPLY  /go to display routine.
ERROR,  CLA    CLL    /if number is too big to be fixed put
        TAD    MAX    /maximum number in location to
        DCA I  11     /be displayed.
        ISZ    INC
        JMP    STRT
        JMP I  DSPLY
```

```
-EXP1,    7765    /-13

A,        0006    /scaling factor equivalent to

          2557    /100 log₁₀e.

          0120

B,        0000    /location to store natural

          0000    /logarithm of number of interupts.

          0000

MAX,      3777

DSPLY,    0200    /starting address of display loop.

DISP,     0554    /starting address of locations to be displayed.

ADRS1,    1371    /starting address of spectral sums.

TALY1,    7716    /-50
```

Sum of Squares Routine

```
*3400
          CLA    CLL

          TAD    ADRS    /set starting address of running sum

          DCA    12      /locations.

          TAD    TALY    /set number of computations.

          DCA    INC

          TAD    PNTR    /set starting address of spectral sums.

          DCA    11

BGN,      CLA    CLL

          TAD I  12      /get floating point number from

          DGA    EXP     /address in location 12.

          TAD I  12

          DCA    HORD

          TAD I  12

          DCA    LORD

          TAD I  12      /get next floating point number from

          DCA    B1      /address in location 12.
```

```
        TAD I  12

        DCA    B2

        TAD I  12

        DCA    B3

        JMS    I7      /call floating point package.

        FADD   B       /add.

        EXIT           /exit from floating point package.

        TAD    EXP     /put result in location for

        DCA I  11      /spectral sums.

        TAD    HORD

        DCA I  11

        TAD    LORD

        DCA I  11

        ISZ    INC     /have all computations finished.

        JMP    BGN     /go to beginning.

        JMP I  GO      /go to address in GO.

TALY,   7716           /- 50.

INC,    0000

ADRS,   0715           /starting address of running sum locations.

PNTR,   1371           /starting address of spectral sums.

GO,     2063           /starting address of log. routine.

