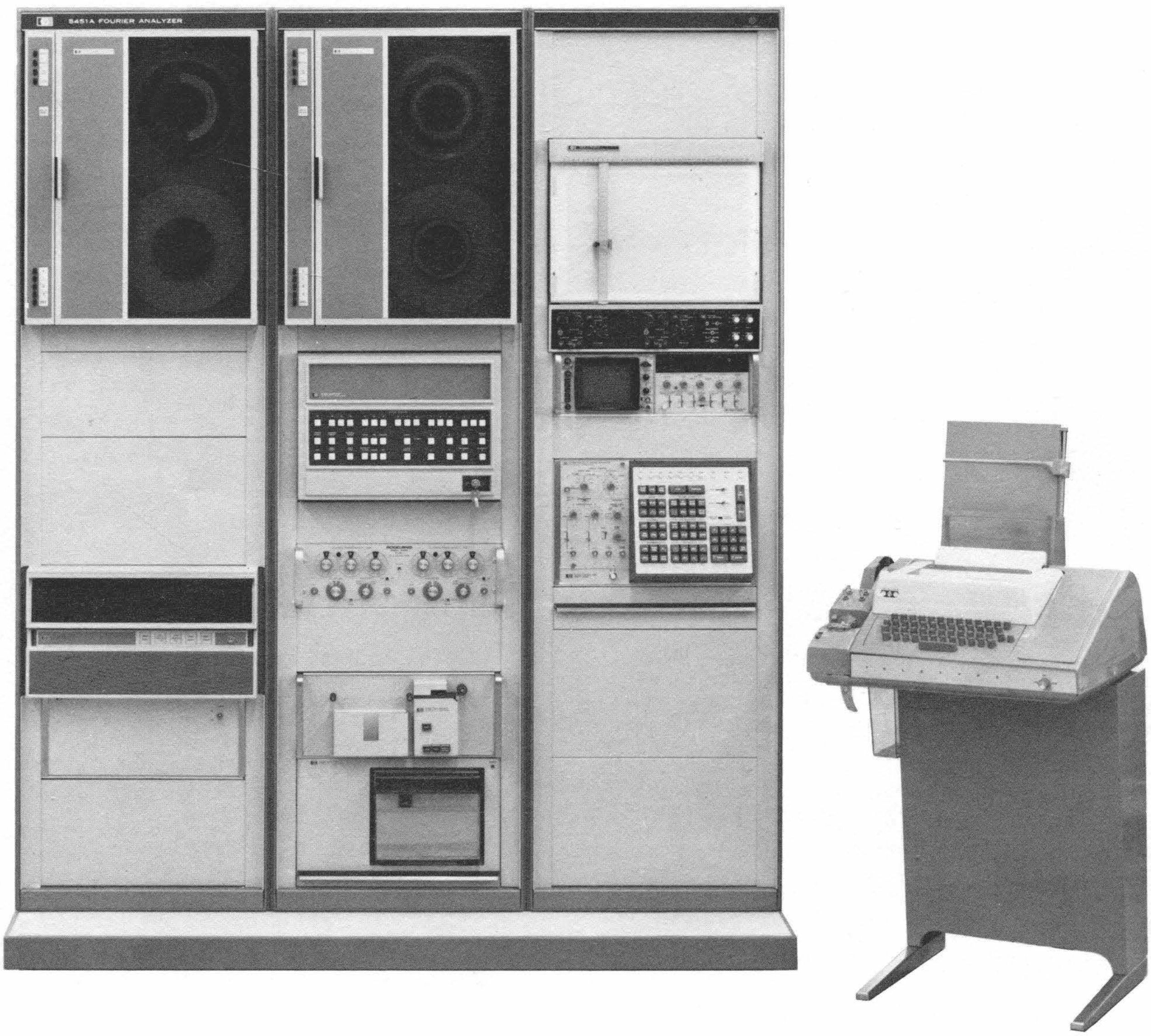
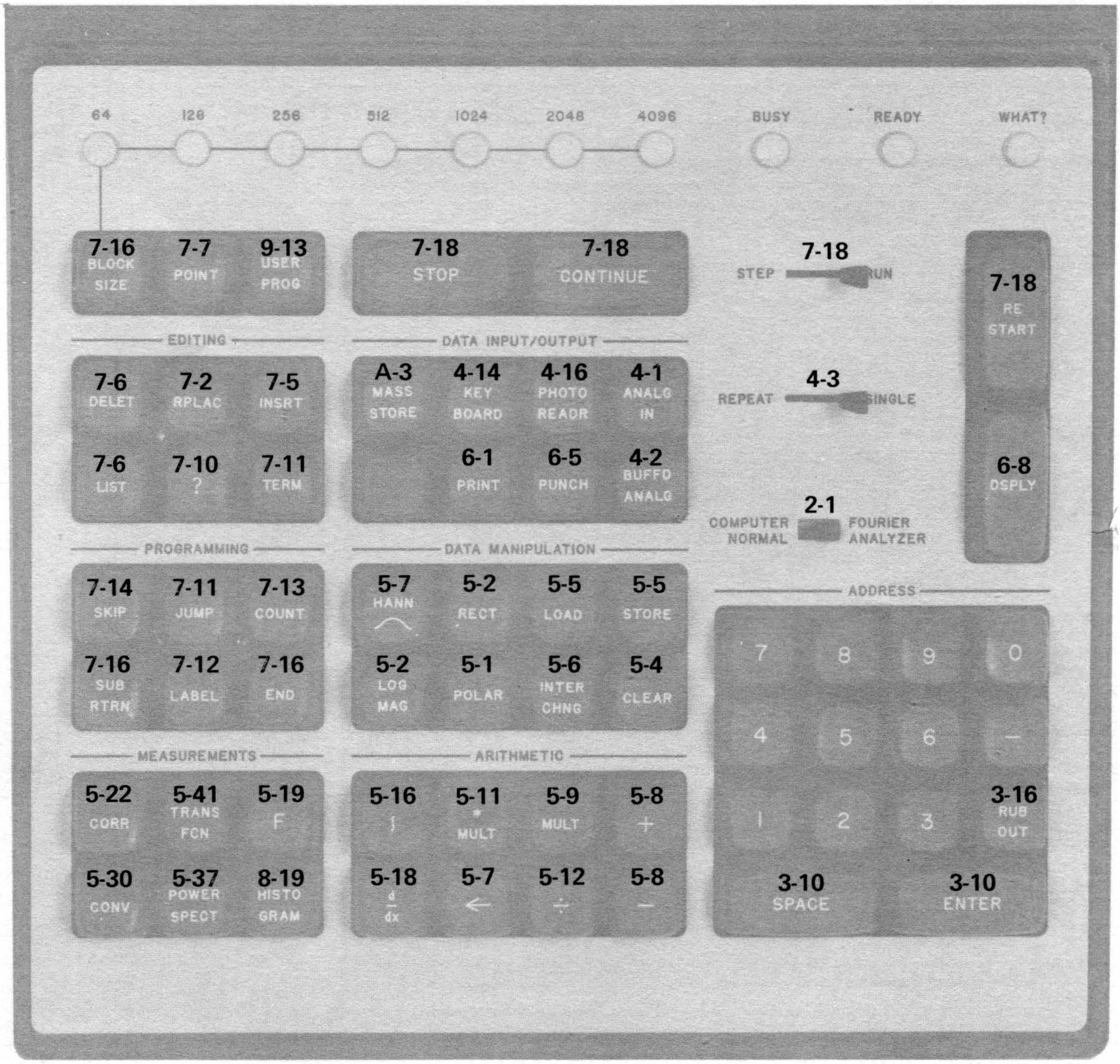
FOURIER ANALYZER SYSTEM

5451A



SYSTEM OPERATING MANUAL

HEWLETT PACKARD



QUICK-KEY REFERENCE

FOURIER ANALYZER SYSTEM

5451A

Printed: APRIL 1972

HOW TO USE THIS MANUAL

If you have just received your Fourier Analyzer, Section I provides background information on your system. If you want to operate the system, turn first to Section II, which gives complete instructions on turning on the instrument and loading the Fourier tape.

Then proceed straight through Section III. This is a self-teaching demonstration of some of the principle keys and input/output functions of the instrument.

After that, you can set up your own experiments based on the information in Sections IV through VIII. To add your own programs to the system, see Section IX on how to write "Y commands". Note the Table of Contents, the marginal references, and the Index in the back of the book as aids in finding specific items in the manual.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

TABLE OF CONTENTS (Cont'd)

TABLE OF CONTENTS (Cont'd)

LIST OF FIGURES

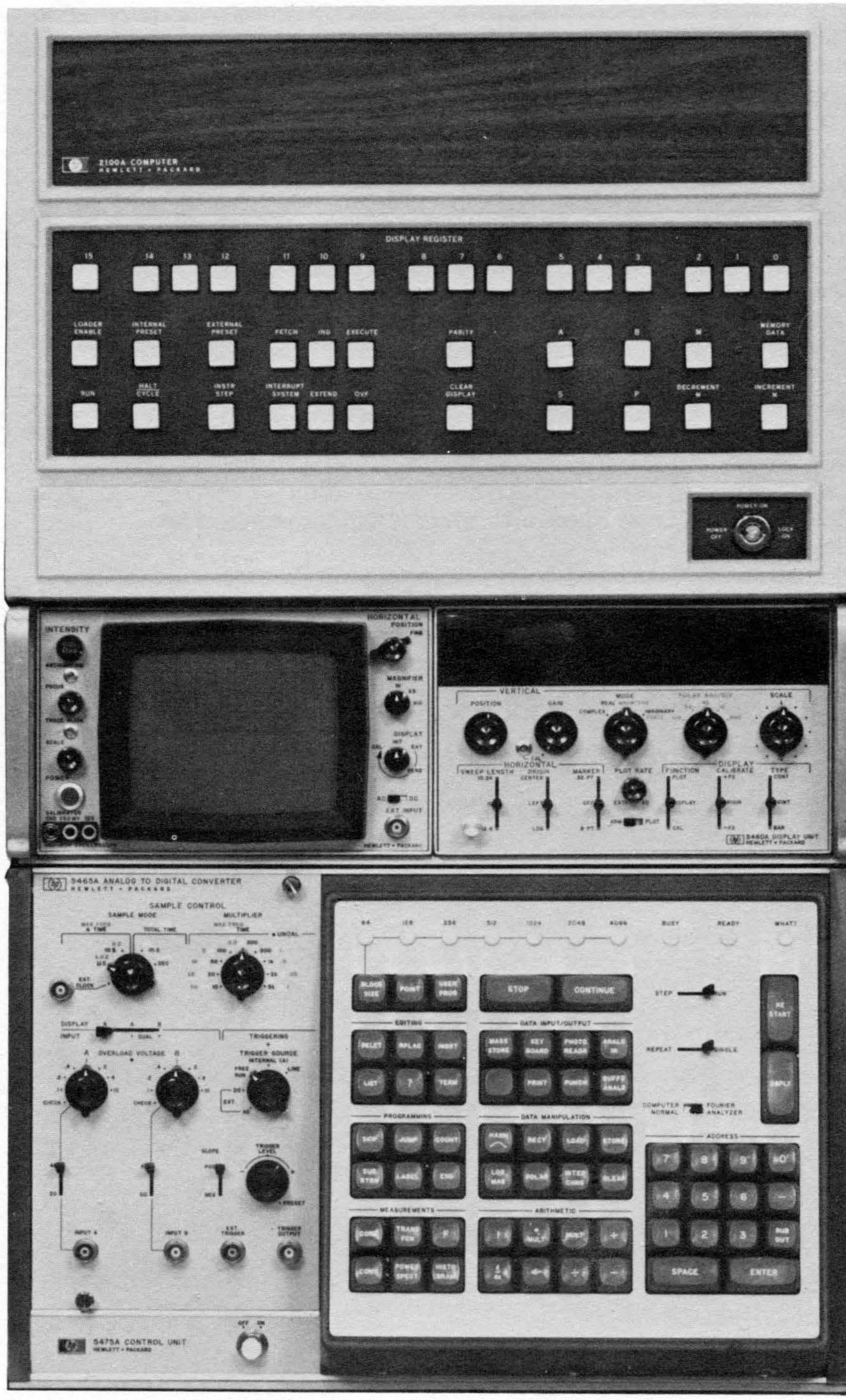
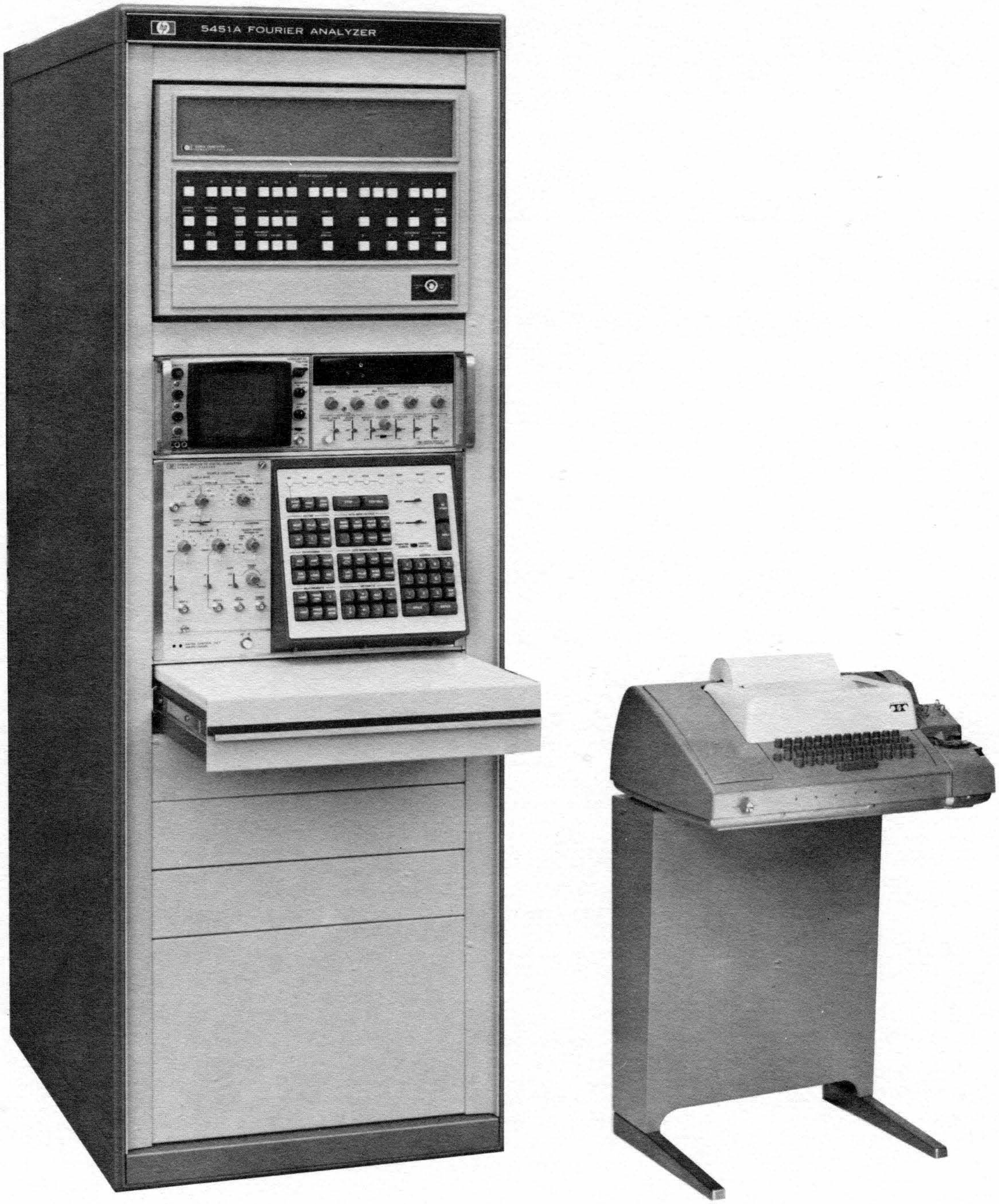

2100A COMPUTER
HEWLETT • PACKARD
DISPLAY REGISTER
INTENSITY
HORIZONTAL
VERTICAL
POSITION
SCALE
5460A DISPLAY UNIT
5465A ANALOG TO DIGITAL CONVERTER
SAMPLE CONTROL
STOP
CONTINUE
SPACE
ENTER
5475A CONTROL UNIT

Figure 1-1. Model 5451A Fourier Analyzer System and Teleprinter



SECTION I

GENERAL INFORMATION

SYSTEM DESCRIPTION

The HP 5451A Fourier Analyzer System performs analyses of time and frequency data containing frequencies from dc to 25 kHz. The system analyzes time-series data such as mechanical vibrations, sonar echoes, tidal action, biomedical phenomena such as brain waves and nerve impulses, voltages and currents in electronic systems, and acoustic phenomena such as speech sounds. These analyses may detect signals hidden in noise, as in sonar, or may locate critical frequencies in transfer functions, and their corresponding coherence functions in complex systems. Both continuous and transient data may be processed.

Keyboard programming allows the user to perform the following operations automatically, without special software:

- Forward and inverse Fourier transform
- Magnitude and phase spectrum
- Power and cross power spectrum
- Transfer function
- Coherence function
- Convolution (digital filtering)
- Auto and cross correlation
- Hanning and other weighting functions
- Histogram
- Scaling
- Cepstrum
- Ensemble averaging (time and frequency)
- User-written subroutines

Six editing keys operate an on-line resident editor so that a sequence of steps configured into an automatic measurement procedure may be changed on-line without the need to do off-line editing, compiling and testing. In fact the series of steps or program used to perform a particular operation can be stored on punched paper tape for easy re-entry into the Fourier Analyzer.

Data input and output is likewise controlled from the Keyboard. Data can be entered in analog form through the 10-bit 2-channel Analog-to-Digital Converter, or in digital form on punched paper tape, or manually from the Keyboard, or directly from a remote computer.

Results of all operations are displayed on the oscilloscope. In addition, results can be printed out in decimal numbers on the Teleprinter, punched on paper tape, sent to a remote computer, displayed on a separate large-screen oscilloscope, or plotted on an external X-Y plotter.

System Identification

The Fourier Analyzer is a completely calibrated system; all displays and data outputs are accompanied by a scale factor relating them to physical units. This calibration results from digital techniques being used in all computations. The 5451A uses the HP 2100A Computer with 8192-word (8K) memory; optionally, the Computer can be purchased with 12K, 24K, or 32K memory. The 2100A can also be used as a stand-alone computer. Computer programming knowledge is not required for operation of the Fourier Analyzer; all operations are controlled through the Keyboard. However, provisions are made to incorporate user-written subroutines into the Fourier program.

SYSTEM CONFIGURATION

A Fourier Analyzer System consists of a basic system plus a number of customer-chosen options. Since the resulting configuration will often be unique, the list of component units is not provided in this manual, but in a separate System Configuration Notice, supplied with the system. This document also tells what Computer interface channels are used by the various units (ADC, Teleprinter, etc.).

IDENTIFICATION NUMBERS

Model Number and Name

Each unit in the standard Fourier Analyzer System is identified by model or specification number and name as a separate instrument; these are:

Model 180AR/DR Oscilloscope (with Option H51)
Model 2100A Computer
Model 2752A Teleprinter
Model 5460A Display Plug-in Unit
Model 5465A ADC Plug-in Unit
Model 5475A Control Unit

Serial Numbers

Each Fourier Analyzer System is identified by a 2-section system serial number (5451A-000). The number is on a stick-on plate mounted on the inside rear of system cabinet. The 3-digit number is a serial number unique to each Fourier Analyzer System.

In addition, each plug-in unit in the Fourier Analyzer System, including the system Computer is identified by a serial number (0000A00000). The first section is a serial prefix number, used to document changes to the unit; the second portion of the serial number is a number unique to each instrument (of that model number).

Include complete model name, model number, and serial number of any unit or units in all correspondence about your system.

CRT WARRANTY

The Fourier Analyzer and each of its individual units are certified and warranted as stated on inside front cover of this manual. The CRT (Cathode Ray

Tube) however, is covered by a warranty separate from the rest of the system. The CRT warranty and warranty claim form are located in the Oscilloscope manual. Should CRT fail within time specified on warranty, return CRT with warranty form completed.