B B1,   0000

  B2,   0000

  B3,   0000
```

The locations designated by EXP, HORD and LORD are the locations of the floating accumulator in the floating point package.

APPENDIX - B

Amplitude Spectrum of Sinusoides and Walsh Functions

The integral equation

$$I(\omega,K) = \frac{1}{T} \int_0^T \text{Wal}(k,t) \cos(\omega t - \alpha)dt = \text{Re.}\frac{1}{T} \int_0^T \text{Wal}(k,t)e^{-j(\omega t - \alpha)}dt$$

.... (1)

T = interval of definition of the Walsh functions

α = relative phase shift

defines the sequency coefficients of the Walsh series

expansion of a sinusoidal or the fourier series expansion of a Walsh

function.

Due to the nature of the Walsh functions the integral

equation can be represented as a sum of sub-integrals whose number

is defined by the order of the highest sequency Walsh function

present in the set.   Therefore equation (1) becomes:

$$\frac{1}{T} \int_0^T \text{Wal}(k,t)e^{-j(\omega t - \alpha)}dt = \frac{e^{j\alpha}}{T} \left[\int_0^\tau \text{Wal}(k,t)e^{-j\omega t}dt + \int_\tau^{2\tau} \text{Wal}(k,t)e^{-j\omega t}dt + \dots \right.$$

$$\left. + \int_{(2^n-1)\tau}^{2^n\tau} \text{Wal}(k,t)e^{-j\omega t}dt \right]$$

.... (2)

where τ = smallest interval of constancy of the set of Walsh functions

defined by 2^n. n = order of the set of Walsh functions

letting T = 1   then $\tau = \frac{1}{2^n}$

Within the time interval τ the Walsh functions assume either of two

values +1 or -1 and therefore can be taken out of the integral signs,

hence equation (2) becomes:-

$$e^{j\alpha}\int_0^1 \text{Wal}(k,t)e^{-j\omega t}dt = \frac{e^{j\alpha}}{j\omega}\left[\text{Wal}(k,0)(1-e^{-j\omega\tau}) + \text{Wal}(k,\tau)e^{-j\omega\tau}(1-e^{-j\omega\tau})+..\right.$$

$$\left. + \text{Wal}(k,(2^n-1)\tau)e^{-j\omega(2^n-1)\tau}(1-e^{-j\omega\tau})\right]$$

$$= \frac{e^{j\alpha}(1-e^{-j\omega\tau})}{j\omega}\left[\text{Wal}(k,0) + \text{Wal}(k,\tau)e^{-j\omega\tau} +...+\text{Wal}(k,(2^n-1)\tau)e^{-j\omega(2^n-1)\tau}\right]$$

$$....(3)$$

let $a = e^{-j\omega\tau}$ then equation (3) becomes

$$\frac{e^{j\alpha}(1-a)}{j\omega}\left[\text{Wal}(k,0) + \text{Wal}(k,\frac{1}{2^n})a +.....+\text{Wal}(k,\frac{2^n-1}{2^n})a^{2^n-1}\right] ....(4)$$

For the set of two Walsh functions equation (4) has the values:-

k	binary k	gray code k	Walsh function		equation - 4
0	0	0	1	1	$(\frac{1-a}{j\omega}).(1+a)e^{j\alpha}$
1	1	1	1	-	$(\frac{1-a}{j\omega}).(1-a)e^{j\alpha}$

for the set of four Walsh functions:-

k	binary k	gray code k	Walsh function				equation - 4
0	0 0	0 0	1	1	1	1	$\frac{e^{j\alpha}(a-1)(1+a+a^2+a^3)}{j\omega}$

$$= \frac{e^{j\alpha}(1-a)(1+a)(1+a^2)}{j\omega}$$

1	0 1	0 1	1	1	-	-	$\frac{e^{j\alpha}(a-1)(1+a-a^2-a^3)}{j\omega}$

$$= \frac{e^{j\alpha}(1-a)(1+a)(1-a^2)}{j\omega}$$

2/

k	binary k	gray code k	Walsh function	equation - 4

$$2 \quad 1 \ 0 \qquad 1 \ 1 \qquad 1 - - 1 \qquad \frac{e^{j\alpha}(a-1)}{j\omega}(1-a-a^2+a^3)$$

$$= \frac{e^{j\alpha}(1-a)}{j\omega}(1-a)(1-a^2)$$

$$3 \quad 1 \ 1 \qquad 1 \ 0 \qquad 1 - 1 - \qquad \frac{e^{j\alpha}(a-1)}{j\omega}(1-a+a^2-a^3)$$

$$= \frac{e^{j\alpha}(1-a)}{j\omega}(1-a)(1+a^2)$$

and for the set of eight Walsh functions:-

k	binary k	gray code k	Walsh function	equation - 4

$$0 \quad 0 \ 0 \ 0 \qquad 0 \ 0 \ 0 \qquad 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \quad \frac{e^{j\alpha}(1-a)}{j\omega}(1+a+a^2+a^3+a^4+a^5+a^6+a^7)$$

$$= \frac{(1-a)}{j\omega}(1+a)(1+a^2)(1+a^4).e^{j\alpha}$$

$$1 \quad 0 \ 0 \ 1 \qquad 0 \ 0 \ 1 \qquad 1 \ 1 \ 1 \ 1 - - - - \quad \frac{e^{j\alpha}(1-a)}{j\omega}(1+a+a^2+a^3-a^4-a^5-a^6-a^7)$$

$$= \frac{(1-a)}{j\omega}(1+a)(1+a^2)(1-a^4).e^{j\alpha}$$

$$2 \quad 0 \ 1 \ 0 \qquad 0 \ 1 \ 1 \qquad 1 \ 1 - - - - 1 \ 1 \quad \frac{e^{j\alpha}(1-a)}{j\omega}(1+a-a^2-a^3-a^4-a^5+a^6+a^7)$$

$$= \frac{(1-a)}{j\omega}(1+a)(1-a^2)(1-a^4).e^{j\alpha}$$

$$3 \quad 0 \ 1 \ 1 \qquad 0 \ 1 \ 1 \qquad 1 \ 1 - - 1 \ 1 - - \quad \frac{e^{j\alpha}(1-a)}{j\omega}(1+a-a^2-a^3+a^4+a^5-a^6-a^7)$$

$$= \frac{(1-a)}{j\omega}(1+a)(1-a^2)(1+a^4)e^{j\alpha}$$

$$4 \quad 1 \ 0 \ 0 \qquad 1 \ 1 \ 0 \qquad 1 - - 1 \ 1 - - 1 \quad \frac{e^{j\alpha}(1-a)}{j\omega}(1-a-a^2+a^3+a^4-a^5-a^6+a^7)$$

$$= \frac{(1-a)}{j\omega}(1-a)(1-a^2)(1+a^4).e^{j\alpha}$$

k binary k gray code k Walsh function

5 1 0 1   1 1 1   1 - - 1 - 1 1 $- \dfrac{e^{j\alpha}(1-a)(1-a-a^2+a^3-a^4+a^5+a^6-a^7)}{j\omega}$

$= \dfrac{(1-a)(1-a)(1-a^2)(1-a^4).e^{j\alpha}}{j\omega}$

6 1 1 0   1 0 1   1 - 1 - - 1 - 1 $\dfrac{e^{j\alpha}(1-a)(1-a+a^2-a^3-a^4+a^5-a^6+a^7)}{j\omega}$

$= \dfrac{(1-a)(1-a)(1+a^2)(1-a^4).e^{j\alpha}}{j\omega}$

7 1 1 1   1 0 0   1 - 1 - 1 - 1 - $\dfrac{e^{j\alpha}(1-a)(1-a+a^2-a^3+a^4-a^5+a^6-a^7)}{j\omega}$

$= \dfrac{(1-a)(1-a)(1+a^2)(1+a^4).e^{j\alpha}}{j\omega}$

From the above expansions it can be seen that if the factors
$(1-a^{\gamma})$ and $(1+a^{\gamma})$ were equated to "1" and "0" respectively, then
each factored expression bears a simple relationship (gray code) to
the binary number representation of the relevant Walsh function
ordered in increasing number of zero crossings.   Therefore equation
(4) can be simply evaluated by converting the number k to gray code
and multiplying the respective factors corresponding to the "1"
and "0" in the gray code of k.   In general each factor $(1-a^{\gamma})$
or $(1+a^{\gamma})$ has the value:

$$1-a^{\gamma} = 1-e^{-jqx} = 2je^{-jq\frac{x}{2}} \sin q\frac{x}{2} = 2 \sin q\frac{x}{2}.e^{-j\left(q\frac{x}{2} - \frac{\pi}{2}\right)}$$

$$1+a^{\gamma} = 1+e^{-jqx} = 2e^{-jq\frac{x}{2}} \cos q\frac{x}{2} = 2 \cos q\frac{x}{2}.e^{-jq\frac{x}{2}}$$

where

$$x = \omega\tau = \frac{\omega}{2^n} \; ; \; q = 0,1,2,4,8,16,\ldots,2^n.; \; n = 0,1,2,3,4\ldots$$

in the case of the set of eight Walsh functions the expression for
the integral equation takes on the following values $\left(\tau = \frac{1}{8}\right)$:-