STORAGE AND SHIPMENT

Packaging

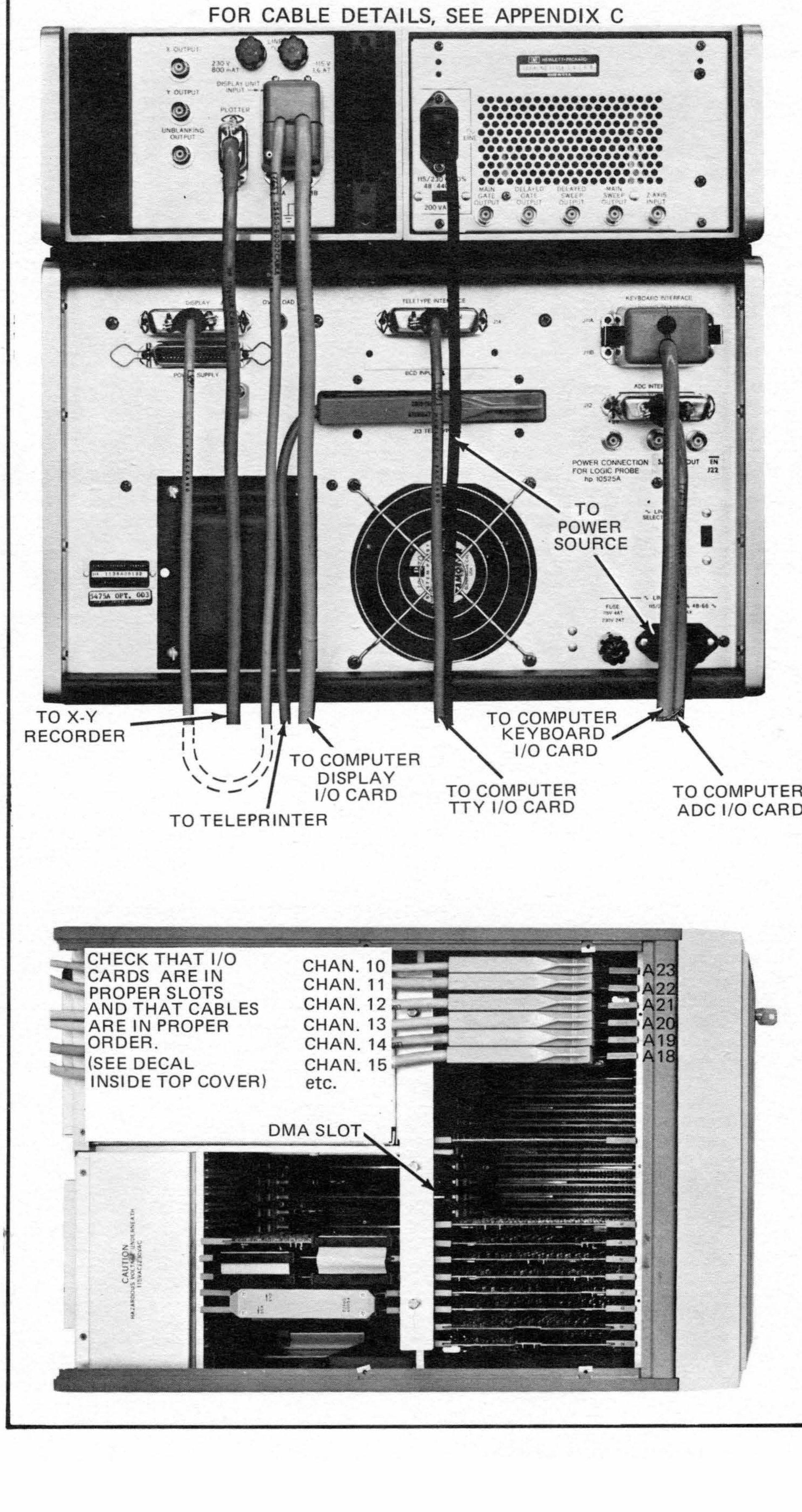
To protect valuable electronic equipment during storage or shipment, always use the best packaging methods available. Your Hewlett-Packard Sales and Service Office can provide packaging material such as that used for original factory packaging. Contract packaging companies in many cities can provide dependable packaging on short notice.

SCOPE OF THIS MANUAL

Operating Information

This manual contains only the information required to operate the Fourier Analyzer. For service or troubleshooting information, see the 5451A System Service Manual. For information on software programming, see the 5451A System Software Manual.

Figure 2-1. 5451A System Ready for Turn-On



SECTION II

TURN-ON/TURN-OFF PROCEDURE

ABOUT THIS SECTION

Power turn-on

This section provides instructions for initial and routine turn-on/turn-off of the Fourier Analyzer System. It includes instructions on loading the master Fourier Program Tape, and reloading the Basic Binary Loader.

PREPARATION

The following assumes that the Fourier Analyzer System has been installed and checked out. To verify proper system cable interconnection, refer to the table in Appendix C or refer to the 5451A System Service Manual. *In the case of rack-mounted systems,* the Teleprinter power cord should be plugged into a socket in the strip on the left side (viewed from rear) of the system cabinet. (Plugging the Teleprinter into this strip ensures that it will be turned off when the system power is turned off; if the Teleprinter is plugged directly into a 115V ac wall socket, it will continue to run when the main system power is turned off.) The Teleprinter data line should be plugged into the Control Unit (J13) as shown in Figure 2-1. Plug the Control Unit, Oscilloscope Mainframe, and Computer power cords into the cabinet strip. And finally, plug the cabinet main power line into a grounded 115V ac source.

For systems that are not rack-mounted, plug the Teleprinter power cord into an ac socket. It is preferable though not required to use a multi-socket box with a switch to plug all the power cords into, so that power can be turned on and off for the entire system by a single switch. The Teleprinter data line should be plugged into the back of the Control Unit (J13) as shown in Figure 2-1. Then plug the Control Unit power cord, Oscilloscope Mainframe power cord, and Computer power cord into 115V ac sockets.

TURN-ON

The following can be considered a first-time turn-on procedure. Thereafter leave all subsystem power switches on, if your system is rack-mounted, and simply use the cabinet power switch to turn the Fourier Analyzer on and off.

1. Set Keyboard switch to FOURIER ANALYZER.

2. Turn on cabinet power.

   For systems with rack-mounting: Press button or turn on switch in upper right corner of cabinet.

   For systems without rack-mounting: There is no master power switch. Turning on each of the subsystems as described below, turns on the system.

3. Turn on Computer power by rotating the switch key to POWER ON. (Turning to LOCK ON position disables front panel pushbuttons.)

4. Turn on Oscilloscope mainframe power by pressing green pushbutton, if it is not lighted.

5. Turn on Control Unit power by pressing white OFF/ON pushbutton, if it is not lighted. Energizing the Control Unit mainframe energizes the Key-

board and the ADC plug-in. When the Keyboard is energized, either the BUSY or READY lamp will be lighted.

a. If the READY lamp is lighted, the system is operating properly in the Fourier mode.

b. If the BUSY lamp is lighted, press RESTART.

Loading tape

If the READY lamp does not light, or the BUSY lamp goes out, all or part of the Master Fourier Program has probably been erased. Load the Fourier Program as described below under "Loading Fourier Program Tape".

6. Auxiliary equipment (Teleprinter, Tape Reader; Tape Punch, etc.) should be turned off when not being used to prevent unnecessary wear on the mechanical parts in these devices.

When the system is turned on, the Computer automatically restarts the Fourier program. If the Fourier Analyzer is in the COMPUTER NORMAL mode only the Keyboard is disabled, the Fourier Analyzer can still be controlled from the Teleprinter.

LOADING FOURIER PROGRAM TAPE

The Fourier Analyzer is shipped from the factory with the Fourier Program Tape loaded into the Computer memory; it should be necessary only to follow the turn-on procedure given above to turn the system on. However, if the Computer has been used as a general-purpose digital computer or if some accident has destroyed the Fourier program in memory, it will be necessary to reload the Fourier Program Tape, using the procedure given below. If you are not sure whether the complete and correct program is in memory, load the tape. Reloading will not modify data that was previously stored in memory by the Fourier program.

NOTE

Some systems may come with two or more program tapes: one for the standard configuration plus others for the system configured with any additional options that might have been purchased. Be sure to read the System Configuration Notice for the options to identify the correct Fourier tape to be loaded.

Procedure for System with 2748A Photoreader (High Speed Tape Reader)

1. Set Keyboard switch to FOURIER ANALYZER enabling Keyboard control.

2. a. Place Fourier tape roll in Photoreader tape holder, feed holes toward rear.

   b. Press POWER pushbutton.

   c. Press LOAD pushbutton.

   d. Run the tape leader underneath the wire-guide and through the pair of feed rollers.

   e. Press READ pushbutton.

3. a. Press Computer HALT button if not in HALT mode. Press to light P button and enter starting address on DISPLAY REGISTER buttons (017700 for 8K memory; 027700 for 12K; 037700 for 16K; 057700 for 24K; 077700 for 32K). All starting addresses are octal numbers.

b. Press INTERNAL PRESET, EXTERNAL PRESET, and LOADER ENABLE.

c. Press RUN button. Tape should run through Photoreader and stop.

d. After the tape has run through, press LOAD (or MANUAL ADVANCE) pushbutton on Photoreader and remove tape. Press POWER switch to off. Rewind tape and return to its box.

Loading tape

e. Press Computer P button (if not lighted) and enter starting address of 000002_8 on DISPLAY REGISTER (i.e. pushbutton 1 lighted only).

f. Press Computer INTERNAL PRESET, EXTERNAL PRESET, and RUN. Program is now in operation.

If tape will not run through:

Check the following:

Are tape feed holes toward instrument front panel?

Are you starting in a blank section of tape (except for feed holes) so that no data is lost?

Is tape running under wire guide in Photoreader?

If you still have problems loading the Fourier Program Tape refer ahead to the more comprehensive "Loading Problems."

Procedure for System using Teleprinter Reader

1. Set Keyboard switch to FOURIER ANALYZER, enabling Keyboard control.

a. Place Teleprinter switch on right side to LINE and switch on left side, near reader, to STOP.

b. Load tape in Teleprinter reader, lowering plastic cover in place.

3. a. Press Computer HALT button if not in HALT mode. Press to light P button and enter starting address on DISPLAY REGISTER buttons (017700 for 8K memory; 027700 for 12K; 037700 for 16K; 057700 for 24K; 077700 for 32K). All starting addresses are octal numbers.

b. Press INTERNAL PRESET, EXTERNAL PRESET, and LOADER ENABLE.

c. Press RUN button. Place switch on Teleprinter reader to START. Tape should read through Teleprinter and stop.

d. Place switch on Teleprinter reader to STOP and switch on right to OFF. Rewind tape.

e. Press Computer P button (if not lighted) and enter starting address of 000002_8 on DISPLAY REGISTER (i.e., pushbutton 1 lighted only).

f. Press Computer INTERNAL PRESET, EXTERNAL PRESET, and RUN. Program is now in operation.

LOADING PROBLEMS

Loading problems

Sometimes operator error or dirty equipment can prevent a tape from being loaded properly. If you are having trouble loading tape in the Photoreader or Teleprinter reader, consider the possibility of having one or more of the problems outlined in the steps below. If the tape still doesn't run, then the Basic Binary Loader — the program which loads tape into the machine—has probably been erased. To reload, see paragraph ahead "Reloading the Basic Binary Loader".

1. Tape not correctly placed in reader.

   a. The row of small feed holes (except for bootstrap loader leader) is offset on the tape; when the tape is placed in the reader, these holes should be nearest the Photoreader front panel, instead of toward you. (More obvious on Teleprinter reader.)

   b. Tape placed too far into reader. Unless otherwise instructed, all data holes should be to left of reading element before tape begins its run through reader. (To rear of element on Teleprinter.)

   c. Tape loaded backward in reader. All tapes supplied with your system are identified by a label at the beginning of the tape. When rewinding the tape, be sure this leader is at the outside of the roll.

2. Tape torn or dirty. Tape that is faulty, or not clean, may be read incorrectly.

3. Dirty Photoreader or Teleprinter reader. To clean:

   a. Try blowing dirt away with your breath.

   b. Remove tape and clean reader using alcohol and brush supplied with your system. Brush feeder holes clean; also lightly brush bottom of reader lens on Photoreader. Blow dirt away with your breath.

4. For more serious problems, refer to the unit service manual.

RELOADING BASIC BINARY LOADER

The Basic Binary Loader (BBL) is the program that loads any absolute binary tape into the Computer memory. It occupies the last 64 words in the 8K or 16K word memories. The program is protected by the Computer front panel LOADER ENABLE button; the only way BBL can be erased is by operating with the LOADER ENABLE button lighted.

Once the BBL has been erased for whatever reason, it must be reloaded using the "boot-strap" (short length of tape) supplied. Systems having accessories such as the magnetic tape or disc unit come with two bootstraps: one for the system configured for the accessories, and one for a standard system without accessories. In any case, use the appropriate bootstrap for your system.

To reload the BBL:

1. Press Computer HALT button if not lighted.

2. Press M button. Enter 000020_8 on DISPLAY REGISTER (button 4 lighted only). Press MEMORY DATA to enter this address code into the memory address register.

3. Enter the following octal numbers into the DISPLAY REGISTER, pressing INCREMENT M after each number except the last. Proceed with column 1, top to bottom, then column 2, and so on.

1037CC*	1025CC*	1023CC*	170001
1023CC*	001727	024026	006004
024021	1037CC*	1024CC*	024020

*The two letters CC represent the tape input device channel. If your machine has a Photoreader, then CC = 10; if it has no Photoreader, then CC = 11 for Teleprinter input. For non-standard systems, such as those with magnetic tape or disc units, check the System Configuration Notice for the correct tape input device channel number.

5. After entering final number 024020 above, press Computer B button and enter starting address on DISPLAY REGISTER (from step 3a on page 2-2 or 2-3).
6. Press P button and enter 000020_8 into the DISPLAY REGISTER. Press INTERNAL PRESET and EXTERNAL PRESET.
7. Place the bootstrap tape P/N 05451-90007 in the Photoreader (if your system has one) or otherwise in the Teleprinter reader the same way as described for the Fourier tape. Press POWER button to on.
8. Press Computer LOADER ENABLE button. Press RUN button.
9. The tape should run through. Press HALT button. Press LOADER ENABLE to off.

NOTE

For systems with other than 8K memory, the following procedure must be used to modify a location in the BBL.

a. Press Computer LOADER ENABLE button. Press M button.

b. Enter address $0m7772_8$ on DISPLAY REGISTER buttons.
where: m = 1 for 8K memory
m = 2 for 12K memory
m = 3 for 16K memory
m = 5 for 24K memory
m = 7 for 32K memory

c. Press MEMORY DATA to enter above address.

d. Enter $1n0100_8$ on DISPLAY REGISTER buttons.

where: n = 6 for 8K memory
n = 5 for 12K memory
n = 4 for 16K memory
n = 2 for 24K memory
n = 0 for 32K memory

e. Press INCREMENT M button. Press LOADER ENABLE to off.

SECTION III

LEARNING TO OPERATE THE FOURIER ANALYZER

ABOUT THIS SECTION

This section is intended to be a self-teaching guide for the new owner of the Fourier Analyzer. It is a demonstration of some of the basic Keyboard functions with photos of how the corresponding scope displays should look. The demonstration also explains the use of the rest of this manual, which is set up in reference form so that the user can quickly find the information he needs for his specific purposes.

Data storage

In order not to interrupt the procedure once it has begun, background information on three important subjects is being placed first: These are (1) data memory format, (2) ADC sampling parameters, and (3) the Keyboard command structure. These should be read before beginning the demonstration.

HOW DATA IS STORED IN THE ANALYZER MEMORY

The Data Word

The smallest element of data that may be entered or taken out of the Fourier Analyzer memory for an input/output processing operation is the data word. Each data word represents a value at a given point in time of a time series, or of a correlation function, or a value of a spectrum at a given frequency. The data word is a 16-bit binary number representing an integer from -32,768 to +32,767, a range of greater than 96 dB. Groups of these data words are collected together to form the next element of data storage, the data block, as shown in Figure 3-1.

The Data Block

The data block may contain a quantity of data words equal to any power of 2 from $2^6 = 64$ to $2^{12} = 4096$ words; minimum/maximum data words are a function of computer memory size and configuration. A data block will contain a set of time samples of a time series or a set of frequency components of a spectrum. This time record or spectrum may represent one element in an ensemble average or it may be the result of some complex operation such as a power spectrum or correlation function.

Associated with each data block are "data qualifiers", consisting of an amplitude scale factor, a block calibrator, and a frequency code. The amplitude scale factor is a power of 10 which multiplies all N data words in the block. The block calibrator is a 16-bit binary number which multiplies all N data words in the block. Its magnitude is always greater than 0.1 but ≤ 1. The data word, block calibrator, and scale factor may thus represent any number from $+1 \times 10^{-512}$ to $+1 \times 10^{+511}$. This numerical system is called "floating point on a block basis", and allows full use of all 16 bits during calculations without overflow. When an overflow occurs in an operation all words in the data block are divided by 2 (shifted right one bit) and the block calibrator and scale factor are compensated accordingly. In this way, full accuracy of the data word is attained and no loss of calibration occurs during calculations. The frequency code defines the sampling parameters used by the Analog-to-Digital Converter (ADC), if data was taken in via the ADC mode. The table of frequency codes is provided at appropriate places throughout this manual, for example, page 4-19.

When the data in a data block represents the values associated with a spectrum or other function of frequency, the data is stored differently than when the data block contains a time series. In the Fourier Analyzer, from an N-point real time series, we compute and display N/2 positive frequencies, plus dc. Each frequency except the highest has 2 independent numerical values: a real (cosine) and an imaginary (sine). The highest frequency has only a real, but no imaginary value. The actual arithmetic is as follows: N/2 positive real frequency values plus dc value = (N/2) +1 real points. The imaginary side has no dc value and no value for the highest frequency, hence it has (N/2) -1 points. Adding the real and imaginary points together we get (N/2) +1 plus (N/2) -1, or a total of N points in the frequency domain from N points in the time domain (see Figure 3-1).

Data block

The discrete finite Fast Fourier Transform (FFT) computes the same information that would be computed by the continuous finite transform. But as the name discrete FFT implies, the computer deals with discrete data words in the time or frequency domain. The FFT algorithm is arranged in such a way that the computer can calculate the time/frequency domain transformation in the shortest time. Therefore the negative frequencies that are generated by the more general double-sided transform are not computed or displayed because the negative half of the frequency (of a real-time series) is symmetrical with the positive half of the spectrum. Considerable computation time can be saved by not using double-sided FFT. The only loss is the redundant spectrum in the negative half plane.

When the data block contains a time record, the first sample point of the time record is contained in the first word (numbered 0) of the data block and the last point in the time record is found in the last word (numbered N - 1) of the data block. However, when the data block contains values in the frequency domain, the real and imaginary values are interlaced (see Figure 3-1). The only exceptions to this are the lowest real frequency (dc) and the highest real frequency (F_n). The dc value is stored in the first word (0) of the data block and the F_n value is stored in the adjacent word (1).

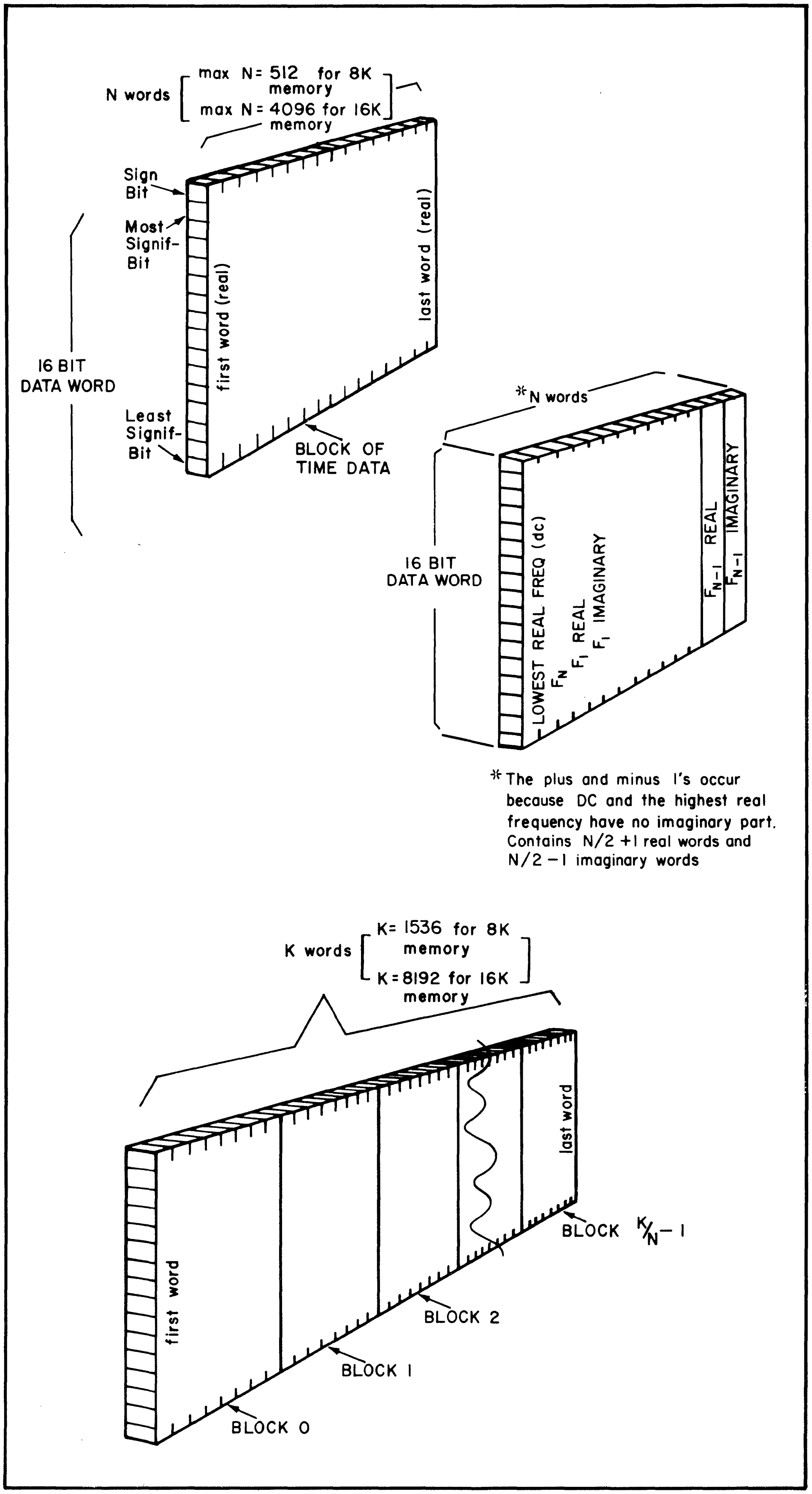
The Input Modes section and the Processing Operations section of this manual describe in detail how to direct the data word into and out of the real/magnitude or imaginary/phase part of memory. In this manual, the term channel will be used for the 1 data word defining a point in the time domain or the 2 data words defining a point in the frequency domain. Thus, there will be N channels in the time domain, and N/2 channels in the frequency domain.

A double precision data block is created under the following conditions: when a self complex conjugate multiply is performed in block 0, on data that is in the frequency domain. A double precision block is always in the frequency domain and does not have any imaginary (phase) frequency values. The least significant bits of a double precision block are stored where the imaginary frequency values were previously. Thus each double precision data word is 32 bits long.

At any particular time all data blocks must be of the same size. The number of blocks that are available for a set of operations can then be determined by dividing the total number of memory words by the block size. For example, if a block size is set to 512 words in a system with 2048 data words available, 4 data blocks numbered from 0 to 3 are available. As shown in Figure 3-1 the data blocks lie in consecutive order in the memory with the first word (0 word) of data block 0 in the first word (the 0 word) of data memory and the last word of the block in the N - 1 word of the data block. If data is contained in memory and the block size is changed, the location of a particular data word is not changed, but the boundaries of the data block are.

Data block

Figure 3-1. Data/Memory Organization



The following example will clarify this point.