k	
0	$\dfrac{\sin\frac{\pi f}{8}}{\frac{\pi f}{8}}\cdot\cos\frac{\pi f}{8}\cdot\cos\frac{\pi f}{4}\cdot\cos\frac{\pi f}{2}\cdot e^{-j\frac{\omega}{2}}\cdot e^{j\alpha}$
1	$\dfrac{\sin\frac{\pi f}{8}}{\frac{\pi f}{8}}\cdot\cos\frac{\pi f}{8}\cdot\cos\frac{\pi f}{4}\cdot\sin\frac{\pi f}{2}\cdot e^{-j\frac{\omega}{2}}\cdot e^{j\left(\alpha+\frac{\pi}{2}\right)}$
2	$\dfrac{\sin\frac{\pi f}{8}}{\frac{\pi f}{8}}\cdot\cos\frac{\pi f}{8}\cdot\sin\frac{\pi f}{4}\cdot\sin\frac{\pi f}{2}\cdot e^{-j\frac{\omega}{2}}\cdot e^{j\left(\alpha+\pi\right)}$
3	$\dfrac{\sin\frac{\pi f}{8}}{\frac{\pi f}{8}}\cdot\cos\frac{\pi f}{8}\cdot\sin\frac{\pi f}{4}\cdot\cos\frac{\pi f}{2}\cdot e^{-j\frac{\omega}{2}}\cdot e^{j\left(\alpha+\frac{\pi}{2}\right)}$
4	$\dfrac{\sin\frac{\pi f}{8}}{\frac{\pi f}{8}}\cdot\sin\frac{\pi f}{8}\cdot\sin\frac{\pi f}{4}\cdot\cos\frac{\pi f}{2}\cdot e^{-j\frac{\omega}{2}}\cdot e^{j\left(\alpha+\pi\right)}$
5	$\dfrac{\sin\frac{\pi f}{8}}{\frac{\pi f}{8}}\cdot\sin\frac{\pi f}{8}\cdot\sin\frac{\pi f}{4}\cdot\sin\frac{\pi f}{2}\cdot e^{-j\frac{\omega}{2}}\cdot e^{j\left(\alpha+3\frac{\pi}{2}\right)}$
6	$\dfrac{\sin\frac{\pi f}{8}}{\frac{\pi f}{8}}\cdot\sin\frac{\pi f}{8}\cdot\cos\frac{\pi f}{4}\cdot\sin\frac{\pi f}{2}\cdot e^{-j\frac{\omega}{2}}\cdot e^{j\left(\alpha+\pi\right)}$
7	$\dfrac{\sin\frac{\pi f}{8}}{\frac{\pi f}{8}}\cdot\sin\frac{\pi f}{8}\cdot\cos\frac{\pi f}{4}\cdot\cos\frac{\pi f}{2}\cdot e^{-j\frac{\omega}{2}}\cdot e^{j\left(\alpha+\frac{\pi}{2}\right)}$

In the above set of expressions the term $e^{-j\frac{\omega}{2}}$ occurs because the integration is taken over the interval $(0,1)$ instead of $\left(-\frac{1}{2},\frac{1}{2}\right)$, therefore neglecting the term $e^{-j\frac{\omega}{2}}$ and taking the real part of the above set of expressions we will have:-

$$k = 0 \qquad C(\omega,0) = Q(\omega) \cos \frac{\pi f}{8} \cos \frac{\pi f}{4} \cos \frac{\pi f}{2} . \cos \alpha$$

$$k = 1 \qquad S(\omega,1) = -Q(\omega) \cos \frac{\pi f}{8} . \cos \frac{\pi f}{4} . \sin \frac{\pi f}{2} . \sin \alpha$$

$$k = 2 \qquad C(\omega,1) = -Q(\omega) . \cos \frac{\pi f}{8} . \sin \frac{\pi f}{4} . \sin \frac{\pi f}{2} . \cos \alpha$$

$$k = 3 \qquad S(\omega,2) = -Q(\omega) . \cos \frac{\pi f}{8} . \sin \frac{\pi f}{4} . \cos \frac{\pi f}{2} . \sin \alpha$$

$$k = 4 \qquad C(\omega,2) = -Q(\omega) . \sin \frac{\pi f}{8} . \sin \frac{\pi f}{4} . \cos \frac{\pi f}{2} . \cos \alpha$$

$$k = 5 \qquad S(\omega,3) = Q(\omega) . \sin \frac{\pi f}{8} . \sin \frac{\pi f}{4} . \sin \frac{\pi f}{2} . \sin \alpha$$

$$k = 6 \qquad C(\omega,3) = -Q(\omega) . \sin \frac{\pi f}{8} . \cos \frac{\pi f}{4} . \sin \frac{\pi f}{2} . \cos \alpha$$

$$k = 7 \qquad S(\omega,4) = -Q(\omega) . \sin \frac{\pi f}{8} . \cos \frac{\pi f}{4} . \cos \frac{\pi f}{2} . \sin \alpha$$

where $Q(\omega) = \dfrac{\sin \frac{\pi f}{8}}{\frac{\pi f}{8}}$; $C(\omega,m)$ and $S(\omega,m)$ are the coefficients of

the even and odd Walsh functions $Cal(m,t)$, $Sal(m,t)$ respectively and

m is the sequency which is given by

$$m = \begin{cases} \frac{k+1}{2}, \text{odd } k \\ \frac{k}{2}, \text{even } k \end{cases} .$$

When $\alpha = 0$ in the above set of expressions corresponding to a cosine

in equation (1) only the even coefficients are presented vice-versa

for $\alpha = \frac{\pi}{2}$ corresponding to a sine in equation (1).

For the set of eight Walsh functions the power density at each seqency is a sum of the squares of the coefficients of Walsh functions having that sequency. For the case of a sequency of (1) the power density is:

$$C(\omega,1)^2 + S(\omega,1)^2 = Q^2(\omega) \cos^2 \frac{\pi f}{8} \sin^2 \frac{\pi f}{2} (\cos^2 \frac{\pi f}{4} \cos^2 \alpha + \sin^2 \frac{\pi f}{4} \sin^2 \alpha)$$

this expression will be invariant to the phase shift α when:

$$\cos^2 \frac{\pi f}{4} = \sin^2 \frac{\pi f}{4}$$

that is for f = 2r+1; r = 0,1,2,3; f = odd integer frequency. In the case of a sequency of 2 the power density is invariant to the phase shift angle when:

$$\cos^2 \frac{\pi f}{8} = \sin^2 \frac{\pi f}{8} \ ;$$

that is for f = 2+4r, r = 0,1,2,3; f = even integer frequency.

In general the sequency power spectrum of sinusoides is invariant to phase shift angle for integer frequencies (integer relative to the period of definition of the Walsh functions). For fractional frequencies the coefficients of the Walsh functions of the same sequency will not be equal and the power density at a sequency of m and an arbitrary phase angle ϕ will be

$$S^2(\omega,m) \sin^2 \phi + C^2(\omega,m)\cos^2\phi$$

$$= \tfrac{1}{2}[S^2(\omega,m) + C^2(\omega,m)] + \tfrac{1}{2}[C^2(\omega,m) - S^2(\omega,m)] \cos 2\phi \qquad ....(5)$$

Referring to Figure B-1 (where T is the period of the Walsh functions and θ is the angle due to the fractional part of the sinusoidal frequency) it can be seen that for an arbitrary starting angle α the ρth power density will be equal to

$$\tfrac{1}{2}[S^2(\omega,m) + C^2(\omega,m)] + \tfrac{1}{2}[C^2(\omega,m) - S^2(\omega,m)]\cos2(\alpha + (\rho-1)\theta)$$

$$\dots\dots (6)$$

and the sum of the ρ power densities will be

$$\tfrac{\rho}{2}[S^2(\omega,m) + C^2(\omega,m)] + \tfrac{1}{2}[C^2(\omega,m) - S^2(\omega,m)].\sum_{r=1}^{\rho}\cos 2(\alpha + (r-1)\theta)$$

$$\dots\dots (7)$$
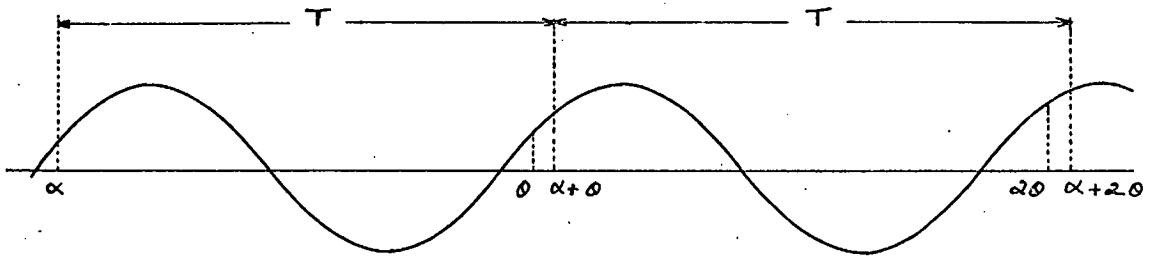
B-1

now $\displaystyle\sum_{r=1}^{\rho}\cos(2\alpha + 2(r-1)\theta) = \text{Re.}\sum_{r=1}^{\rho}e^{j2(\alpha + (r-1)\theta)}$ $\dots\dots (8)$

$$= \text{Re.}\,e^{j2\alpha}(1 + e^{j2\theta} + e^{j4\theta} + e^{j6\theta} + \dots e^{j(2\rho-2)\theta}$$

$$= \text{Re.}\,e^{j(2\alpha + (\rho-1)\theta)}.\frac{\sin\rho\theta}{\sin\theta}$$

therefore $\displaystyle\sum_{r=1}^{\rho}\cos(2\alpha + 2(r-1)\theta) = \text{Re }e^{j(2\alpha + (\rho-1)\theta)}\frac{\sin\rho\theta}{\sin\theta}$

$$= \cos(2\alpha + (\rho-1)\theta)\frac{\sin\rho\theta}{\sin\theta}$$