Example. If the block size is 512, and a time series is represented in memory, there will be 512 numbered data positions representing an equal number of time points. Consider the 260th word of block 0 which is also numbered channel 259 of block 0. If the block size is changed to 1024, the new 0 block will include the 260th word of memory and this data point will still be channel 259 of block 0. However, if the block size had been changed from 512 to 256, the 260th word of memory would now be in the 2nd data block and would be the 4th word of data block 1. This word would be numbered channel 3 of data block 1 since the first channel in the data block is number 0.

ADC sampling

Warning: If the block size is expanded with a given set of data in memory, there may be an error in the scale factor display. The reason is as follows: suppose the block size is 128, and the block 0 scale factor is $1(10^1)$ and the block 1 scale factor is $2(10^2)$. Now the block size is doubled. Then block 0 contains data having two different scale factors. The first will dominate and be read out on the scale factor display meaning that the display will be wrong for the second half of the block. This must be kept in mind whenever block sizes are changed while there is data in more than 1 block.

However, when an original block is broken up into smaller blocks, the new blocks will take on the scale factor of the original block (unless the coded block size command is given). For instance, if block 0 is halved, the scale factor for the new block 1 will be the same scale factor as was the original block 0. (See Block Size command, page 7-16.)

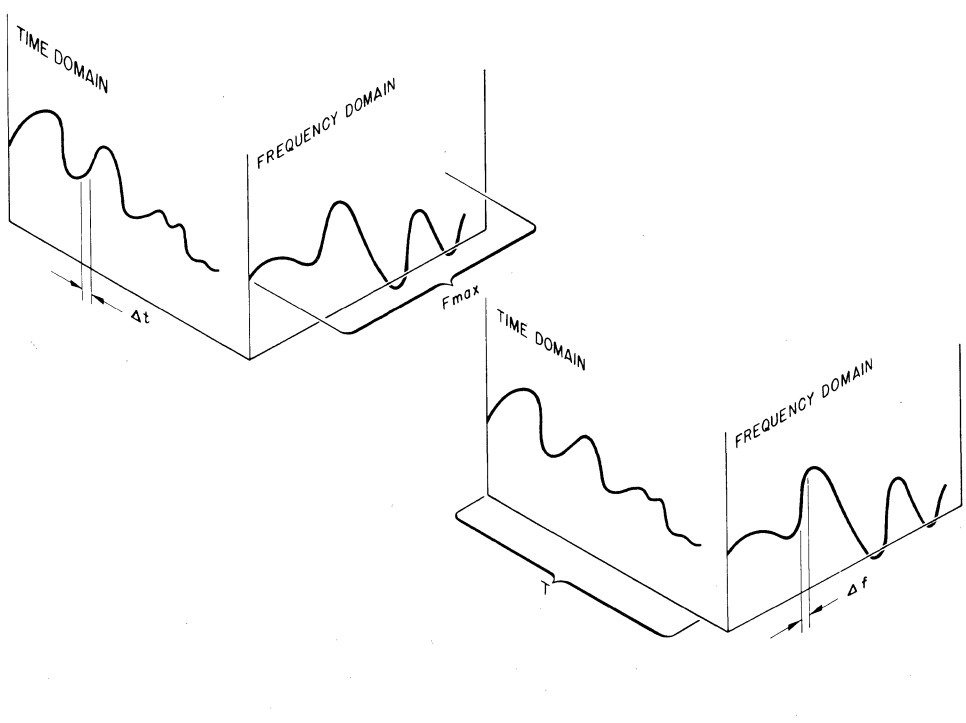
ADC SAMPLING PARAMETERS

Figure 3-2 shows a sampled time record and its corresponding spectrum. Terms which define the scale of the time and frequency records are Δt, T, F_{max}, F_s, Δf. These terms are related to each other as follows:

The sample frequency (F_s) and the sample interval (Δt) are reciprocals of each other:

$$F_s = 1/\Delta t$$

Figure 3-2. ADC Sampling Parameters



Shannon's sampling theorem requires that there be 2 samples for the highest frequency in the record, so that

$$F_{max} = \frac{F_s}{2} = \frac{1}{2\Delta t}$$

If N data points are taken (i.e., a data block size of N), with samples spaced Δt apart, a time sample record T will be recorded in memory whose length is

$$T = N\Delta t$$

ADC sampling

Since each spectrum has two values associated with each frequency (i.e., real part and imaginary part, or magnitude and phase), the display in the frequency domain will have only N/2 data points. Thus, the frequency resolution (Δf) will be determined by

$$\Delta f = \frac{F_{max}}{N/2} = \frac{F_s}{N}$$

and since

$$F_s = \frac{1}{\Delta t}$$

and

$$T = N\Delta t$$

we can see that

$$\Delta f = \frac{1}{T}$$

i.e., the frequency resolution (Δf) is determined by the sample record length (T). The four quantities Δf, F_{max}, Δf, and T completely determine the time and frequency scales of the Fourier Analyzer.

For further clarification, the parameters are broken down below according to their domain.

Time Domain Parameters

Δt — The time between samples, called the "sample interval". (Δ TIME on the ADC panel).

N — The number of samples taken: this is the data block size (BLOCK SIZE on the Keyboard).

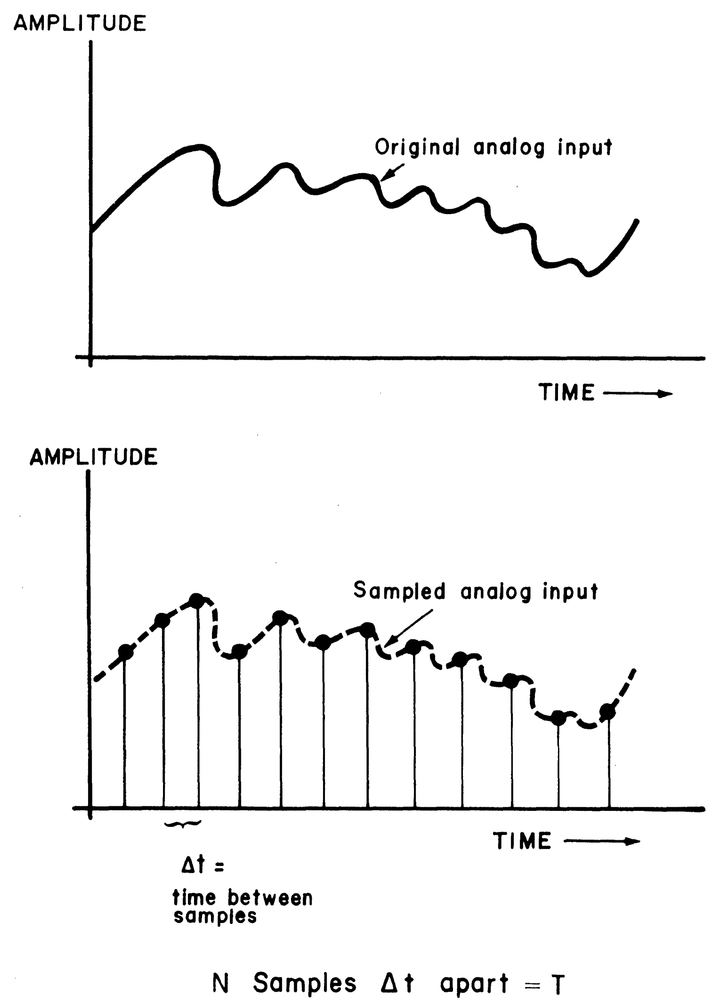
T — The total time of the sample record, called "total record length". (TOTAL TIME on the ADC panel). From Figure 3-3 it can be seen that:

ADC sampling

total record length = No. of samples x sample interval

$$T = N \times \Delta t$$

Figure 3-3. Time Domain Parameters



Frequency Domain Parameters

ADC sampling

Δf The number of Hz between frequency points, or, the frequency resolution. Origin of display is 0 Δf (dc component); next point is 1 Δf; next point is 2 Δf; next 3 Δf, etc. No finer resolution than Δf may be assigned to any frequency value.

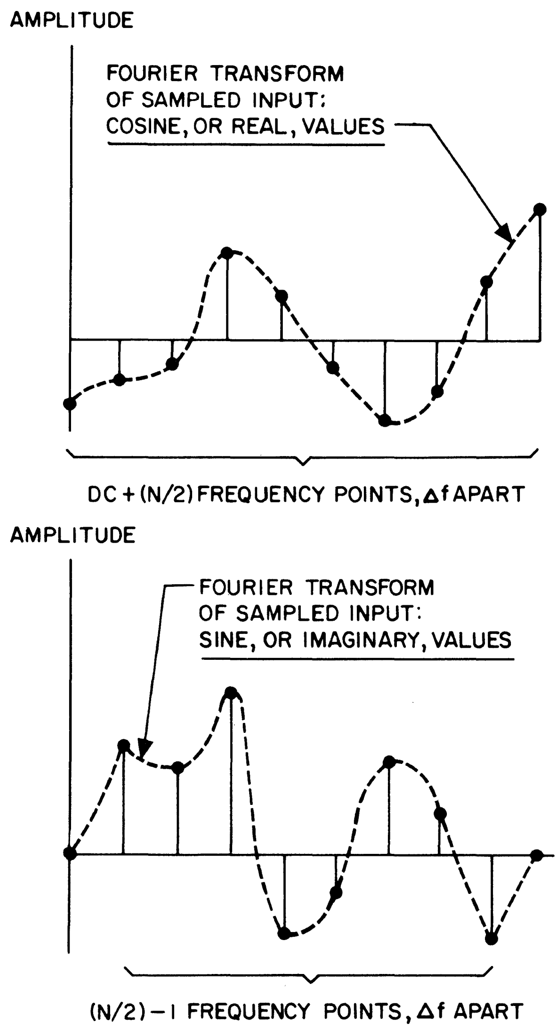
N/2 The number of frequency points: the frequency domain occupies a block of N points, but this block is broken into two displays: real or imaginary (depending on MODE switch setting). The real (cosine) display consists of (N/2) + 1 points. The imaginary (sine) display consists of (N/2) - 1 points, for a total of N points. It should be understood that these N points cover N/2 frequencies, since there is a real and imaginary value for each frequency (except for dc and highest frequency, which only have real values).

F_{max} The maximum frequency of the display, or in other words, the bandwidth. (MAX FREQ on the ADC panel). From Figure 3-4 it can be seen that:

Maximum frequency = No. of frequency points x frequency resolution

$$F_{max} = N/2 \times \Delta f$$

Figure 3-4. Frequency Domain Parameters



Choosing Sample Parameters

Table 3-1 on the opposite page summarizes the time and frequency domain parameters (Δt, F_{max}, Δf, T). The table permits the user to obtain the best trade-off on the parameters he is interested in. The SAMPLE MODE switch and the MULTIPLIER switch on the ADC panel enable the user to select a convenient value for one of the two values on each side of the SAMPLE MODE switch. The other value is then automatically fixed, as shown in Table 3-1. Thus if you choose a frequency resolution (Δf) then the total record length (TOTAL TIME) is automatically fixed. The remaining two points are determined by the block size. The following example presents a typical situation.

ADC sampling

Suppose you must have a 1 Hz frequency resolution and at the same time a 5 kHz maximum frequency. Go into Table 3-1 at line 3. In this case,

$$\Delta f = 1$$

In the last column, you see that the equation relating frequency resolution and maximum frequency is:

$$F_{max} = N/2 \times \Delta f$$

so:

$$5000 = N/2 \times 1$$
$$N = 10{,}000$$

but the largest block size in the 8K machine is 512, and in the 16K machine is 4,096. So an N of 10,000 is impossible. Something has to give. Suppose you agree to settle for a lower maximum frequency. Assuming you have an 8K machine, substitute for N a block size of 512. This will give the largest possible F_{max}.

$$F_{max} = 512/2 \times 1$$

$$F_{max} = 256 \text{ Hz}$$

If you want a 1 Hz resolution in an 8K machine, you must settle for an F_{max} of 256 Hz.

This is the kind of manipulation of ADC parameters which the user must be able to do. The parameters are set with the SAMPLE MODE and MULTIPLIER switches on the ADC, plus the BLOCK SIZE key on the Keyboard.

Aliasing

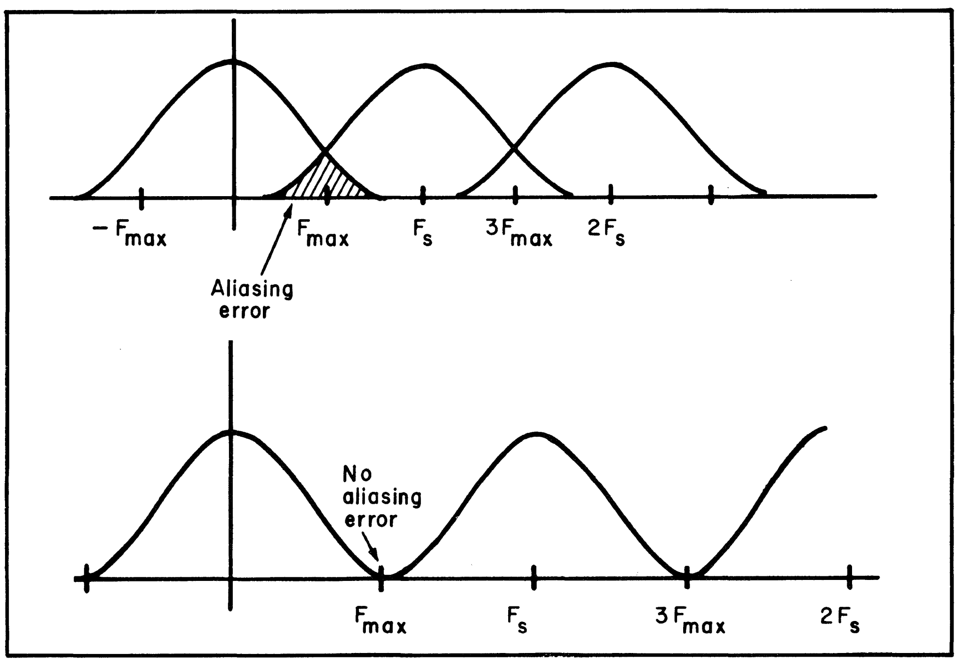
Aliasing is a phenomenon that develops with analog inputs, and which must be kept in mind to avoid possible erroneous results. It comes about from the fact that when an analog input is sampled, the spectrum replicates around multiples of the sample frequency F_s, as shown opposite. Now since F_{max} is half of F_s, it follows that if any frequencies greater than F_{max} are present they will fold back as frequencies less than F_{max}.

Table 3-1. Selecting Values for Data Sampling Parameters

Choose convenient round number for parameter shown.	Chosen parameter automatically fixes the value of parameter below, because of relationship in parentheses.	Then make either of the remaining two parameters (can't be both) as close as possible to the desired value by choosing N* in the relationships shown.
1. Δt	F_{max} $\left(F_{max} = \frac{1}{2\Delta t}\right)$	T $(T = N\Delta t)$ Δf $\left(\Delta f = \frac{1}{N\Delta t}\right)$
2. F_{max}	Δt $\left(\Delta t = \frac{1}{2F_{max}}\right)$	T $(T = N\Delta t)$ Δf $\left(\Delta f = \frac{1}{N\Delta t}\right)$
3. Δf	T $\left(T = \frac{1}{\Delta f}\right)$	Δt $\left(\Delta t = \frac{T}{N}\right)$ F_{max} $\left(F_{max} = \frac{N}{2} \cdot \Delta f\right)$
4. T	Δf $\left(\Delta f = \frac{1}{T}\right)$	Δt $\left(\Delta t = \frac{T}{N}\right)$ F_{max} $\left(F_{max} = \frac{N}{2} \cdot \Delta f\right)$

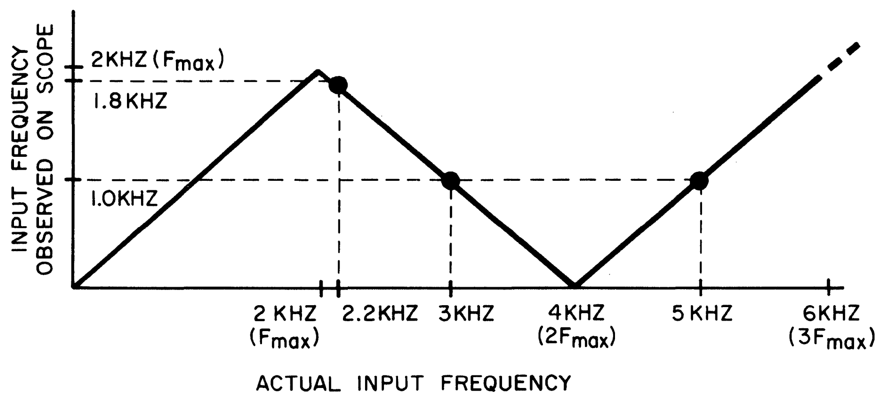
*N, the data block size, is always a power of 2.

ADC sampling



In the figure below, F_{max} is 2 kHz, F_s is 4 kHz. A frequency of 2.2 kHz will therefore be seen as 1.8 kHz; 3 kHz as 1; 5 kHz again as 1, etc.

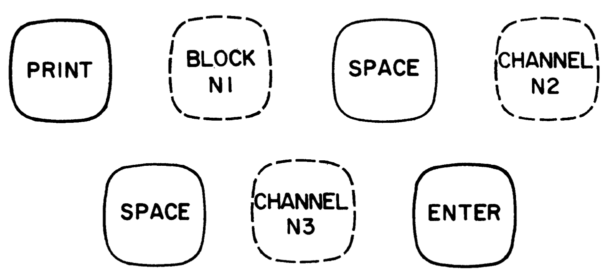
To avoid the problem, the user has to make sure that the F_{max} he sets is higher than the highest frequency in the data. (Or use a low pass filter to eliminate unwanted higher frequencies.)



Keyboard format

THE KEYBOARD COMMAND

All operations on the Fourier Analyzer are initiated from the Keyboard. Suppose you want to print out, on the Teleprinter, the contents of channels 22 to 31 of data block 2. First turn to the Index and look up the PRINT command. It is found on page 6-1, as follows:



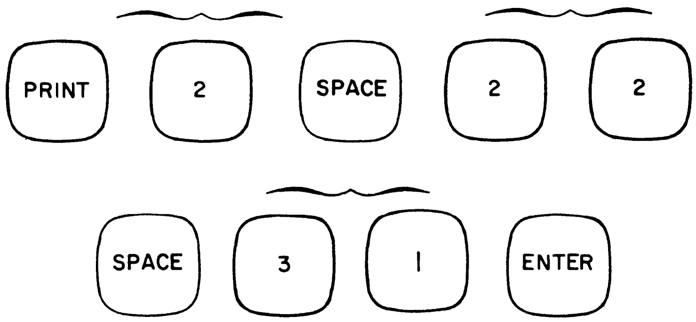
where:

N1 is the number of the data block to be printed out.

N2 is the first channel to be printed out.

N3 is the last channel to be printed out.

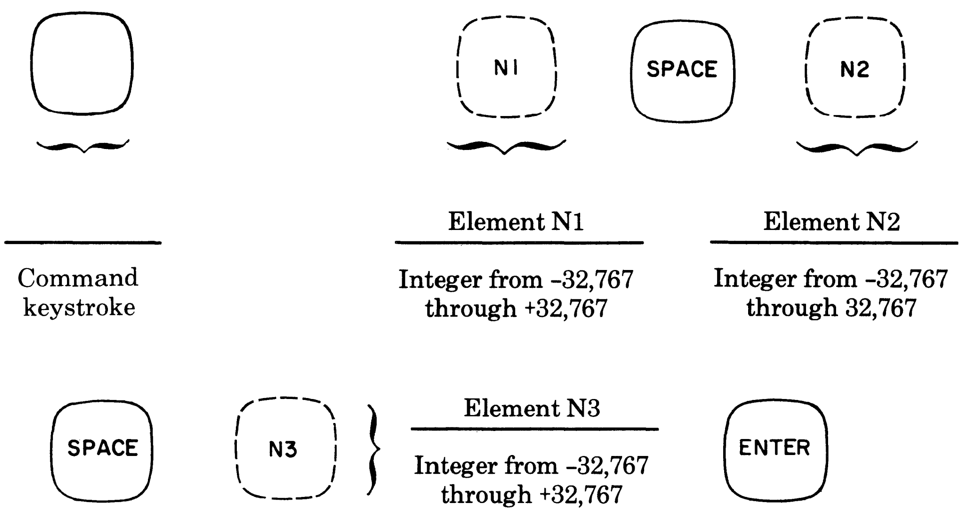
In this case, the command to be given the machine is:



Keyboard format

General Form

All operations on the Fourier Analyzer are initiated from the Keyboard. The general form of the Keyboard command is:



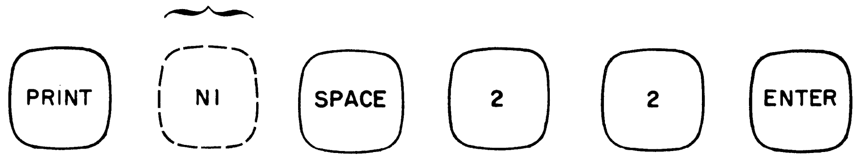
Meaning of Default

A default system is built in to the Keyboard program to save keystrokes in many situations. For instance, underneath the PRINT command is a table of default values as follows:

Element	Meaning of Element	Default Value of Element
N1	data block to be printed out	data block 0
N2*	starting channel of printout	whole data block is printed out
N3*	end channel of printout	N2

*If N2 and N3 are defaulted, whole data block is printed.

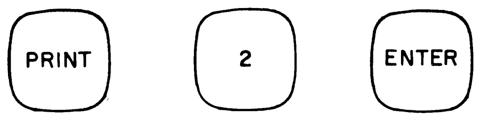
This table tells what the machine assumes if certain elements in the command are omitted. For example, if element N3 is omitted, so that the command is:



Keyboard format

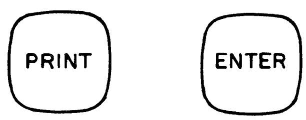
Then only one channel will be printed out, namely channel N2, because the default value of N3 is N2 - in other words, the stopping and starting channel is the same.

Suppose both N2 and N3 are defaulted, so that the command punched into the machine is:



Now, according to the table, the whole of data block 2 will be printed out.