Substituting the above into equation (7) will give:-

$$\begin{pmatrix}\text{average power density} \\ \text{at sequency m}\end{pmatrix}. \rho = \frac{\rho}{2}(S^2(\omega,m) + C^2(\omega,m) + \frac{1}{2}(C^2(\omega,m)$$

$$- S^2(\omega,m)).\cos(2\alpha+(\rho-1)\theta)\ \frac{\sin\rho\theta}{\sin\theta} \qquad \dots\ (11)$$

for frequencies where $f = $ integer $+ \frac{1}{4}, \frac{1}{2}$ or $\frac{3}{4}$ cycles θ will be $\frac{\pi}{2}$, π or $\frac{3\pi}{2}$ respectively. For an even number ρ the phase dependent term in equation (11) will equal zero in the case of $\theta = \frac{\pi}{2}, \frac{3\pi}{2}$ since $\frac{\sin\rho\theta}{\sin\theta} = 0/1$, while for the case $\theta = \pi$ the phase dependent part of equation (11) will be multiplied by $\rho \cos2\alpha$, since $\frac{\sin\rho\theta}{\sin\theta} = \frac{0}{0}$.

APPENDIX C

Absolute Value of Walsh Coefficients
for Various Sinusoidal Frequencies of One
Volt Amplitude

Walsh Coefficient for	frequency cycles/second							
	f=1	f=2	f=5	f=6	f=11	f=12	f=15	f=16
sal 1	·637	0	·1275	0	·0583	0	·0435	0
cal 1	·637	0	·1275	0	·0583	0	·0435	0
sal 2	0	·637	0	·2122	0	0	0	0
cal 2	0	·637	0	·2122	0	0	0	0
sal 3	·264	0	·308	0	·141	0	·018	0
cal 3	·264	0	·308	0	·141	0	·018	0
sal 4	0	0	0	0	0	·21	0	0
cal 4	0	0	0	0	0	·21	0	0
sal 5	·0526	0	·459	0	·2104	0	·0036	0
cal 5	·0526	0	·459	0	·2104	0	·0036	0
sal 6	0	·264	0	·5115	0	0	0	0
cal 6	0	·264	0	·5115	0	0	0	0
sal 7	·1268	0	·191	0	·0866	0	·0084	0
cal 7	·1268	0	·191	0	·0866	0	·0084	0
sal 8	0	0	0	0	0	0	0	0
cal 8	0	0	0	0	0	0	0	0

Sal 9	·012	0	·102	0	·1623	0	·0857	0
cal 9	·012	0	·102	0	·1623	0	·0857	0
sal 10	0	·0526	0	·3415	0	0	0	0
cal 10	0	·0526	0	·3415	0	0	0	0
sal 11	·005	0	·244	0	·3912	0	·0355	0
cal 11	·005	0	·244	0	·3912	0	·0355	0
sal 12	0	0	0	0	0	·5115	0	0
cal 12	0	0	0	0	0	·5115	0	0
sal 13	·026	0	·163	0	·2614	0	·1784	0
cal 13	·026	0	·163	0	·2614	0	·1784	0
sal 14	0	·1268	0	·1414	0	0	0	0
cal 14	0	·1268	0	·1414	0	0	0	0
sal 15	·063	0	·07	0	·108	0	·4307	0
cal 15	·063	0	·07	0	·108	0	·4307	0
sal 16	0	0	0	0	0	0	0	·637
cal 16	0	0	0	0	0	0	0	·637
sal 17	·003	0	·017	0	·0651	0	·3975	0
cal 17	·003	0	·017	0	·0651	0	·3975	0
sal 18	0	·012	0	·0424	0	0	0	0
cal 18	0	·012	0	·0424	0	0	0	0
sal 19	·0013	0	·0415	0	·1571	0	·1621	0
cal 19	·0013	0	·0415	0	·1571	0	·1621	0

sal 20	O	O	O	O	O	·3415	O	O
cal 20	O	O	O	O	O	·3415	O	O
sal 21	·0002	O	·062	O	·2346	O	·0322	O
cal 21	·0002	O	·062	O	·2346	O	·0322	O
sal 22	O	·005	O	·1035	O	O	O	O
cal 22	O	·005	O	·1035	O	O	O	O
sal 23	·0006	O	·0255	O	·0972	O	·0778	O
cal 23	·0006	O	·0255	O	·0972	O	·0778	O
sal 24	O	O	O	O	O	O	O	O
cal 24	O	O	O	O	O	O	O	O
sal 25	·0062	O	·0935	O	·0522	O	·0116	O
cal 25	·0062	O	·0935	O	·0522	O	·0116	O

REFERENCES

(1) Ahmed, N., K.R. Rao,   "Bifore Hadamard Transform"
     A.L. Abdusatar   I.E.E.E. Transactions
                               Audio and Electro-acoustics
                               AU-19; No. 3; Sept. 1971.

(2) Bobwetter, C.   Analog Sequency Analysis and Synthesis
                               of Voice Signals"
                               Proceedings 1970 Symposium
                               Applications of Walsh Functions
                               Catholic University of America,
                               Washington D.C.

(3) Gibbs, J.E.   "Some Properties of Functions on the
                               Non-Negative Integers Less than 2^n"
                               National Physical Laboratory
                               D.E.S. Report No. 3, December, 1969.

(4) Gibbs, J.E.,   "Walsh Functions as Solutions of a
     Millard, M.J.   Logical Differential Equation"
                               National Physical Laboratory D.E.S. Report
                               No. 1, 1969.

(5) Gethoffer, H.   "Speech Processing with Walsh Functions"
                               Proceedings 1972 Symposium
                               Applications of Walsh Functions
                               Catholic University of America,
                               Washington, D.C.

(6) Harmuth, H.F.   "Transmission of Information by Orthogonal
                               Functions"
                               New York Springer Verlag, 1969.

(7)/

(7)   Harmuth, H.F.   "A Generalised Concept of Frequency and
                            Some Applications".
                            I.E.E.E. Transactions Information Theory
                            I.T.-14 Number 3.   May 1968.

(8)   Harmuth, H.F.   "Applications of Walsh Functions in
                            Communications"
                            I.E.E.E. Spectrum 1969, pp.82-91.

(9)   Jordan, J.R.,   "Correlation-Function Display and Peak
      M.S. Beck.   Detection".
                            Electronics Letters   Volume 8 No. 24,
                            November 1972.

(10) Lee, J.D.   Review of Recent Work on Applications
                            of Walsh Functions in Communications.
                            Proceedings 1970 Symposium.
                            Applications of Walsh Functions
                            Catholic University of America,
                            Washington D.C.

(11) Murray, G.G.   "Digital Walsh Filter Design"
                            Proceedings 1971 Symposium.
                            Applications of Walsh Functions,
                            Catholic University of America,
                            Washington D.C.

(12) Ohnsorg, F.R.   "Spectral Modes of the Walsh-Hadamand
                            Transform"
                            Proceedings 1971 Symposium.
                            Applications of Walsh Functions,
                            Catholic University of America,
                            Washington D.C.

(13) Pichler, F.   "Walsh Functions and Linear System
                            Theory"