Finally, suppose N1, N2, and N3 are defaulted, so that the command given to the machine is:



Now the whole of data block 0 will be printed out because the default value of N1 is block 0.

Thus, a considerable saving in key strokes is possible using the default system. In the above example, to print out all of block 0 requires only two keystrokes, whereas without the default system it would require seven.

In all cases, the default of elements N1, N2, and N3 causes the operation to be performed on all of data block 0.

No command is executed or entered until the ENTER key is pressed. If, during the entry of a command, a mistake is made, it can be erased by pressing the RUB OUT key, then ENTER key.

Operations Between Two Data Blocks

If a command defines an operation between two data blocks (for example, the STORE and LOAD commands, which shift data between blocks), one of these blocks is always block 0, and therefore it is not named in the command.

If the operation is a computation, then the result will reside in block 0, and the data in the named block will be left unchanged. For example, the command to multiply block 0 by block 1 is:

[MULT] [1] [ENTER]

The result will be in block 0, block 1 will be left unchanged.

A DEMONSTRATION OF SOME BASIC KEYBOARD FUNCTIONS

Using the keyboard

In the following demonstration, we will enter a pulse through the Analog-to-Digital Converter (ADC) and exercise some of the ADC controls. Then we will enter a pulse manually from the Keyboard and proceed through the Fourier transform to a series of other commands which show the coordinate systems available on the Fourier Analyzer. Finally, we will go through an example of a simple Keyboard program.

Instrument Preparation Required

At this point, it is assumed that the Fourier Analyzer has been installed and checked out according to the Operational Check provided in the Fourier Analyzer System Service Manual. Power should be on, and the READY light should be lit. For instructions on turning on the Fourier Analyzer, see Section II.

Entering an Analog Pulse

1. Set ADC controls

In order to eliminate the need for an external signal generator, we will use the internally-generated CHECK pulse as an analog input. This pulse has an amplitude of about 51 mV, a length of approximately 1100 μsec, and a repetition rate of the line frequency.

We will begin by taking 11 samples of each pulse. That means the sample interval (Δt) must be 1100 μsec/11 = 100 μsec. Therefore, on the ADC set:

SAMPLE MODE to kHz/μs

MULTIPLIER to black 100

To enable the CHECK signal to be entered into the ADC, set:

INPUT A attenuator to CHECK

Next set:

DISPLAY/INPUT to A/A

This tells the Fourier Analyzer to accept a single channel input. The other two positions of the switch are for dual channel input.

Next set:

TRIGGER SOURCE to FREE RUN

This will cause the Computer to take in blocks of data as fast as it can, that is, without waiting for any external trigger signal.

Then set:

REPEAT/SINGLE switch (on Keyboard) to REPEAT

This makes the Computer display successive blocks of data as they are sampled by the ADC, and permits us to observe the effects of various switch positions on the pulse. In the SINGLE position, the Computer takes in one block only and displays it. In the REPEAT position, the ADC continually records and displays analog data. The SINGLE position must be selected when it is desired to continue with a program or to proceed with further Keyboard commands.

Using the keyboard

2. <u>Set Display Unit controls</u>

Now set the following controls on the Display Unit:

POSITION to situate X axis on center horizontal line of scope

GAIN to CAL

MODE to REAL/MAGNITUDE

SCALE to straight-up

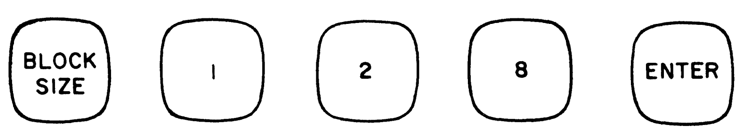
Lever switches to center, except:

DISPLAY TYPE to BAR

3. <u>Set block size</u>

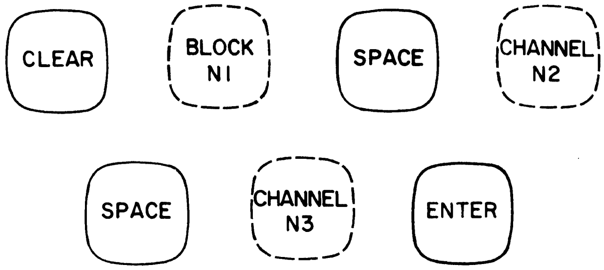
Next we choose a block size, let us say, arbitrarily, 128 points. Setting block size is a Keyboard command, and to find out what the correct command is, we look up "block size" in the Index. The block size command is given on page 7-16. This procedure of looking up commands will become less and less important as you learn the equipment, but here in this first demonstration, we will go through the exercise several times. As a further help in locating information, the remaining sections of this book are organized in logical order: Input Modes, Processing Operations, Output Modes, Programming and Editing keys, Sample Programs, and Writing New Programs.

As explained on page 7-17 to obtain a block size of 128, the keys to press are:



Note that the "128" light on the Keyboard is now lit, and there are now 128 points across the scope.

Let us enter the pulse into data block 0. To be sure there is no residual data in the block, we will clear it first (though this is not necessary, since analog input operations automatically erase data previously in a block). Looking up CLEAR in the Index, we find it is explained on page 5-4. The general form is seen to be



where:

N1 is the data block to be cleared.

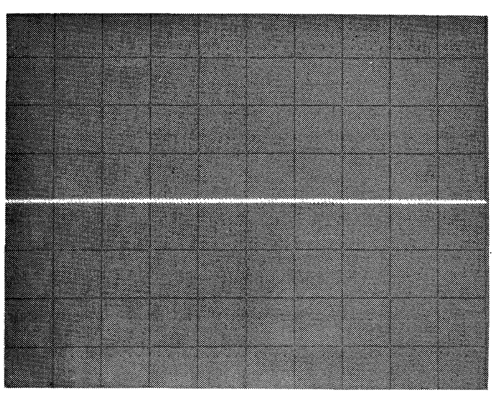
N2 is the first channel to be cleared.

N3 is the last channel to be cleared.

The default table shows that, to clear the whole of block 0, the command should be:

Press these keys now. To prove that block 0 is in fact cleared, we use the DISPLAY command. Looking it up in the Index, and checking the default table, we see that block 0 can be displayed by pressing:

Display should now look as follows:

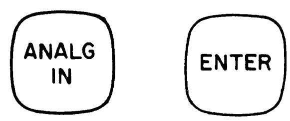


Note: If you make a mistake in entering a command, press the RUB OUT key and ENTER key then re-enter the command. The RUB OUT key, however, can only be used before ENTER is pressed. Otherwise, an erroneous command would be executed. If after entering a command, a WHAT? signal appears on the Keyboard (meaning that the command was illegal) press RESTART and re-enter the proper command. (A list of WHAT? signals as typed on the Teleprinter, with their meanings and probable causes is given in the Appendix.)

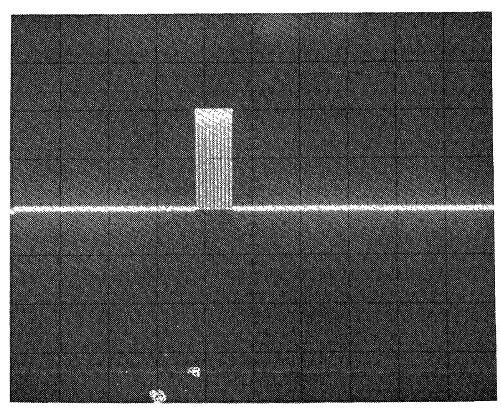
4. Give ANALoG IN command

Using the keyboard

We are now ready to give the ANALoG IN command which is explained in the Input Modes section on page 4-1. Since the CHECK pulse is going into block 0, and that is also the block we want to view (as opposed to some other block where action might be going on), we make use of the default values and press:

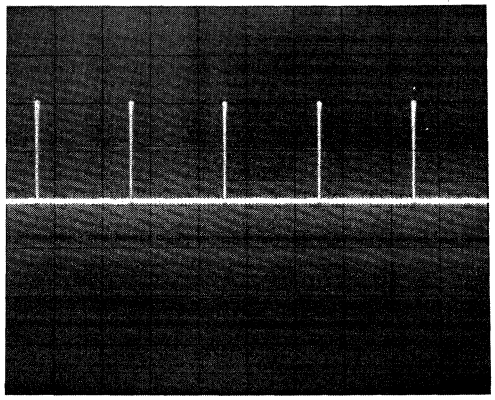


CHECK pulses should now move across the screen somewhat as shown below:

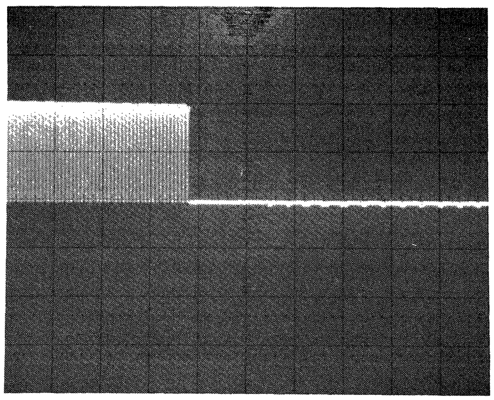


5. Vary sample interval (Δt)

Let us vary the sample interval (Δt) and see how this affects the display. Slowly switch the MULTIPLIER switch on the ADC toward the right, increasing the sample interval (Δt), and note how fewer and fewer samples of each pulse are taken. Turn switch to the left, thus decreasing the interval, and note how more and more samples of each pulse are taken. The two extremes should look approximately as follows:



Large sample interval, e.g., 1 — 2K μsec



Small sample interval, e.g., 10 — 20 μsec

Reset:

MULTIPLIER to black 100

Relationship of data sampling parameters

Note that the SAMPLE MODE switch setting says kHz-μs. This is because there is a reciprocal relationship between sample interval (Δt) and the maximum frequency (Fmax) which the ADC can detect without aliasing in the input signal. For example, at the current setting of 100 μs, the maximum frequency detectable is 5 kHz (as can be seen from the blue numbers). There is also a reciprocal relationship between the total record length (T) and the frequency resolution (Δf). These relationships were explained earlier in this section, beginning page 3-4. Some of the information is repeated with the ANALoG IN command in the Input Modes section.

6. Vary trigger controls

The positions of the SAMPLE MODE switch divide or multiply the MULTIPLIER setting by 1000.

Set:

SAMPLE MODE to left-hand Hz/ms (Δ TIME)

and note that, when the MULTIPLIER switch is set to its left-most position, the effect of very large sample intervals (i.e., 10 msec for a 1 msec pulse length) can be seen on the display. Also, if you look carefully at the TRIGGERING light, you will see it flicker quite slowly: this is because it takes a long time (128 times 10 msec, or 1.28 seconds) for each data block to be collected in memory. Now reset:

SAMPLE MODE to kHz/μs

MULTIPLIER to black 100

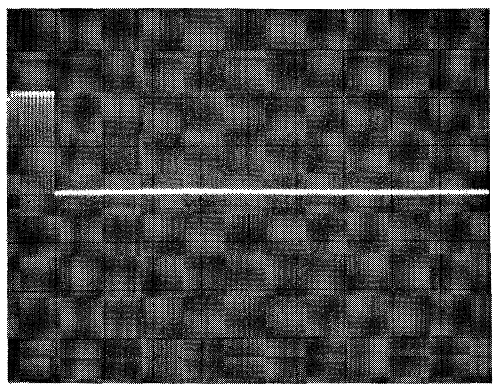
Now set:

Using the keyboard

TRIGGER SOURCE to INTERNAL (A) (the "A" means that this mode of triggering can only be done on input channel A)

SLOPE TO POS

and adjust the TRIGGER LEVEL until the pulse appears on the left of the scope display. Display should appear as follows:



Here the ADC is triggering on the positive slope of the pulse. Set:

SLOPE to NEG

and note that pulse disappears, since now the ADC is triggering on the negative slope, beyond which there is no pulse. Set:

SLOPE to POS

Again turn the MULTIPLIER switch right and left several positions to observe how changing the sample interval changes the pulse display. Finally, set:

TRIGGER SOURCE to LINE

Now the ADC is triggering at the frequency of the line.

7. Hold block of analog data

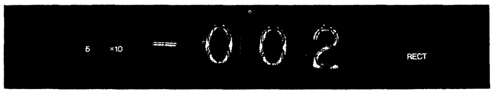
Next, let us hold one block of data in memory. To do this, we must first set the REPEAT/SINGLE switch to SINGLE. Note that the scale factor display is blank when the switch is in the REPEAT position. Now set:

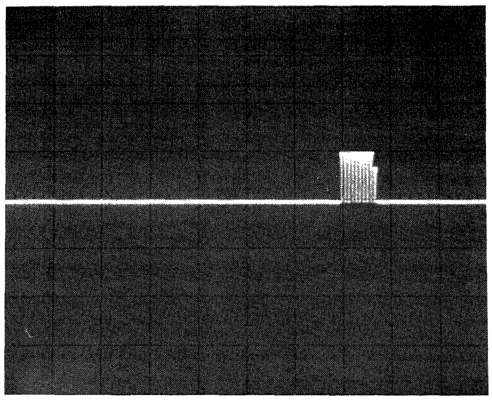
REPEAT/SINGLE switch to SINGLE

The scale factor display now comes on and the block of data in memory is displayed.

Understanding the scale factor display

The scale factor display gives the vertical scale of the scope. The factor is expressed as 1, 2 or 5 times 10 to some exponent. For example, the scale factor for the above CHECK pulse might be 5 x 10 -002, meaning 5 x 10^{-2}, so that each vertical division is 0.05, or 50 millivolts.





Thus the CHECK pulse would be on the first division. This scale factor is displayed automatically at all times except for special cases such as REPEAT/ SINGLE switch in REPEAT position, as we saw earlier. Turn the SCALE switch left and right and note the increase and decrease in the size of the pulse, and how the scale factor changes accordingly.

Review

So far we have taken an analog pulse into the ADC, seen how some of the sampling controls work, looked at triggering, and explained the scale factor display. Next we want to put in a pulse manually, then do a Fourier transform, and see the coordinate systems available on the Fourier Analyzer. We will also exercise some of the Display Unit controls.

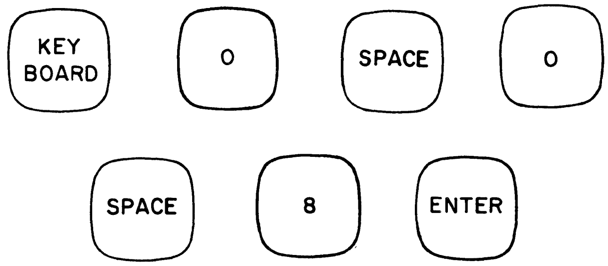
Entering a Pulse Manually

Now let us look at another means of entering data, namely, the manual Keyboard method. First then, we look up the sequences to this command under "Manual Data Input" or "Keyboard Data Input" in the Index. The command for Keyboard data entry is given on page 4-12 and is a long one. We will not duplicate any of the information, but instead will simply show the keys that need to be pressed.

First, let us clear the data block. Press:

CLEAR ENTER

Since the pulse will be rectangular, we will use the block-fill mode of entry (same value in a succession of channels). Let the pulse be 9 channels long. Therefore, press:



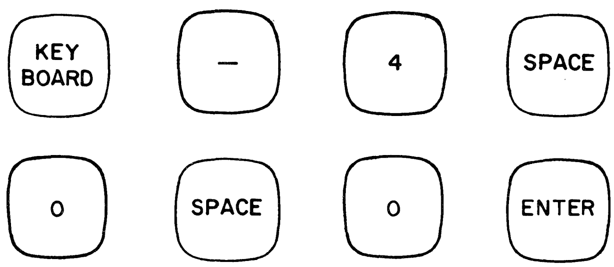
Using the keyboard

Note that we fill to channel 8 to set a pulse 9 channels long because the first channel is labeled 0.

Check that the BUSY light comes on, meaning that the machine is waiting for further Keyboard commands.

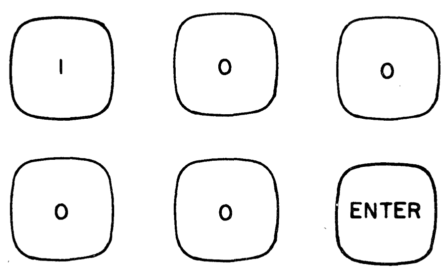
8. Enter scale factors

Let the block multiplier be 10^{-4}; since this is a rectangular pulse in the time domain, the coordinate code is 0, (per the information on the scale factor command, page 4-12). The frequency code doesn't matter, so we will arbitrarily use 0. Therefore, the keys to press are as follows (use the minus key in the numerical Address group, not the Arithmetic group):

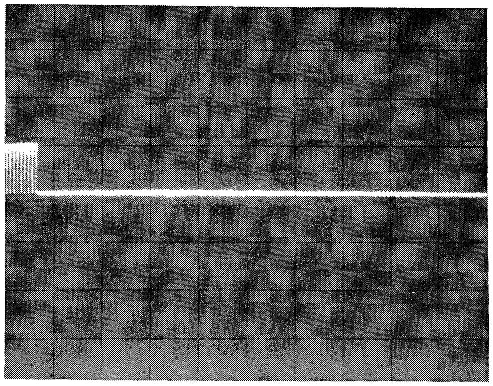


9. Enter data

Since we are in the time domain, we are entering real values. The block multiplier put in was 10^{-4}. Let us aim for a 1-volt amplitude on the pulse. Therefore, we press:



The pulse should now appear on the screen as shown below. Note that the READY light is now on.



To save having to re-enter the pulse later, should the need arise, we can store it in block 1 by pressing:

STORE 1 ENTER

Now it is in both block 0 and block 1, with block 1 being displayed. To view block 0 again, press:

DSPLY ENTER

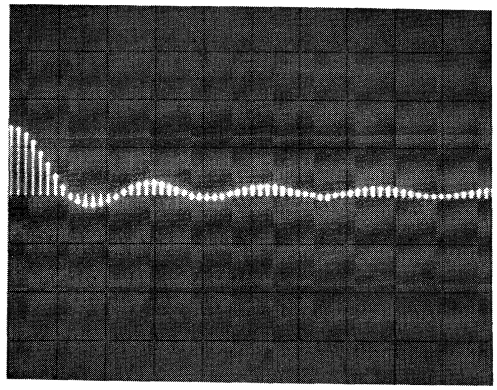
10. Do Fourier transform

We are now ready to do the Fourier transform. Rather than look this up, by now we should be able to guess intelligently as to what the command should be for a transform in block 0. Namely:

F ENTER

The Fourier transform consists of two spectral series: a real or cosine series and an imaginary or sine series. The MODE switch on the Display Unit should already be set to REAL/MAGNITUDE and the display should appear as follows:

Using the keyboard



This is the familiar curve, sin x/x.

Determining horizontal scale

The vertical scale we know how to read, from the discussion on page 3-19. The pulse has no frequency, since it was manually entered, but had it been taken in via the ADC, it could be read two ways: (1) a rough scale is available by considering each horizontal division on the scope to be $F_{max}/10$ (there are 10 horizontal divisions). A more accurate scale is determined by simple calculation. If the SAMPLE MODE switch is in one of the two left-hand positions (as it is in our case), then the formula to use is:

$$\frac{F_{max}}{N/2} = \Delta f$$

F_{max} is read directly from the SAMPLE MODE and MULTIPLIER switches; N is block size. Δf is the frequency resolution, and the horizontal scale then reads dc on the vertical axis, next point to the right is $1\Delta f$, next $2\Delta f$, etc. In our case, therefore, the calculation is:

$$\frac{5000}{128/2} = 78.12 \text{ Hz}$$

There is 78.12 Hz between each point on the scope, with 0 Hz on the left.

If the SAMPLE MODE switch is in one of the two right-hand positions, then the formula is:

$$\Delta f(N/2) = F_{max}$$

where Δf is read directly from the switch settings, N is the block size.

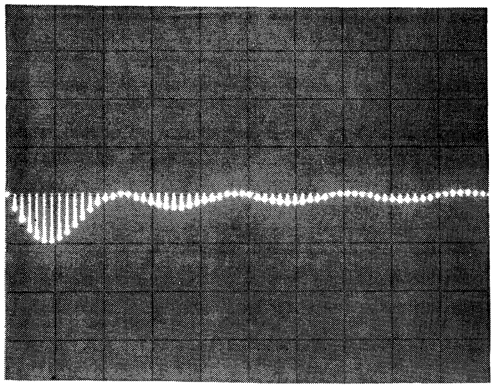
The full explanation of how these various data sampling parameters are related was given earlier in this section, beginning page 3-4. It is also repeated with the Analog Input command information in Section IV. In many cases, of course, the user will be interested in setting frequency parameters, and then will have to use other relationships to determine the values in the time domain.

The real (cosine) and imaginary (sine) series

With the MODE switch in the REAL/MAGNITUDE position, the real (cosine) series is displayed. Now set:

MODE switch to IMAGINARY/PHASE

The resulting display of the imaginary portion of the spectrum should appear as follows:



Polar magnitude and phase display

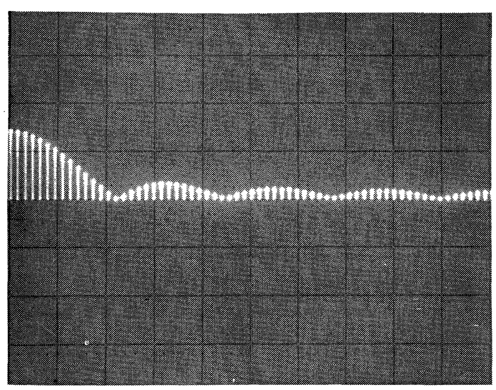
Another Keyboard command will convert the above display to polar coordinates, i.e., magnitude and phase. First set:

MODE switch to REAL/MAGNITUDE

then press:

POLAR ENTER

The display should present the magnitude portion of the spectrum as follows:

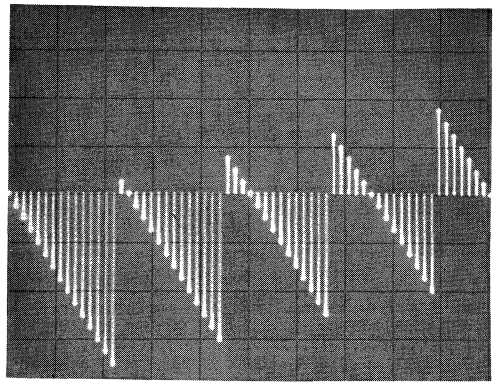


To display the phase portion of the spectrum, set: :

MODE switch to IMAGINARY/PHASE

Now the scope shows the phase display, with the scale reading 0 to +180° on the top half of the screen, and to 0 to -180° on the bottom half.

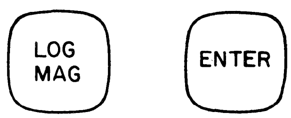
Using the keyboard



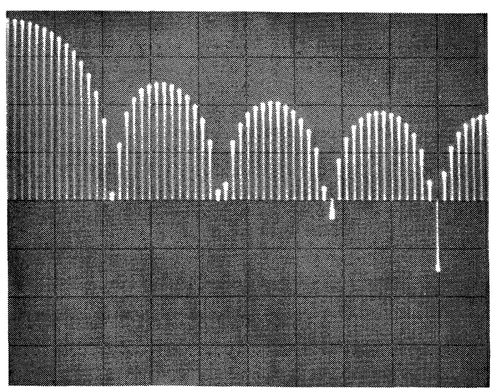
Frequency scale, of course, is as before. Note that the scale factor display is blank. For phase displays, the scale factor is given (and set) on the POLAR ANG/DIV switch. Vary the phase scale by turning the switch left and right. Set:

MODE switch to REAL/MAGNITUDE

and press:



This converts the vertical scale to logarithmic, giving a display as follows:

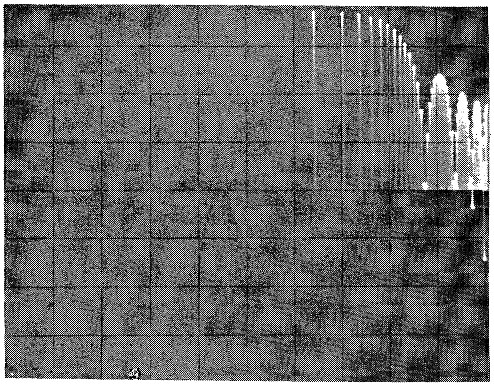


At this point, set the TYPE switch on the Display Unit to POINT and then to CONT positions and observe the effect on the display. Reset switch to BAR.

To make a Bode′ plot (log vertical and horizontal scales), we set:

ORIGIN switch (on Display Unit) to LOG

with the resulting display:



Set:

ORIGIN back to LEFT

and press the following keys to return the display to linear coordinates:

POLAR ENTER

Now again we are looking at the polar magnitude display. Next, to return to rectangular coordinates, press:

RECT ENTER

Markers are available for every 8th or 32nd point in the display, as an aid in identifying points of interest. Set the MARKER switch on the Display Unit to 8 then 32. Some adjustment of the scope intensity may be necessary to make the markers stand out.

Review

We have now seen how the typical Keyboard command works, exercised some of the Display Unit controls, and in the process observed the different kinds of coordinate systems available in the Fourier Analyzer. Next, and finally for this learning section, we will go through a simple program.

KEYBOARD PROGRAMMING

What is a program in the Fourier Analyzer?

A program is a sequence of commands which the Fourier Analyzer will perform automatically. A command element is a word key, or a number, as was described under the discussion of the general form of the Keyboard command, page 3-10. When you begin a program, these memory locations may be empty, but most likely will be partially filled with commands from a previous program. Therefore, setting up a new program can be considered to be editing the contents of the program memory. Hence, every program entry must begin with an editing command--that is, a command involving one of the Editing keys on the Keyboard.

Keyboard programming

In general, you will begin with a RePLACe command, naming line 0 as the line to be replaced by one or more new commands. These new commands will then automatically push previous commands down in the memory and off the end. The BUSY light should come on after the RePLACe command and remain on while the program is being entered.

After the program steps are entered, the END command is given to indicate the end of the program. Then the TERMinate command is used to terminate the programming mode, i.e., return to the READY mode.

To check that the steps have been entered correctly, the LIST command is used, producing a listing on the Teleprinter. In the listing, line numbers are automatically assigned to each command, these line numbers being the sum of the previous command elements up to that point. If any corrections or changes are required, one again returns to the programming mode by one of the Editing keys, in this case, RePLACe, DELETe, or INSeRT, referencing the line numbers to be changed or corrected. After the changes have been made, the TERMinate command again terminates the programming mode.

There are two ways to start the program running: one is from a given label, the other is from a given line. To start from a given label, the JUMP command is used, in which the number of the label is named. To start from a given line, the POINT command is used to set the internal pointer to the desired starting line. Then the CONTINUE key is pressed. The pointer always points to the line being processed during the running of a program, or to the line being edited during an editing operation.

A Program Example

As an introduction to programming the Fourier Analyzer, we will do a simple averaging program, again employing the CHECK signal used in our earlier discussion of the analog input. We will take the CHECK signal in randomly, i.e., so that the position of the signal in the data window will be random, then average a number of inputs to arrive at an average value of the signal. This, of course, will be a positive value since the CHECK signal is always positive. In brief, the program is as follows:

Clear block 1, as this is the block where the successive sums will be accumulated. Here, any residual data could cause erroneous results.

Analog input to block 0. This takes in a block of data which consists of the CHECK signal at some random location in the block.

Add block 0 to block 1. This is the summing step in computing the average. The sum resides in block 0.

Store block 0 in block 1. This shifts the new sum into block 1 so that a new block of data can be entered into block 0.

Count (i.e., repeat) steps from analog input the number of times desired to achieve the final sum.

Divide block 1 by the number of count times, above, to compute the average.

End program to terminate program mode.

Procedure

Keyboard programming

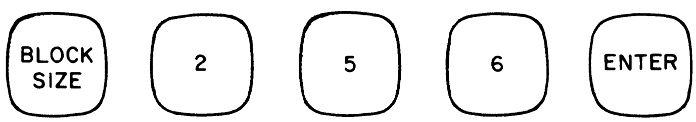
The CHECK signal has an amplitude of about 51 mV, and a length of 1100 μsec. Since its repetition rate is the power line frequency (60 Hz), the period is about 17 msec (or 20 msec for 50 Hz line frequency). We will be sure to get at least 1 pulse in every input if we choose a sample interval (Δt) of 100 μsec, and a block size of 256. This then gives a total record length of 0.1 x 256 or 25.6 msec, which is more than the 17 msec period. Therefore set:

SAMPLE MODE switch to kHz/μs

MULTIPLIER switch to black 100

TRIGGER SOURCE switch to FREE RUN

Next, set the block size by pressing:

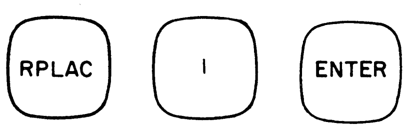


Set:

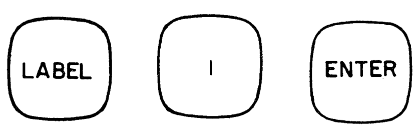
Teleprinter switch to LINE

to provide a printout of each Keyboard command. A complete list of the Teleprinter symbols, in alphabetical order, is given in the Appendix.

Now, to begin programming, use the RePLACe command. This, as all commands, can be looked up in the Index. The command is:



Next, we label the starting point of the program:



Then we clear data block 1, for the following reasons: several steps later, data in block 0 will be added to block 1, an operation which could cause erroneous results the first time around if residual data were in block 1.

For this reason, it is wise to clear block 1 at this point. Press:

CLEAR | 1 | ENTER

Keyboard programming

Next, put in another label. This is because the remaining portion of the program will be a loop, i.e., a repeated sequence of steps, and this is the starting point. Press:

LABEL | 2 | ENTER

Now the analog input is entered. We will take the data into block 0, and display that block. Utilizing default values, the command is:

ANALG IN | 0 | SPACE | 0 | ENTER

Since this program is intended to produce an average, we have to sum up the individual analog inputs. The data comes in block 0: if we add it to block 1, then the sum will reside in block 0 (per the discussion of the general command form, page 3-12, and the definition of the "+" command itself, page 5-8.) Then we will shift the summation into block 1. In short, the sums will accumulate in block 1. Therefore, the next two commands are:

\+ | 1 | ENTER

and

STORE | 1 | ENTER

To obtain the average, we will sum 250 inputs then divide by 250. This means, first, repeating the process from label 2 through the above step 250 times, which is entered by the command:

COUNT 2 SPACE 2
5 0 ENTER

"2" being the label. Then press, for the division:

÷ 1 SPACE 2
5 0 ENTER

This is the end of the program. Press:

END ENTER

And finally, to get out of the programming mode, we must press:

TERM ENTER

The BUSY light is now off and the READY light is on, indicating we are out of the programming mode. To list the program, press:

LIST ENTER

Keyboard programming

CORRECT LISTING ON TELEPRINTER	MEANING
1 L 2	Label 1
4 CL 1	Clear block 1
7 L 1	Label 2
10 RA 0 0	Analog input to block 0, display block 0
14 A+ 1	Add block 0 to block 1
17 X> 1	Store contents of block 0 in block 1
20 # 1 250 0	Count (repeat steps) from label "2", 250 times. 0 means that no repetitions had occurred at time of LIST command. If the program is stopped in the midst of running, and then the COUNT line is listed, the number of repetitions up to the STOP will be given.
25 : 1 250	Divide block 1 by 250
29 •	End of Program

Corrections and Changes

To illustrate how corrections and changes are made, let us modify the analog input command. As it now stands, it displays block 0, which is the block into which each new input is read. It will also be interesting to view block 1, which is the block where the sums of the inputs accumulate. To change this line, we can use either the two commands DELETe and INSeRT, or the single command RePLACe. Let us use the latter. Note that all of these commands are in the Editing group, thus fulfilling the requirement that all programming must begin with an Editing command.

In the RePLACe command, the line to be replaced is named. The Analog Input command is line 10, therefore press:

RPLAC | 1 | 0 | ENTER

The new command is to enter data into block 0 and display block 1. Therefore, press:

ANALG IN | 0 | SPACE | 1 | ENTER

Now, to exit the programming mode, press:

TERM | ENTER

Again, to check the program, press:

LIST | ENTER

The program can again be listed to check the proper entry. It should appear as follows:

```
 1 L     2
 4 CL    1
 7 L     1
10 RA    0      1
14 A+    1
17 X>    1
20 #     1    250     0
25 :     1    250
29 .
```

Note that the line numbers change automatically below the line that was changed, to accommodate the new elements.

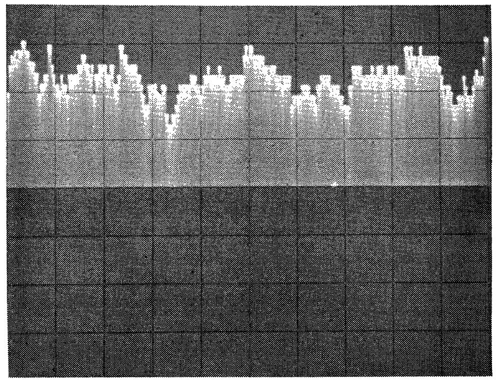
Running the Program

There are two ways of starting a program. One is by setting the pointer to the starting line number, and then pressing the CONTINUE key; the other is by using a JUMP command. We will do both. First, the JUMP command. Here, a label, rather than a line number, is used. Press:

JUMP | 2 | ENTER

Keyboard programming

The program should now run through and stop, displaying the final average. A typical display might be as follows:



Note that the average is not 51 millivolts, the height of the pulse, but something less. This is because the pulse duty cycle is less than 100%.

To start the program using the pointer, press the following keys, remembering that, with the pointer command, a line rather than a label is named.

POINT   1   ENTER

Then press:

CONT

The program should again run through.

SECTION IV

INPUT MODES

ABOUT THIS SECTION

This section provides operating instructions for the following input modes; analog (ADC), punched tape (Photoreader or Teleprinter), and manual (Keyboard). All three methods are part of the standard 5451A Fourier Analyzer.

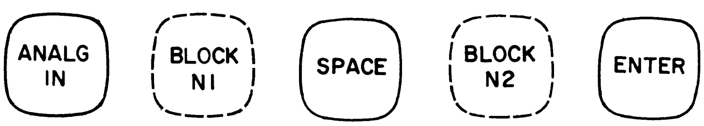
ANALOG INPUTS (ANALoG IN & BUFFereD ANALoG KEYS)

Analog inputs

The Analog Input commands take in data from one or both channels of the Analog-to-Digital Converter (ADC). There are two types of Analog Input commands: Analog In and Buffered Analog. Both commands can be used in single or dual channel operation and are unique in that they permit the user to take in data to one data block while simultaneously displaying another. This has its primary use in programs, where it allows the user to obtain an automatic display as the program proceeds. Before using an Analog Input command, be sure you understand the use of the DISPLAY/INPUT switch, the REPEAT/SINGLE switch, and the TRIGGERING controls, as explained in later paragraphs.

Analog In Command

Before giving an ANALoG IN command for single channel operation, the DISPLAY/INPUT switch on the ADC has to be in the channel A/A position. In this case, channel A will be the input channel. The command form is as follows:



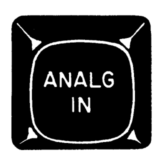
where:

N1 is the data block into which input A is to be read.

N2 is the data block to be displayed. (REPEAT/SINGLE switch in SINGLE position.)

Default table is given below:

Element	Meaning of Element	Default Value of Element
N1	Data block into which data is to be read	Data block 0
N2	Data block to be displayed	No display



For dual channel operation, the ANALoG IN command is identical to the single operation, except that the DISPLAY/INPUT switch must be in one of the DUAL positions. Channel A will be read into block N1 and channel B into block N1+1.

BUFFD ANALG **Buffered Analog Command**

The Buffered Analog command uses the last two data blocks as buffer blocks; this permits the input of data from one or both channels of the ADC into buffer blocks while simultaneously doing an operation in another block. If the length of operation in the other block is less than the input, then no data will be lost in successive input records, and a real time analysis will be performed. The command form for single channel Buffered Analog is as follows: (DISPLAY/INPUT switch on ADC to A/A position).

Analog inputs

BUFFD ANALG | BLOCK N1 | SPACE | BLOCK N2 | ENTER

where:

N1 is the data block into which input A is to be read.

N2 is the data block to be displayed. (REPEAT/SINGLE switch in SINGLE position).

Default table is given below:

Element	Meaning of Element	Default Value of Element
N1	Data block into which data is to be read	Data block 0
N2	Data block to be displayed	No display

A dual channel operation is also possible with the BUFFereD ANALoG command so long as the DISPLAY/INPUT switch is placed to one of the DUAL positions. Then channel A will be read into block N1 and channel B into block N1+1. The command form and default table are identical to single channel operation.

Sample program

The primary application of the BUFFereD ANALoG command is in a user's program, where real time analysis is desired. The program must contain a COUNT command, as real time input will continue until runout of the loop counter. (See USER PROGram key, page 9-13 & COUNT command, page 7-13.) A listing of a sample program using analog buffering is shown as follows:

AUTO SPECTRUM SUMMATION AVERAGE

1	L	1		Label 1
4	CL	1		Clear block 1
7	L	2		Label 2
10	RB	0	1	Buffered Analog Command
14	A+	1		Add block 0 to block 1
17	X>	1		Store block 0 into block 1
20	#	2	100	Repeat steps from label 2, 100 times.
25	•			End program

Setting controls

If the BUFFereD ANALoG command comes from the Keyboard there is only one input sweep. But if the command comes from the user's program stack, real time input continues until runout of the next COUNT command.

The only restriction on the buffering process is that several data blocks must be sacrificed to serve as blocks for real time input (one for single channel input and two for dual channel).

DISPLAY/INPUT Switch

This switch determines single or dual channel operation and which of the two analog input channels, A or B, will be displayed during the Analog Input command. With the REPEAT/SINGLE switch in REPEAT position, the display part of the Analog Input command is overridden. The three positions of the DISPLAY/INPUT switch are explained below.

A/A: sets ADC for single channel operation through channel A. In the REPEAT mode, channel A is automatically displayed.

A/DUAL: sets ADC for dual channel operation with channel A displayed automatically in REPEAT mode.

B/DUAL: sets ADC for dual channel operation with channel B displayed automatically in REPEAT mode.

REPEAT/SINGLE Switch

The purpose of this switch is to enable the user to continuously observe the analog input. This is done by setting the switch to the REPEAT position. Whatever channel has been set on the DISPLAY/INPUT switch will then be continuously displayed. When the switch is set to the SINGLE position, only one sample record of the input is taken. Note that, in the REPEAT position, the BUSY light is on, meaning that no other commands can be executed. The switch must be in the SINGLE position to execute other commands. The starting time for each sweep when the switch is in the REPEAT position, is determined by the setting of the TRIGGERING controls.

WARNING: If either the SAMPLE MODE or MULTIPLIER switches are changed during a read cycle, as is possible in the REPEAT mode, a lockup may occur. This will be indicated by the display going blank, and the failure of the Fourier

Analyzer to execute further commands. To remedy this, press the RESTART key and re-enter the Analog Input command.

In programming, if the switch is in the REPEAT position, then when the program reaches the analog-in step, it will pause and the analog input will be continuously displayed. As soon as the switch is set to SINGLE, the program will continue. If the switch is in the SINGLE position when the program reaches that step, it will simply display the single input and then proceed.

TRIGGERING Controls

When the Analog Input command is executed, the trigger circuit is armed. Triggering then occurs in accordance with the TRIGGER SOURCE position, as described below:

Setting controls

FREE RUN: The arming function is the trigger source; a sample record is immediately read into the Computer memory at the rate set by the SAMPLE MODE and MULTIPLIER switches. As soon as the block is filled, the trigger starts a new sample record in.

INTERNAL (A): The ADC triggers on the signal applied to channel A, as long as the signal has one division peak-to-peak amplitude on the display. The voltage level at which triggering occurs is set by the TRIGGER LEVEL control: + = positive amplitude; - = negative amplitude. The PRESET position sets the level at about 0. The SLOPE control sets the slope at which triggering will occur (positive = increasing side, negative = decreasing side).

EXTERNAL: Same as internal triggering except that the user brings in the trigger through the EXT TRIGGER input. In this mode, the trigger will work from any signal with a peak-to-peak amplitude greater than 100 mV. For simplicity of operation, no input attenuator for this mode is provided. However, a logarithmic limiter in the ADC makes triggering possible over a wide dynamic range. This results in the TRIGGER LEVEL control (for this mode only) being more sensitive for small trigger signal amplitudes than large — that is, the trigger level resolution is a constant percentage of the trigger signal amplitude. The SLOPE control works the same as the internal mode.

LINE: ADC triggers synchronously with the 50 or 60 Hz line. No slope or polarity setting is available for this position.

Input Attenuators

The two attenuator switches on the ADC are marked A and B OVERLOAD VOLTAGE, one for each input channel. The OVERLOAD VOLTAGE light is between them. The numbers represent the maximum plus or minus voltage which the ADC can accept without overload. When overload occurs, the OVERLOAD VOLTAGE light comes on, and the attenuator should be set to the next highest position.

The damage level for any particular range is approximately 46 dB (200 times peak level).

The system takes into account the OVERLOAD VOLTAGE switch settings whenever the Fourier Analyzer receives data from the ADC. Thus, all further data operations are on a calibrated basis. It is not necessary to record the OVERLOAD VOLTAGE switch setting or use a calibration signal to establish the absolute value of a frequency or time record.

The CHECK position inputs an internally generated rectangular pulse to the ADC; the pulse has a nominal 51 mV amplitude, a 1.1 msec duration, and the frequency of the line (50 or 60 Hz). It is used to obtain an analog input when no external sources are available.

Signal Inputs

The time varying phenomenon to be analyzed is applied to the INPUT A channel BNC-type connector for single channel analysis. For dual channel measurements such as transfer function, the two related time-varying signals are applied to INPUT A and INPUT B channel connectors.

How to Set Sampling Rates

Setting controls

The rate at which the ADC samples the analog input is set by the SAMPLE MODE and MULTIPLIER switches. There are four sampling parameters: The two in the time domain are the interval between samples (Δt), and the total record length (T); in the frequency domain, there is the frequency resolution (Δf), and the maximum frequency, (F_{max}). The user should have an understanding of how these parameters are related by reading the discussion in Section III beginning page 3-4. The following tells how to set the parameter values on the ADC switches.

The left side of the SAMPLE MODE switch permits the user to select either a desired maximum frequency (F_{max}) or a desired sample interval (Δt). Choosing one automatically fixes the other, because of the relationship:

$$F_{max} = \frac{1}{2\Delta t}$$

The two positions of the switch enable you to select two different ranges of magnitude, which differ by a factor of 1000. Values are set with the MULTIPLIER switch: block numbers for Δt, blue for F_{max}. The other two parameters, frequency resolution (Δf) and sample record length (T), are found by the equations:

$$\Delta f = (F_{max})/N/2)$$

$$T = N\Delta t$$

where N is the block size.

Alternatively, the right side of the SAMPLE MODE switch permits the user to select either a desired frequency resolution (Δf), or a desired sample record length (T). One automatically fixes the other because of the relationship;

$$\Delta f = 1/T$$

The remaining two parameters, F_{max} and Δt can be computed from the equations:

$$F_{max} = N/2\,\Delta f$$

$$\Delta t = \frac{T}{N}$$

The following table, a duplicate of the one on page 3-9, sums up the above relationships, and makes the choosing of desired parameters a little easier.