                            Proceedings 1970 Symposium Applications
                            of Walsh Functions, Catholic University
                            of America, Washington D.C.

(14)/

(14) Pichler, F.   "Walsh Functions and Optimal Linear
                            Filters"
                            Proceedings 1970 Symposium,
                            Applications of Walsh Functions,
                            Catholic University of America,
                            Washington D.C.

(15) Pratt, W.K.,   "Hadamard Transform Image Coding"
     J. Kane, H.C. Andrews. Proceedings of the I.E.E.E.
                            Volume 57, No. 1. January 1969.

(16) P.D.P.-8   Maintenance Manual.

(17) P.D.P.-8   Floating Point System Programing Manual
                            8-5-S.

(18) P.A.L. III   Programing Manual.

(19) Robinson, G.S.,   "A Design Procedure for Nonrecursive
     R.L. Granger.   Digital Filters Based on Walsh Functions".
                            Proceedings 1971 Symposium,
                            Applications of Walsh Functions,
                            Catholic University of America,
                            Washington, D.C.

(20) Robinson, G.S.   "Logical Convolution and Discrete Walsh
                            and Fourier Power Spectra".
                            I.E.E.E. Transactions Audio and Electro-
                            acoustics. Volume AU-20, No. 4. Oct. 1972.

(21) Rushforth, C.D.   "Fast Fourier Hadamard Decoding of
                            Orthogonal Codes"
                            Information and Control Vol. 15, page 33-
                            37, July 1969.

(22) Stafford, E.M.,   "Control Applications of Walsh Functions"
     T.S. Durrani.   Proceedings 1971 Symposium Theory and
                            Application of Walsh Function. Hatfield
                            Polytechnic, June 1971.

(23)/

(23) Shanks, J.L.   "Computation of the Fast Walsh Transform"
                           I.E.E.E. Transactions Computers.
                           Volume C-18 pp. 457-459. May 1969.

(24) Sheingold, D.H.   "Analog-Digital Conversion Handbook"
                           Analog Devices Inc., Massachusetts.

(25) Signetics T.T.L.   Data Book.

(26) Signetics M.O.S.   Static Shift Registers Data Book.

(27) Townsend, H.R.A.   "The Value of E.E.G. Frequency Analysis
                           in Hepatic Encephalopathy"
                           Journal of the Royal College of Surgeons
                           of Edinburgh. Vol. 15, pp. 151-157,
                           May 1970.

(28) Walsh, J.L.   "Remarks on the History of Orthogonal
                           Functions"
                           Proceedings 1970 Symposium,
                           Application of Walsh Functions,
                           Catholic University of America,
                           Washington D.C.