Table 4-1. Selecting Values for Data Sampling Parameters

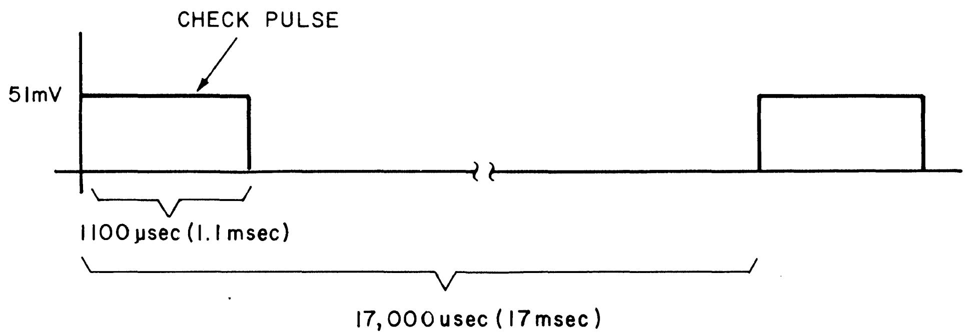
Choose convenient round number for parameter shown.	Chosen parameter automatically fixes the value of parameter below, because of relationship in parentheses.	Then make either of the remaining two parameters (can't be both) as close as possible to the desired value by choosing N* in the relationships shown.
1. Δt	$F_{max} \left(F_{max} = \frac{1}{2\Delta t}\right)$	$T \ (T = N\Delta t)$ $\Delta f \left(\Delta f = \frac{1}{N\Delta t}\right)$
2. F_{max}	$\Delta t \left(\Delta t = \frac{1}{2F_{max}}\right)$	$T \ (T = N\Delta t)$ $\Delta f \left(\Delta f = \frac{1}{N\Delta t}\right)$
3. Δf	$T \left(T = \frac{1}{\Delta f}\right)$	$\Delta t \left(\Delta t = \frac{T}{N}\right)$ $F_{max} \left(F_{max} = \frac{N}{2} \cdot \Delta f\right)$
4. T	$\Delta f \left(\Delta f = \frac{1}{T}\right)$	$\Delta t \left(\Delta t = \frac{T}{N}\right)$ $F_{max} \left(F_{max} = \frac{N}{2} \cdot \Delta f\right)$

*N, the data block size, is always a power of 2.

Setting controls

Example of Analog Input

Single input. We will use the CHECK signal on INPUT A as the analog input signal. This signal is a rectangular pulse of about 51 mV amplitude and 1100 μsec (1.1 msec) length, as shown below. The frequency of the pulse is the line frequency. This means that the period T is the reciprocal of the line frequency, or about 17,000 μsec (17 msec) for a 60 Hz line. Thus, the pulse is present in only 1/17th of the time T.

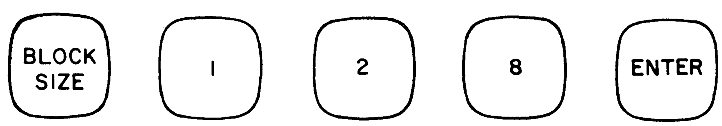


Since, as much as possible, we want to look at the pulse, and not the space in between, we will begin by choosing a Δt that will give us an adequate number of samples of the pulse only. 11 samples would be adequate: the pulse is 1100-μsec long. 1100/11 gives us a Δt of 100 μsec. Therefore, we set

SAMPLE MODE to kHz/μs
MULTIPLIER to black 100

Setting controls

None of the remaining sample parameters F_{max}, Δf, or T are critical in this case, so we can choose any data block size, N, we wish. Let it be 128. Press



As far as triggering is concerned, we can let the triggering frequency be the same as the frequency of the pulse, so set

TRIGGER SOURCE to LINE

The remainder of the triggering adjustments are not important in this case. Next, to give us an input, we set

OVERLOAD VOLTAGE switch A to CHECK

And, because this is only a single channel input from INPUT A, set

DISPLAY/INPUT to A/A

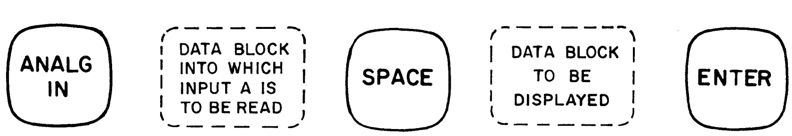
As a start, we will only look at one sweep of the input, so set:

Keyboard REPEAT/SINGLE switch to SINGLE

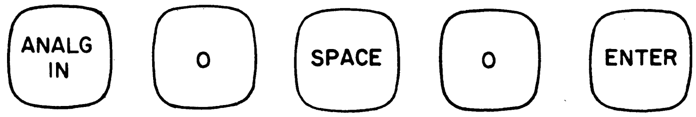
Finally, check that the display controls are set as follows:

MODE to REAL MAGNITUDE
SCALE to straight up position
Lever switches to center position

Since we have a single channel input, the form of the Analog Input command (from page 4-1) must be:



Let us read the CHECK pulse into block 0, and of course display that same block. Therefore we press:



If we wanted to continuously read in and display the pulse, we would simply set

Keyboard REPEAT/SINGLE switch to REPEAT

and give the Analog In command. The slight flicker in the display is the signal coming in at the triggering frequency (LINE).

Examples of Setting the Sample Controls

Setting controls

Setting a desired frequency resolution (Δf). A signal with a spectrum that is known to contain no energy above a frequency of about 5 kHz, is to be measured with a resolution of 10 Hz per channel. Looking at Table 4-1, we can see we are on the third line, because we want to go in with Δf as a convenient round number.

Set $\Delta f = 10$ Hz by setting ADC switches:

SAMPLE MODE to right-hand Hz/ms

then set:

MULTIPLIER to white 10

The sample record length T (TOTAL TIME) has now been fixed at 100 msec, as can be read from the above (black) switch settings. As we would like F_{max} to be about 5 kHz, we use the relationship $F_{max} = (N/2) \times \Delta f$ (from third line of Table 4-1) and see that if N = 1024, F_{max} will have closed to the desired value.

$$F_{max} = 1024/2 \times 10 = 5.12 \text{ kHz}$$

So, set the block size to 1024.

From the remaining equation (third line, Table 4-1), Δt can be computed as follows:

$$\Delta t = \frac{T}{N} = \frac{100 \text{ msec}}{1024} = 97.6\,\mu\text{sec}$$

Setting a desired maximum frequency (F_{max}). The signal in the previous example will have a display calibration of 512 Hz/Div (F_{max}/10 cm = display calibration). This may not be the most desirable number to work with. From Table 4-1, second line, now, we see that we can trade off on the value of Δf, in return

Setting controls

Table 4-1. Selecting Values for Data Sampling Parameters

Choose convenient round number for parameter shown.	Chosen parameter automatically fixes the value of parameter below, because of relationship in parentheses.	Then make either of the remaining two parameters (can't be both) as close as possible to the desired value by choosing N* in the relationships shown.
1. Δt	F_{max} $\left(F_{max} = \frac{1}{2\Delta t}\right)$	T $(T = N\Delta t)$ Δf $\left(\Delta f = \frac{1}{N\Delta t}\right)$
2. F_{max}	Δt $\left(\Delta t = \frac{1}{2F_{max}}\right)$	T $(T = N\Delta t)$ Δf $\left(\Delta f = \frac{1}{N\Delta t}\right)$
3. Δf	T $\left(T = \frac{1}{\Delta f}\right)$	Δt $\left(\Delta t = \frac{T}{N}\right)$ F_{max} $\left(F_{max} = \frac{N}{2} \cdot \Delta f\right)$
4. T	Δf $\left(\Delta f = \frac{1}{T}\right)$	Δt $\left(\Delta t = \frac{T}{N}\right)$ F_{max} $\left(F_{max} = \frac{N}{2} \cdot \Delta f\right)$
*N, the data block size, is always a power of 2.		

for a convenient round number for F_{max}, or in other words a convenient round number for display calibration. This is done as follows:

Set F_{max} = 5 kHz by setting ADC switches:

SAMPLE MODE to kHz/μs

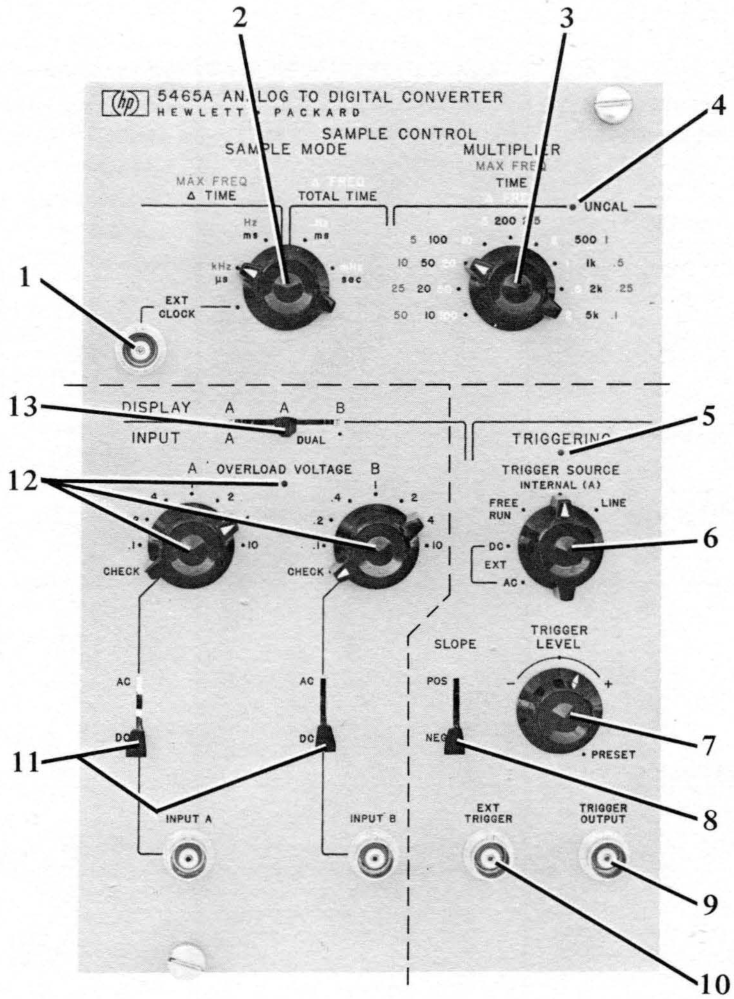
then set

MULTIPLIER to blue 5.

Δt is now fixed at 100 μsec as can be read from the black switch settings. We are now on the second line of Table 4-1 because we have chosen F_{max} to be a convenient value (i.e., 5 kHz). Therefore we experiment with values of N in the equation $\Delta f = 1/(N \times \Delta t) = F_{max}/(N/2)$ until we get as near to the desired value of 10 Hz for Δf as possible. It turns out this occurs when N remains at its value of 1024.

$$\Delta f = \frac{5 \text{ kHz}}{1024/2} = 9.6 \text{ Hz}$$

Figure 4-1. 5465A Analog-to-Digital Converter (ADC)



1. **EXT CLOCK:** permits external control of sample rate (F_s) and allows for a one-to-one correspondence between the external clock and the sample mode. Requires a TTL level input.

   The sample rate may be as high as 50 kHz for single channel operation or 20 kHz for dual channel operation. The sample rate is related to the external clock frequency (F_c) as follows:

where:

$$F_s = \frac{10F_c}{M}$$

F_s = sample frequency

F_c = external clock frequency

M = black number to which MULTIPLIER switch is set.

To use the external clock operating mode, set the SAMPLE MODE switch to EXT CLOCK. To obtain a one-to-one relationship between F_c and F_s, set the MULTIPLIER switch fully counterclockwise to the black 10 position. At any other position, F_s is divided down from F_c proportionately.

2. **SAMPLE MODE:** in left half of range, selects maximum frequency (F_{max}), called MAX FREQ, and sample interval (Δt), called Δ TIME.

In right half of range, selects frequency resolution (Δf) called Δ FREQ, and total record length (T), called TOTAL TIME.

Switching between the two positions on either side is equivalent to multiplying or dividing MULTIPLIER switch value by 1000.

See Table 4-1, page 4-9, for instructions on choosing parameters.

3. **MULTIPLIER:** selects values for parameters chosen by SAMPLE MODE SWITCH.

4. **UNCAL:** lights when input sampling conditions set up by SAMPLE MODE, MULTIPLIER, and DISPLAY INPUT switches are not valid.

5. **TRIGGERING Light:** indicates that ADC is sending data to Computer. When data transfer is completed, light goes off. Operations with low duty cycles, such as the input of short blocks of data at high frequencies, will only light this indicator dimly.

6. **TRIGGERING SOURCE:**

LINE: trigger operates at power line frequency. Neither the SLOPE nor TRIGGER LEVEL switches are active in this mode.

INTERNAL (A): trigger operates on signal applied to INPUT A, as long as this signal has 1 division peak-to-peak amplitude on the display. The triggering point on the signal is controlled by the TRIGGER LEVEL and SLOPE controls.

FREE RUN: trigger operates whenever an encode command is received from Computer. In this mode, blocks of data will be collected as fast as they can be accepted by the Fourier Analyzer, but not in synchronization with any external trigger signal. This mode is useful when data is to be used in computing power spectra, autocorrelation, or other functions that do not contain phase or time information between two signals.

EXT: trigger operates from trigger signal applied to EXT TRIGGER jack. In this mode, the trigger will work from any signal with a peak-to-peak amplitude greater than 100 mV. For simplicity of operation, no input attenuator for this mode is provided. However, a logarithmic limiter in the ADC makes triggering possible over a wide dynamic range. This results in the TRIGGER LEVEL control (for this mode only) being more sensitive for small trigger signal amplitudes than large--that is, the trigger level resolution is a constant percentage of the trigger signal amplitude. The SLOPE control works as it does in the INTERNAL (A) mode.

AC: places a capacitor in series with the input jack to block dc.

DC: no capacitor, signal is coupled directly through.

7. **TRIGGER LEVEL:** established voltage level on input waveform at which triggering occurs. + = positive amplitude, - = negative amplitude. SLOPE switch setting determines whether triggering occurs on positive slope (increasing side) or negative slope (decreasing side) of input waveform.

PRESET: sets trigger level at approximately 0 volts. This is useful for inputs such as sinewaves, and random noise signals.

8. **SLOPE:**

POS: triggering occurs on positive slope (increasing side) of input waveform. Voltage level at which triggering occurs is set by TRIGGER LEVEL switch.

NEG: triggering occurs on negative slope (decreasing side) of input waveform. Voltage level is set by TRIGGER LEVEL switch.

9. **TRIGGER OUTPUT:** puts out a TTL level pulse of approximately .5 μsec duration at triggering rate set by TRIGGER SOURCE control.

10. **EXT TRIGGER:** input for external trigger signal. TRIGGER SOURCE control must be set on EXT to use this input.

11. **INPUTS A and B and AC-DC switches:**

INPUT A & INPUT B: inputs for channel A and B analog signals; BNC-type connectors.

AC: puts blocking capacitor between input connector (INPUT A OR INPUT B) and input attenuator (OVERLOAD VOLTAGE switch). This capacitor passes frequencies down to 5 Hz (3 dB down). AC switch position should only be used when the dc in the signal to be analyzed is greater than about 50 times the rms amplitude of the signal itself. Input signal amplitude for switch in AC position must never exceed 200V dc plus 200 times the peak level set on the OVERLOAD VOLTAGE switch.

DC: couples input signal straight from connector (INPUT A or INPUT B) to input attenuator (OVERLOAD VOLTAGE switch). Peak signal amplitude must never exceed 200 times the peak level set on the OVERLOAD VOLTAGE switch.

NOTE

If either the SAMPLE MODE or MULTIPLIER switches are changed during a read cycle, as is possible in the REPEAT mode, a lockup may occur. This will be indicated by the display going blank, and the failure of the system to execute further commands. To remedy this, press RESTART key and re-enter ANALoG IN command.

12. **A OVERLOAD VOLTAGE B** (also called input attenuators): numbers represent input voltage ranges of ±0.1V through ±10V. The numbers are the peak voltages (+ or -) allowed in each range. Total range ±0.1V to ±10V) is 40 dB. The damage level for any particular range is approximately 46 dB)200 times peak level). An input voltage which exceeds the peak voltage set will light the OVERLOAD VOLTAGE light. This indicator lights if so much as a single sample in a record exceeds the maximum. *The Fourier Analyzer takes into account the OVERLOAD VOLTAGE switch setting whenever the Computer receives data from the ADC. Thus, all further data operations are on a calibrated basis. Therefore, it is not necessary to record the OVERLOAD VOLTAGE switch setting or use a calibration signal to establish the absolute value of a frequency or time record.*

CHECK position: enters a 51 mV pulse of 1100 μsec length into the ADC. This pulse has the frequency of the power line. When using this test signal, the ADC may be triggered in the LINE, INTERNAL, or FREE RUN mode, or in the EXT mode if an external trigger signal is provided.

13. **DISPLAY/INPUT** works in conjunction with REPEAT/SINGLE switch on Keyboard: see full explanation, page 4-2.

A/A: sets ADC for single channel operation (through INPUT A). In REPEAT mode, A is automatically displayed, and the display part of the ANALoG IN command is overridden.

A/DUAL: sets ADC for dual channel operation with channel A displayed automatically in REPEAT mode. In the REPEAT mode, the display part of the ANALoG IN command is overridden.

B/DUAL: sets ADC for dual channel operation with channel B displayed automatically in REPEAT mode. In the REPEAT mode, the display part of the ANALoG IN command is overridden.

MANUALLY ENTERING DATA FROM THE KEYBOARD

How Done

Data can be manually entered into the Fourier Analyzer directly from the Control Unit Keyboard. This mode of input, and the Teleprinter output, are a direct data interface between man and machine because they use a decimal number system. The punched tape mode (Photoreader input and Punch output), on the other hand, is a data interface between the Fourier Analyzer and itself or other computers, and uses a binary number system.

Command to Manually Enter Data

The manual data input command is in four parts as explained in the following paragraphs. Having read this, refer thereafter to Figure 4-2 at the end for easy reference.

Manual inputs

1. Enter block-fill or point-by-point fill command. Block fill means that a single data value will be entered into the channel range specified, as for example, to form a rectangular pulse. Point-by-point fill means that a sequence of different values will be entered into the channel range, as in the case of a triangular waveform. Note that there are no defaults in the block-fill command. (The block-fill command with N3 defaulted of course becomes the point-by-point fill.)

   After either of these commands is given, the BUSY light comes on, meaning that no other commands except the scale factors (step 2) and data (step 3) will be executed, unless the TERM ENTER keys are pressed. In that case, the Fourier Analyzer reverts immediately to the READY state.

2. Enter scale factors. These are:

   N1--the block multiplier exponent, i.e., N1 in the expression "10^{N1}" which multiplies every data word to be entered in step 3. Note: the data word to be entered should be as close to (but less than) the number 32767 as possible. N1 should be adjusted accordingly. Thus, for example, to enter the data word 1, it would be best to use an N1 of -4 (10^{-4}) and then use the data word 10000. If N1 had been 0 ($10^0 = 1$), and the data word punched in had been 1, the word would exist in memory as 00001, and the loss of the least significant digit would mean the loss of the data word. Keep in mind that the data word entered must be less than 32768, however. Thus, 1 could not be entered as 100,000 (x 10^{-5}). 4 would have to be entered as 4000 (x 10^{-3}), not as 40,000 (x 10^{-4}). The range of N1 is -512 to +511.

   N2--the coordinate code, from the table below, which tells the machine the type of data entry you will be making.

Coordinate Code	Time	Frequency	Rectangular	Polar	Log	Linear	Single Precision	Double Precision
0	x		x			x	x	
2	x		x		x		x	
4		x	x			x	x	
5 *		x		x		x	x	
7 *		x		x	x		x	
12		x	x			x		x
14		x	x		x			x

*With polar data, the coordinate code can be 16, 32, or 48 larger than given to indicate polarity of dc and F_n.

N3--the frequency code. This can be any number you wish from 0 through 16,383. The term "frequency code" comes from the fact that, when analog data is punched or printed out, this position in the data format is reserved for a number which represents the SAMPLE MODE and MULTIPLIER switch settings on the ADC. (The complete frequency code table is shown on page 4-19.) If N3 is defaulted, the machine assumes the previous value.

If data has already been entered in memory, and only the scale factors are being changed, the Keyboard input cycle can be terminated at this point by pressing TERM and ENTER. The Fourier Analyzer then goes from BUSY to READY.

3. Enter data

The data word is entered by simply pressing the required integer keys on the Keyboard, then ENTER. For time domain data, there will be one word, followed by ENTER, for each point. Example: 4000 ENTER. For frequency domain data there will be two words for each point, and these are entered as follows:

In the case of rectangular coordinates:

real data word [SPACE] imaginary data word [ENTER]

in the case of polar coordinates:

magnitude data word [SPACE] phase data word [ENTER]

Note: a 0 must always be entered for the imaginary or phase dc value and the imaginary or phase value of the highest frequency (F_n) whenever these two points are part of the data. The reason for this is explained under the discussion of the Fourier Transform used in the Fourier Analyzer, page 5-19.

4. Press TERM ENTER

This is to terminate the manual input cycle. It is not required for a block fill command. The system now goes from BUSY to READY, and is ready for further operations.

Important note: The first of the four Keyboard entry steps can be included in a program (i.e., determination of block or point-by-point fill). The program will halt when it reaches this step, until the remainder of the steps have been entered. Then the program will automatically continue.

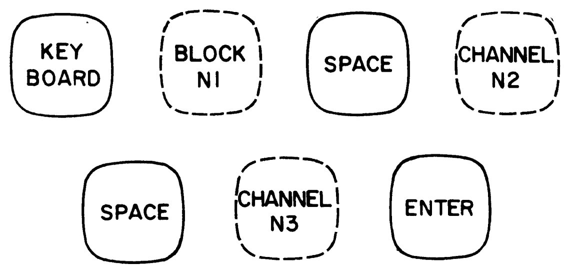
Manual inputs

Figure 4-2. Manual data entry: a 4-part command

1. **Enter Command For:**

No defaults allowed, except as shown

Block Fill

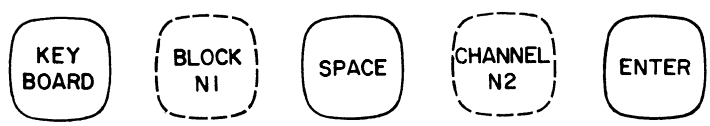


where:

N1 is the data block.
N2 is the first channel.
N3 is the last channel.

or

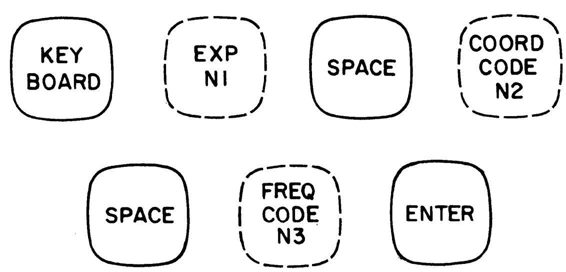
Point-by-Point Fill



where:

N1 is the data block (default value = block 0).
N2 is the first channel (default value = channel 0).

2. **Enter Scale Factors**

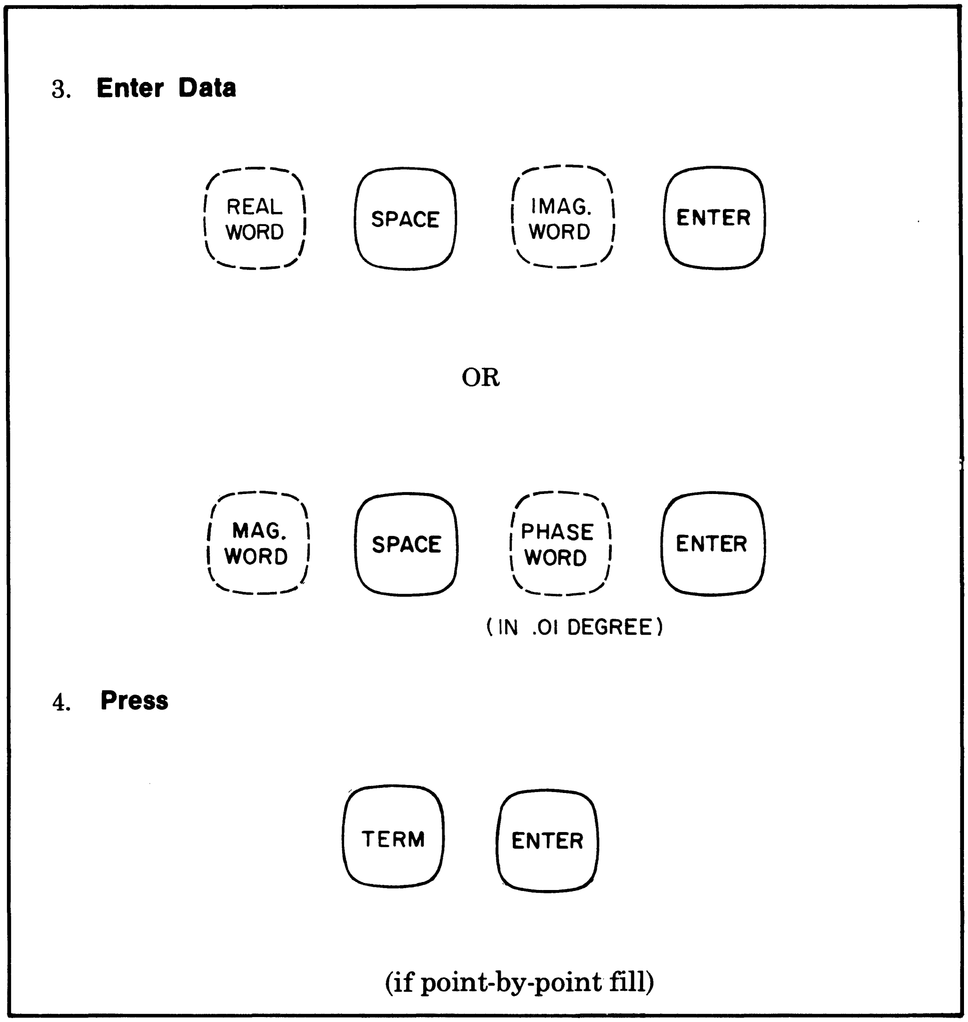


where:

N1 is the block multiplier exponent (default value = previous value).
N2 is the coordinate code (default value = previous value).
N3 is the frequency code (default value = previous value).

If N1 is defaulted, numbers will be stored in memory as they are entered; that is, without regard to existing scale factors. This is referred to as "entering integers".

Figure 4-2. Manual Data Entry: A 4-Part Command (Continued)



Alternate inputs

Alternate Inputs

The KEY BOARD command is intended to enable entry from the Teleprinter keyboard, however the Photoreader can also be used as the input device. This requires that a manual data tape be prepared on the Teleprinter in accordance with steps 2, 3, and 4 herein, and that pushbutton 3 of the Computer be pressed before entering step 1 of the manual data entry command. The PHOTO READeR and RePLACe/INSeRT commands are also affected by the state of bit 3. The following table shows the effect of bit 3 on the input devices for these commands. (This table is repeated wherever these commands are introduced.)

Command	Bit 3 = 0	Bit 3 = 1
KEY BOARD	Teleprinter	Photoreader
PHOTO READER	Photoreader	Teleprinter
RPLAC or INSRT	Teleprinter	Photoreader

PUNCHED TAPE INPUT

How Done

The optional Photoreader and High Speed Punch make an effective input/output pair as data can be punched out and re-entered rapidly and accurately. Data that is punched out for hard copy storage can be re-entered at any time by using a Keyboard input command. The paper tape reader on the Teleprinter can also be used to enter data, however processing is appreciably slower. Both methods are described in the following paragraphs, along with the applicable Keyboard input commands.

Input Command

To make the tape run through the Photoreader of Teleprinter and enter data into the Fourier Analyzer, use the following command:

Punched tape inputs

[PHOTO READR] [BLOCK NI] [SPACE] [CHANNEL N2]
[SPACE] [CHANNEL N3] [ENTER]

where:

N1 is the data block into which the data is to be entered.
N2 is the first channel in that data block to receive data.
N3 is the last channel to receive the data.

Table below gives default values for this command:

Element	Meaning of Element	Default Value of Element
N1	data block into which data is to be entered	data block 0
N2*	first channel to receive data	channel 0
N3*	last channel to receive data	successive data words on the tape are entered channel-by-channel beginning with N2, i.e., a point-by-point command results.

*NOTE: if N2 and N3 are both given, then the first data word only on the tape is entered in all the channels, N2-N3, i.e., a block fill results.

Data Format Required

All data tapes generated by the PUNCH command on the Fourier Analyzer can be re-entered by PHOTO READeR command. A tape generated by the PRINT command cannot be entered by PHOTO READeR command because data formats are different. Tapes from other computers can be entered if they have the correct data format, as spelled out in Figure 4-3.

Important note: When the tape contains frequency domain data, there must be a zero for the imaginary or phase (dc) term and for the imaginary or phase value of the highest frequency (F_n) when these points are part of the data. For example:

dc, 0, 1st real, 1st imag, 2nd real, 2nd imag,
----, last real, 0 (last imag.)

If in doubt about a tape, print it out on the Teleprinter to see if it conforms to the correct format.

Loading Tape Via Photoreader

Punched tape inputs

Data tape is loaded in the same way as the Fourier Program Tape. That is:

1. Place data tape roll in Photoreader tape holder, feed holes toward the instrument.
2. Press POWER pushbutton.
3. Press LOAD pushbutton.
4. Run the tape leader underneath the wire guide and through the pair of feed rollers.
5. Press READ pushbutton.
6. Give Photoreader input command, as pictured on preceding page.

Loading Tape Via Teleprinter

Before entering tape on the Teleprinter, pushbutton 3 of the Computer DISPLAY REGISTER must be lighted. This flags the Fourier Analyzer that data is to be taken in via the Teleprinter. After loading, the Switch Register is cleared by the program so that further data can be taken in by the Keyboard. The following paragraph explains the effect of bit 3 on the input devices under various input modes.

1. Place data tape in teleprinter reader. Lower plastic cover in place.
2. Place Teleprinter reader switch to STOP and right hand switch to LINE.
3. Code Photoreader input command as pictured on page 4-16.
4. Tape should run through. Place left switch to STOP and right switch to OFF. Remove tape and rewind.

Input Device Selection

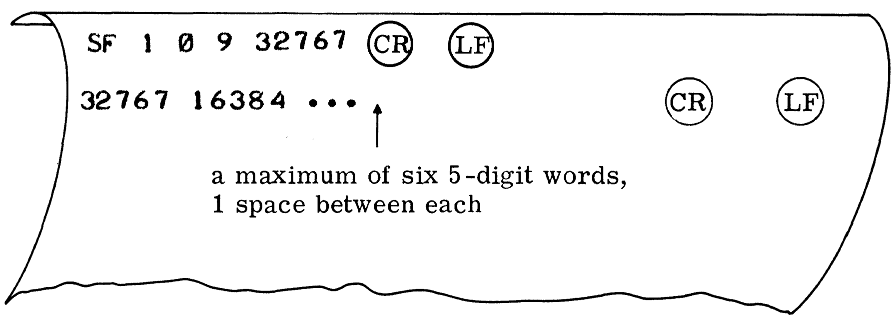
The PHOTO READeR command assumes that data will be entered on the optional Photoreader. If this device is not part of the system, the PHOTO READeR command will apply to the Teleprinter reader, but only if bit 3 of the Computer DISPLAY REGISTER is lighted. Other Keyboard input commands (such as KEYBOARD data or command entry and RePLACe or INSeRT key entry) also are affected by the state of bit 3. The following table shows the effect of bit 3 on input device selection for these commands:

Command	Bit 3 = 0	Bit 3 = 1
PHOTO READR	Photoreader	Teleprinter
KEY BOARD	Teleprinter	Photoreader
RPLAC or INSRT	Teleprinter	Photoreader

Punched tape inputs

Note: Bit 4 affects *output* conditions, see Section VI for similar table.

Figure 4-3. Required Data Format for Punched Tape



CR = carriage return

LF = line feed

SF stands for "scale factor"

1 is "k" in the expression "10^k".

0 coordinate code from following table. In above example, data is in time domain, rectangular coordinates.

Coordinate Code	Time	Frequency	Rectangular	Polar	Log	Linear	Single Precision	Double Precision
0	x		x			x	x	
2	x		x		x		x	
4		x	x			x	x	
5 *		x		x		x	x	
7 *		x		x	x		x	
12		x	x			x		x
14		x	x		x			x

*With polar data, the coordinate code can also be a number greater by 16, 32, or 48 to indicate F_n negative (plus 16), dc negative (plus 32), or both points negative (plus 48).

Punched tape inputs

9 any arbitrary number from 0 through 16,383, OR, a frequency code from the following table. The table expresses data sampling parameters in terms of SAMPLE MODE and MULTIPLIER switch settings on the ADC. All tapes produced by the Fourier Analyzer thus contain a frequency code. But it is not required in tapes from other computers so long as any number from 0 through 16,383 is inserted in its place.

In the example above, pertinent data sampling parameters were F_{max} of 0.5 Hz, Δt of 1000 msec. See table below.

Freq. Code	F_{max} (MAX FREQ)	Δt (Δ TIME)	Freq. Code	Δf (Δ FREQ)	T (TOTAL TIME)
47	50 kHz	10 μsec	63	100 Hz	10 msec
46	25 kHz	20 μsec	62	50 Hz	20 msec
45	10 kHz	50 μsec	61	20 Hz	50 msec
44	5 kHz	100 μsec	60	10 Hz	100 msec
43	2.5 kHz	200 μsec	59	5 Hz	200 msec
42	1 kHz	500 μsec	58	2 Hz	500 msec
41	.5 kHz	1000 μsec	57	1 Hz	1000 msec
40	.25 kHz	2000 μsec	56	.5 Hz	2000 msec
39	0.1 kHz	5000 μsec	55	.2 Hz	5000 msec
15	50 Hz	10 msec	31	100 mHz	10 sec
14	25 Hz	20 msec	30	50 mHz	20 sec
13	10 Hz	50 msec	29	20 mHz	50 sec
12	5 Hz	100 msec	28	10 mHz	100 sec
11	2.5 Hz	200 msec	27	5 mHz	200 sec
10	1 Hz	500 msec	26	2 mHz	500 sec
→ 9	0.5 Hz	1000 msec	25	1 mHz	1000 sec
8	0.25 Hz	2000 msec	24	.5 mHz	2000 sec
7	0.1 Hz	5000 msec	23	.2 mHz	5000 sec

32,767 is the data block calibrator associated with every data point in the block. The calibrator word system is as follows: 0 = 0; 32,767 = 1. So, in above example the block calibrator = 1. If this number is defaulted the value 32,767 is automatically inserted. The user need not be concerned with it other than know it is present and will be punched out when the PUNCH command is given.

32,767 16,384
are data words. Data word system is as follows: 0 = 0; 32,767 = 1. So, in above example where block calibrator = 1, the first data word = 32,767 x 10^1 = 32,767 x 10 = 32,7670 = 10. Second data word is 16,384 x 10^1 = 16,384 x 10 = 163840 = 5. No data words can have a magnitude greater than 32,767.

To convert any set of data into this number system: (block calibrator = 1, i.e. 32,767)

Punched tape inputs

1. Choose a k that will make 10^k slightly larger than the largest number in the data.

   For example, if data is 0.5, 1, 2, 9, choose k = 1, because 10^1 = 10, which is one higher than 9.

   Second example: if data is 0.0005, 0.001, 0.002, 0.009 choose k = -2 because 10^{-2} = 0.01.

2. Then convert data into Fourier Analyzer data words per following formula:

$$\frac{\text{data value}}{10^k} \times 32{,}767 = \text{Fourier Analyzer data word}$$

   For example, if data value = 2, as in first example above, then data word is:

$$\frac{2}{10} \times 32{,}767 = 6{,}553$$

   And, from second example, taking data value = 0.001, data word is:

$$\frac{0.001}{0.01} \times 32{,}767 = 3{,}277$$

SECTION V

PROCESSING OPERATIONS

ABOUT THIS SECTION

This section describes Keyboard functions concerned with the processing of data. The functions are broken down according to the three processing groups on the Keyboard: **Data Manipulation** keys are used to transfer data between data blocks or to change coordinates of a given data block. They do not change the domain of the data. **Arithmetic** keys perform basic arithmetic operations on data blocks. **Measurements** keys involve complex operations, most of which change the basic units of the data. Fourier Transform (F), for example, changes the data domain from time to frequency, or vice versa.

POLAR Polar Coordinates

This command converts data in a frequency domain data block from rectangular coordinates real (cosine) series and imaginary (sine) series to linear polar coordinates. If the data block is in time domain or in double precision the command is not executed. The command form is:

Polar coordinates

POLAR   BLOCK NI   ENTER

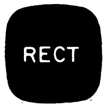
where:

N1 is the data block. Default value = data block 0.

In the magnitude display, the vertical axis is amplitude and the horizontal axis is frequency. In the phase display, the vertical axis is degrees, and the horizontal axis again frequency. Switching between the two displays is accomplished by the MODE switch on the Display Unit panel. Magnitude is always a positive value. The phase display runs 0 to +180° on the top half of the scope screen, and 0 to -180° on the bottom half. Thus, a phase of 190° would be shown as -170°.

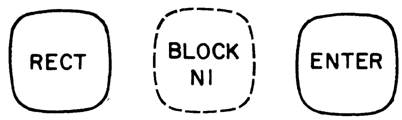
The magnitude scale can be expanded using the SCALE switch on the Display Unit panel. The phase scale is expanded by using the POLAR ANG/DIV switch. The SCALE switch has no effect on the phase display; note also that the scale factor display is dark when phase is being shown, since the POLAR ANG/DIV switch defines the scale factor.

POLAR



Rectangular Coordinates

This command converts data in a selected data block from polar, log or log polar coordinates to linear rectangular coordinates real (cosine) series and imaginary (sine) series. The command structure is:



where:

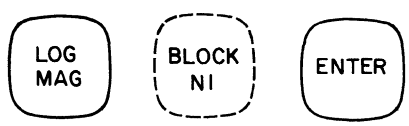
N1 is the data block. Default value = data block 0.

In rectangular coordinates, the vertical axis on the scope is amplitude and the horizontal is frequency. By using the MODE switch on the Display Unit, one can display either the real (cosine) components or the imaginary (sine) components of a spectrum.

Rect. coordinates
Log magnitude

Log Magnitude

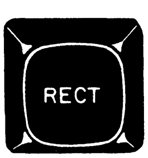
This command takes the log magnitude of the data block, point by point, and stores result back into data block. The command structure is:



where:

N1 is the data block. Default value = block 0.

If the data block is in double precision the result will be in double precision. If the data block is in frequency domain but not in double precision it will first be converted into polar coordinates (magnitude and phase) and then the log of the magnitude will be taken.

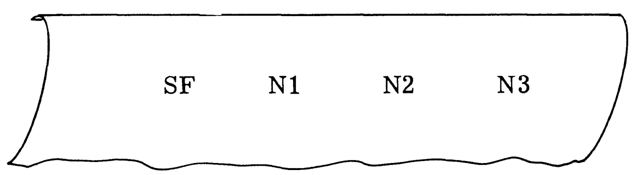


Log Scale Factor Display

The scale factor display specifies the offset, i.e., the value of the top line of the scope display in dB, referenced to 1 volt. The scope calibration is 10 dB/division. The scale switch on the Display Unit permits the user to adjust the offset to decade value such as 0, 20, 40, etc. For each position turned to the right, -4 dB is added to the offset.

Log Printouts

In a logarithmic printout on the Teleprinter (accomplished by the PRINT command), the dB values are printed in hundredths. The scale factor N1 at the start of the printout, i.e.,



is the number of dB offset *which must be added* to all dB values in the data. This offset will be some multiple of 10, regardless of what the scale factor display was before the PRINT command. In other words, the machine automatically makes sure that the offset is converted to a multiple-of-ten offset.

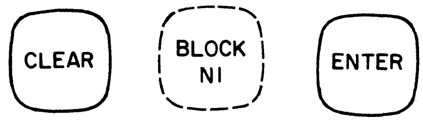
Summary of Coordinate Transformation Commands

COORDINATES OF DATA BLOCK AFTER EXECUTION OF COMMAND

Domain of Data	LOG MAG	POLAR	RECT
Time	Log	Not Executed	Linear
Freq	Log Polar	Linear Polar	Linear Rectangular

Clear

There are two types of CLEAR functions: block and partial block. The command to clear an entire data block changes domain of data block to time linear rectangular coordinates. The command form is:

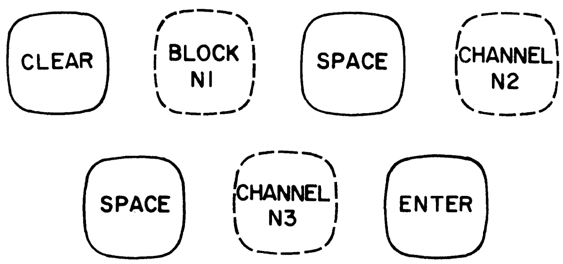


where:

N1 is the data block. Default value = block 0.

The command to partially clear a block, i.e., a given range of channels, does not change coordinates. The format is as follows:

Clear



where:

N1 is the data block.
N2 is the first channel to be cleared.
N3 is the last channel to be cleared.

Following is table of default values:

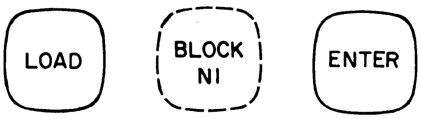
Element	Meaning of Element	Default Value of Element
N1	Data block to be cleared	Block 0
N2	First channel to be cleared	Channel 0
N3*	Last channel to be cleared	Channel N2

*If N3 is defaulted, then channel N2 will be the start and end of the range of differentiation.

CLEAR

Load

This function transfers data from block N1 to block 0. Any previous data in block 0 is written over. Block 0 is displayed. Data in block N1 is left untouched. The command is:



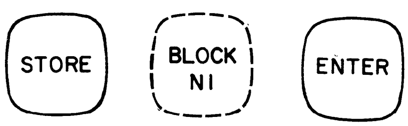
where:

N1 is the data block. Default value is 0.

To remember the difference between the LOAD and STORE commands think: "LOAD 0, STORE elsewhere."

Store

This function transfers data from block 0 to block N1. Block N1 is displayed and the data in block 0 is left untouched. Any previous data in block N1 is written over. The command structure is:



where:

N1 is the data block. Default value is 0.

To remember the difference between the LOAD and STORE commands, think: "LOAD 0, STORE elsewhere."

INTER CHNG Interchange

This function interchanges data in blocks 0 and N1. Block 0 is displayed. No data in any block is lost as a result of this operation. The command is:

[INTER CHNG] [BLOCK NI] [ENTER]

where:

N1 is the data block. Default value is 0.

In order to interchange two data blocks neither of which is block 0, and do so without loss of data in any other block, a procedure such as that shown below must be used. Three different sets of data, A, B, and C reside initially in the three blocks 0, 1, and 2. The commands shown will interchange B and C without loss of set A.

Initial conditions:

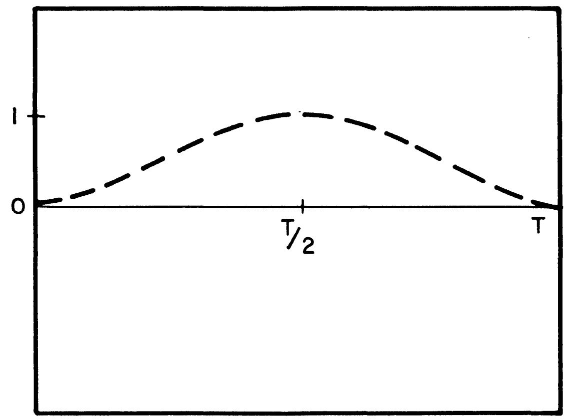
Data Block	0	1	2
Set of Data in Block	A	B	C

The following commands interchange blocks 1 and 2 without loss of block 0:

Data block	0	1	2
The command INTERCHNG 1 ENTER yields...	B	A	C
The command INTERCHNG 2 ENTER yields...	C	A	B
The command INTERCHNG 1 ENTER yields...	A	C	B

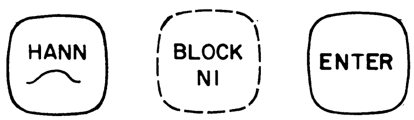
Interval-centered Hanning

This command multiplies a data block by the window function, ½ - ½ cos 2 π t/T. The data block must be in the time domain for the command to be executed.



To use this command to suppress leakage lobes in the frequency domain, Hann data block N1 and then execute a Fourier transform.

The command for Hann is:



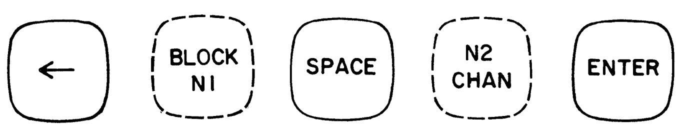
where:

N1 is the data block. Default value = block 0.

Repeated Hanns are permitted but data block must be in time domain.

Block Shift or Rotate

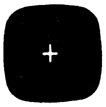
This shifts or rotates the entire block a given number of channels to the left. If in frequency domain, real will be rotated into real and imaginary into imaginary. F_n, the highest real frequency, rotates with the imaginary. The command form is:



where:

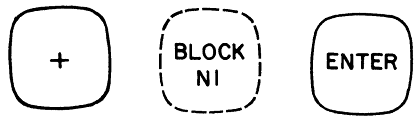
N1 is the data block to be shifted. *No default allowed.*

N2 is the number of channels the block is to be shifted.
No default allowed.



Block Addition

This command linearly adds the data in any block, point by point, to the data in block 0, and stores and displays the results in block 0. The data in the other block is left unchanged. If only one block is double precision, then a single precision add will be performed; if both blocks are in double precision, a double precision add will be performed. The command structure is:

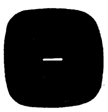


where:

N1 is any data block. Default value = block 0 (i.e., in that case, data block 0 is added to itself.)

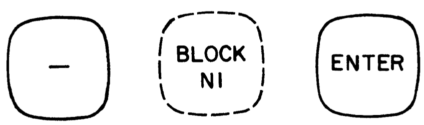
Since block addition is a linear operation, the data blocks should be in linear rectangular coordinates for more meaningful results. However, the magnitudes in polar coordinates may be added (for example, to obtain an ensemble average), but the resulting phase sum may be meaningless. If the two data blocks are of different domain or coordinates, the result will be in the domain or coordinate of block 0.

Addition
Subtraction



Block Subtraction

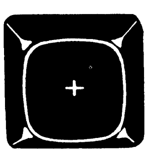
This command subtracts the data in any block, point-by-point, from the data in block 0, and stores and displays the result in block 0. The data in the other block is left unchanged. If only one block is double precision, then a single precision subtract will be performed; if both blocks are in double precision, a double precision subtract will be performed. The command structure is:

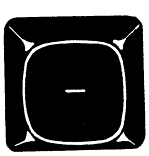


where:

N1 is any data block. Default value = data block 0 (in that case, result would be zero as data is being subtracted from itself.)

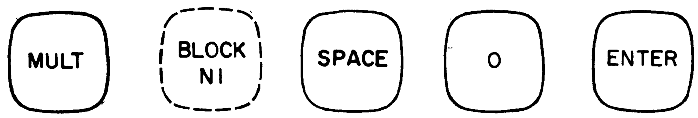
Since block subtraction is a linear operation, the data blocks should be in linear rectangular coordinates for more meaningful results. However, the magnitudes in polar coordinates may be subtracted (for example, to obtain an ensemble average), but the resulting phase sum may be meaningless. If the two data blocks are of different domain or coordinates, the result will be in the domain or coordinate of block 0.





Loop Counter Multiply

This command is used solely inside a count loop in a program; its purpose is to multiply any block by the loop counter integer. The command is executed only inside a loop, in a user entered program, while the program is running and the BUSY light is on. If the user attempts to execute the command directly from the Keyboard, a WHAT? signal will result. The command is:



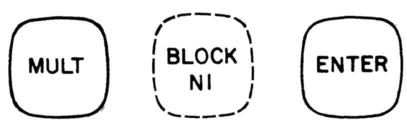
where:

N1 is the block to be multiplied. *No default allowed.*

0 is a code number for this command.

Block Multiply

This command performs a linear multiplication of block 0, point by point by any other block, N1, with the results being stored and displayed in block 0. The data in the other block remains unchanged. If only one block is in double precision, a single precision multiply will result; if both blocks are in double precision, a double precision multiply will result. The command form is:



where:

N1 is any data block. Default value = block 0 (i.e., block 0 is then squared).

The domain of block 0 determines the type of point by point multiplication. Since multiplication is a linear operation, the data blocks involved should both be in the same domain and linear rectangular coordinates for meaningful results.

Multiply

MULT Integer Multiply

This function multiplies block N1 by the integer N2, and stores and displays the result in block N1. The command form is:

MULT [BLOCK N1] SPACE [INTEGER N2] ENTER

where:

N1 is the data block. *No default allowed.*

N2 is the integer, which can be any value from -32,767 through +32,767 (excluding zero which is loop counter multiply). *No default allowed.*

To multiply a data block by a fraction or a decimal, integer multiplication and division are used. For example, to multiply a data block by $\sqrt{2}(1.414)$, integer multiply the block by 1414, then integer divide by 1000.

MULT Complex Multiply

This function multiplies block N1 by a complex number (N2 + jN3). The command format is:

MULT [N1] SPACE [N2] SPACE [N3] ENTER

where:

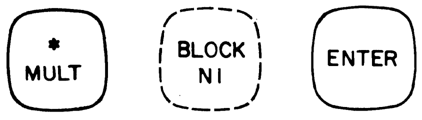
N1 is the data block. *No default allowed.* N2 is the real part and N3 is the imaginary part of a complex number, which can be any value from -32,767 to +32,767. *No default allowed.*

If N1 is in the time domain, N3 is disregarded and an integer multiplication is executed (N1 x N2).

MULT

Block Conjugate Multiply

This function multiplies block 0 point-by-point by the complex conjugate of any other block, N1, and displays and stores the result in block 0. The data in the other block remains unchanged. If block 0 is in the time domain, a non-conjugate multiply will be performed. If only one block is in double precision, a single precision multiply will result; if both blocks are in double precision, a double precision multiply will result.



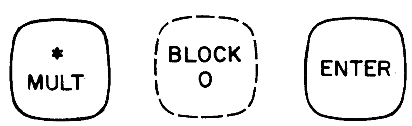
where:

> N1 is any data block. Default value = block 0 (in that case, block 0 would be multiplied by the complex conjugate of itself. This is the operation used to obtain the power spectrum from the Fourier transform.)

Since block conjugate multiply is a linear operation the data blocks involved should be in linear rectangular coordinates for meaningful results.

<u>Creating Double Precision Blocks</u>

When a self-conjugate multiply is performed (block 0 multiplied by its complex conjugate) then the result will be a double precision data block. A frequency domain data block which is completely real and where each data word (excluding dc and F_n) is 32 bits, is known as a double precision block. A self-conjugate multiply can only be executed by the following command:



where:

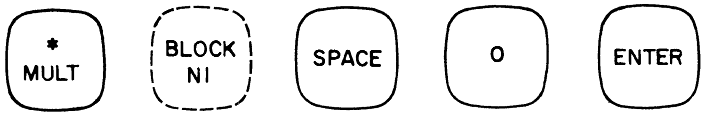
> Block 0 is in the frequency domain.

Since the results of a self-conjugate multiply are purely real, the least significant bits are stored where the imaginary parts were formerly; thus each data word is now double precision (32 bits). A self-conjugate multiply is the only processing command that can create a double precision block. A cross-conjugate multiply whose result can have both real and imaginary parts is always single precision (16 bits).

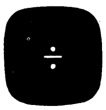
A resident Y-Command is available which allows the user to disable/enable double precision block creation. (See page 9-18).

Conjugate

This command conjugates block N1, that is, changes the sign of the imaginary part of each channel. The command is:



This command will be executed only if block N1 is in the frequency domain, single precision.

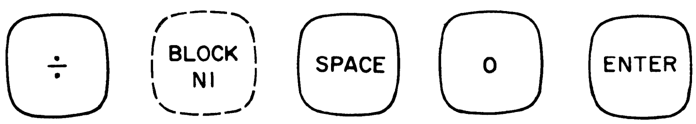


Loop Counter Divide

This command is used solely inside a count loop in a program; its purpose is to divide any block by the loop counter integer. The command is executed only inside a loop, in a user entered program, while the program is running and the BUSY light is on. If the user attempts to execute the command directly from the Keyboard, a WHAT? signal will result. The command is:

Conjugate

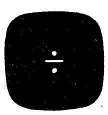
Divide



where:

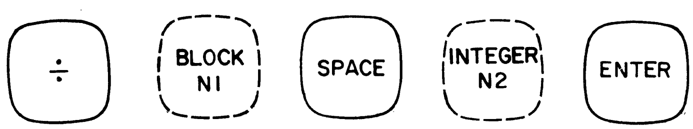
N1 is the block to be divided. *No default allowed.*

0 is the code number for this command.



Integer Divide

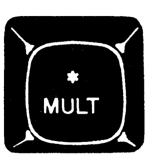
This function divides block N1 by the integer N2. The result is stored and displayed in block N1. The command structure is:

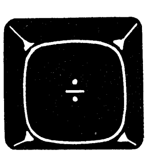


where:

N1 is the data block to be divided. *No default allowed.*

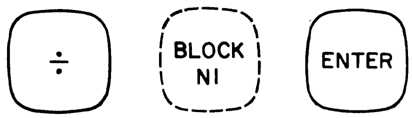
N2 is the integer, which can be any positive value from 1 through +32,767. *No default allowed.*





÷ Block Division

This command divides the data in block 0, point by point, by the data of any other block, N1, and displays and stores the result in block 0. The data in the other block remains unchanged. If both blocks involved are double precision the result will be double precision. If block 0 is in double precision but block N1 is not, then block 0 will be truncated before the command is executed. If block N1 is in double precision but block 0 is not, the result will be single precision. The command form is:



where:

N1 is the data block by which block 0 is to be divided.
No default allowed.

Since block division is a linear operation, both data blocks should be in linear rectangular coordinates for meaningful results. The domain of block 0 determines type of point by point division.

The number system used in the Fourier Analyzer is floating point on a block basis. This means that all the data words stored in a block represent numbers between -1 and +1 and that the block scale factor applies to all words in the data block and scales them to the correct value. When two data blocks are divided, the scale factors of the two are first subtracted, and the resulting scale factor determined. Then the division proceeds on a channel-by-channel basis using a complex division algorithm. Since the largest number that can be stored in memory is equivalent to 1.0 (multiplied by the block scale factor) the following procedure is used to assure that no number larger than 1.0 need be stored.

When the quotient overflows 1.0 in a channel, the quotients in all prior channels plus the overflow channel are divided by 2 and the scale factor is adjusted, leaving the result with the correct numerical value. The unprocessed channels of the dividend are also divided by 2, assuring the proper result in those channels. This procedure is repeated until the quotient between the two numbers of a given channel is less than 1.0. If this process were allowed to proceed unchecked, as in the case where the divisor block has a very low value for one channel, and the dividend block a value near unity for that channel, the result would be that the high ratio for the channel would be computed with accuracy, while all other values in the data block would be scaled down (divided by 2) so many times as to lose any significant resolution or accuracy. To limit this problem, a maximum of 8 scaledowns are allowed. This means that the quotient and dividend will be divided by 256 in the worst case. A resident Y-command is available which allows the user to select the number of scaledowns (see page 9-19). Scaledown of the data block is only performed when the division is successful. If a division is not successful, the quotient is set equal to zero and block division continues.

Example: The following example illustrates the use of a number of processing operation keys and specifically illustrates the effects of the dividend overflow. Set a block size of 128 and enter a constant value of 3.2 into all channels of data block 0 with the following set of commands:

KEY BOARD | 0 | SPACE | 0 | SPACE

1 | 2 | 7 (or 8) | ENTER

KEY BOARD | – | 4 | SPACE | 0 | ENTER

3 | 2 | 0 | 0 | 0 | ENTER

Divide

A constant value as shown in Figure 5-1(a) is now in data block 0. Store this function in data block 1 with the following command:

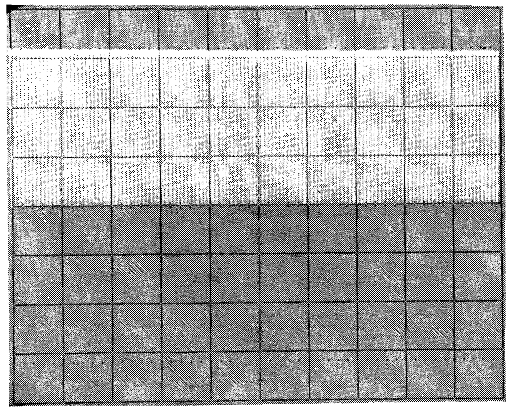
STORE | 1 | ENTER

Note that the function stored in block 1 is identical to that which is left in block 0. This can be checked with the following commands:

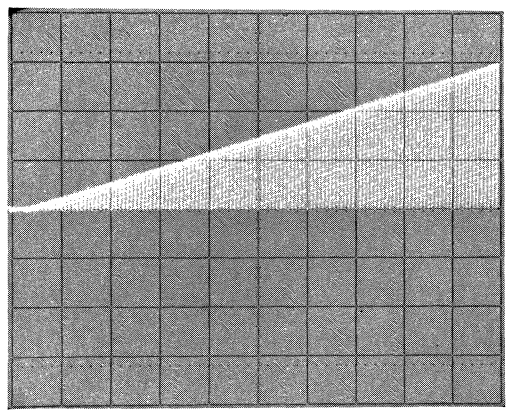
DSPLY | 1 | ENTER

DSPLY | ENTER

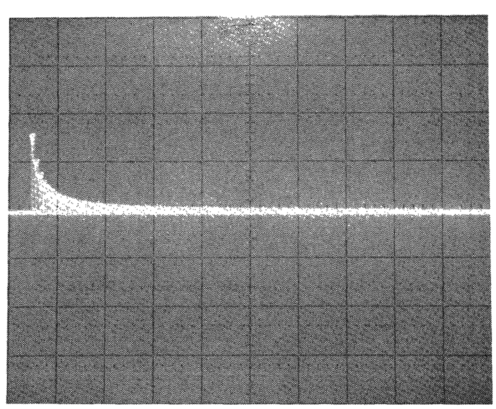
Figure 5-1. Example of Block Division, Showing Overflow



(a) Constant of 3.2 in all channels



(b) After integration of channels 5 through 127



(c) After block division is set to zero

Divide the value in data block 1 by the integer value 64 with the following integer divide command:

[÷] [1] [SPACE]
[6] [4] [ENTER]

Note that the constant in block 1 is now 3.2/64 or 0.05. If we integrate block 1 from channel 5 to 127, the integral of the constant value will be a ramp which starts with channel 5 and increases to the end of the data block. This command will execute the integration (format explained in paragraph following):

[∫] [1] [SPACE] [5] [SPACE]
[1] [2] [7] [ENTER]

Divide Integrate

The result of this integration is shown in Figure 5-1(b). If the constant value in data block 0 is divided by data block 1 the low values in the first few channels will cause the divide to overflow. Do this division with the following command:

[÷] [1] [ENTER]

The result of the block division is shown in Figure 5-1(c). Note that the first six channels are set to zero due to overflow.

[∫] Integration

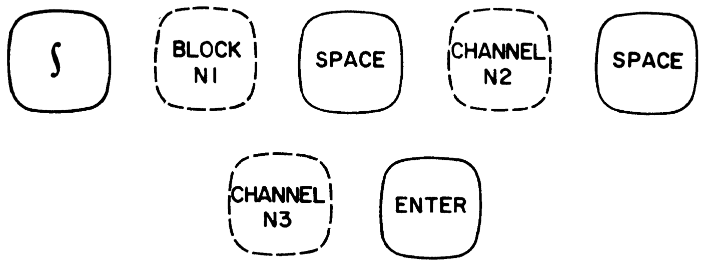
There are two types of integration commands: block integration, in which the entire data block is integrated, and partial block integration, in which only a given number of channels in the block are integrated. The command for block integration is:

[∫] [BLOCK NI] [ENTER]

where:

N1 is the data block. Default value = block 0.

The command for partial block integration is:



where:

N1 is the data block.
N2 is the first channel of the range to be integrated.
N3 is the last channel of the range to be integrated.

Default table is as follows:

Element	Meaning of Element	Default Value of Element
N1	data block	block 0
N2	first channel to be integrated	channel 0
N3*	last channel to be integrated	channel N2
*If N3 is defaulted, then channel N2 will be the start and end of the range of integration.		

This function was designed principally for integrating spectra. In a normal spectrum, the amplitude at each frequency is taken as the total area in the channel width Δf. Integrating the spectrum is then simply the sum of the amplitudes channel by channel. The individual area approximations are taken to be rectangular, not trapezoidal. In the integration process, the value of the first integrated channel remains unchanged, the value of the second integrated channel becomes the sum of the first and second channel amplitudes, the value of the third integrated channel becomes the sum of the second (sum) plus the third, etc. In the Fourier Analyzer, integration and differentiation are exactly reciprocal processes.

Differentiation

There are two types of differentiation commands: block differentiate, in which an entire data block is differentiated, and partial block differentiate, in which a given range of channels is differentiated. The command structure for block differentiation is:

[d/dx] [BLOCK N1] [ENTER]

where:

N1 is the data block. Default value = block 0.

The command for partial differentiation is:

Differentiate

[d/dx] [BLOCK N1] [SPACE] [CHANNEL N2]

[SPACE] [CHANNEL N3] [ENTER]

where:

N1 is the data block.
N2 is the first channel to be differentiated.
N3 is the last channel to be differentiated.

Just as integration in the Fourier Analyzer is merely the summing of amplitudes, so differentiation is successive subtraction of amplitudes (the so-called first difference approach to differentiation). The value of the first channel is left unchanged. The value of the second channel is than taken as the value of the second in the original function minus the value of the first. The value of the third is the original third minus the second (difference), etc.

It will be noticed that, after repeated differentiations of functions which have slopes that are relatively low, repeated integrations will not bring back the exact original function. This is because the difference between small differences (low slope) is a small number, and repeated differentiations eventually yield numbers too small for the resolution of the Fourier Analyzer. Repeated integrations cannot bring these lost values back, hence, the slight discrepancy between the reintegrated function and the original.

Element	Meaning of Element	Default Value of Element
N1	data block	block 0
N2	first channel to be differentiated	channel 0
N3*	last channel to be differentiated	channel N2

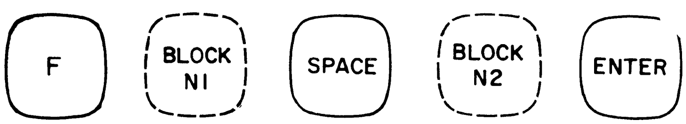
*If N3 is defaulted, then channel N2 will be the start and end of the range of differentiation.

F Fourier Transform

This command takes the forward or inverse Fourier transform of a data block depending on domain of block. The Fourier transform is taken on single precision blocks; if the data block is double precision it will be truncated before command is executed. The Fourier transform can be taken on one data block, or on two data blocks in succession.

The command structure for Fourier transform is:

Fourier transform



where:

N1 is a data block. Default value = block 0.

N2 is another data block. Default value = transform is taken only on block N1.

A Fourier transform always takes data blocks N1 and N2 from one domain to the other. The result of a Fourier transform is always in rectangular coordinates. The value of the imaginary part of the dc and the imaginary part of the highest frequency channels is always 0, as explained below under "Fourier Transform used in Fourier Analyzer."

Fourier Transform Used in Fourier Analyzer

The following is the definition of the Discrete Finite Transform (DFT) used in the Fourier Analyzer:

(forward) $$F(m\Delta f) = \frac{1}{N} \sum_{n=0}^{n=N-1} f(n\Delta t)\, e^{-\frac{i2\pi nm}{N}}$$

(inverse) $$f(n\Delta t) = \sum_{m=0}^{m=N-1} F(m\Delta f)\, e^{+\frac{i2\pi nm}{N}}$$

where:

Δt = time increment

Δf = frequency resolution

N = the total number of points in the time domain, i.e., the block size

Fourier transform

N time points theoretically suggest there should be N frequency points. Since the data computed is always a pure real time series, the negative frequencies that are normally present in a double-sided transform are perfectly symmetrical with the positive frequencies. These negative frequencies therefore are redundant data, and hence are not stored. In the Fourier Analyzer, from an N-point real time series, we compute and display N/2 positive frequencies, plus dc. Each frequency except the highest has two independent numerical values: a real (cosine) and an imaginary (sine). The highest frequency has only a real, but no imaginary value. The actual arithmetic is as follows: N/2 positive real frequency values plus dc value = N/2 + 1 real points. The imaginary side has no dc value and no value for the highest frequency, hence it has N/2 + 1 - 2, or N/2 - 1 points. Adding the real and imaginary points together we get N/2 + 1 plus N/2 - 1, or a total of N points in the frequency domain from N points in the time domain.

The display of the imaginary components of the spectrum always shows a 0 in the dc and the highest frequency channels. Likewise 0 phase shows in these channels.

When taking an inverse transform of a frequency spectrum, the Fourier Analyzer assumes that the spectrum is symmetrical and thus computes a pure real time series.

Important Note: When entering frequency domain values from the Keyboard or the Photoreader, zero must be entered for the imaginary or phase value of dc and the highest frequency, when these two points are part of the data.

Phase Considerations

Each frequency in the spectrum after a Fourier transform can be written as:

$$F(m\Delta f) = A(m\Delta f) + iB(m\Delta f)$$

The spectrum values of the real part, A, and the imaginary, B, depend on the starting point in time, i.e., the relative phase of the signal. Thus, the linear Fourier spectrum shape will vary both in amplitude and polarity from sample record to sample record, depending on the starting point of the signal. To get a constant spectrum shape, independent of the starting point of the signal, it is necessary to convert the real and imaginary components into magnitude and phase or into the appropriate power spectrum. This can be done using the

POLAR key to obtain a magnitude spectrum, or, in the case of an autopower spectrum, by doing a self complex conjugate multiply (*MULT). (See Creating Double Precision Blocks, page 5-11.)

In the case of polar coordinates, the magnitude can be written;

$$|F(m\Delta f)| = \sqrt{A^2 (m\Delta f) + B^2 (m\Delta f)}$$

and the phase can be written;

$$\angle F (m\Delta f) = \arc\tan \frac{B(m\Delta f)}{A(m\Delta f)}$$

The power spectrum is;

$$F(m\Delta f)^2 = A^2(m\Delta f) + B^2 (m\Delta f)$$

While the magnitude and power spectrum do not depend on the starting point or phase of the input time series, the phase angle, of course, does.

Amplitudes in Spectrums

Since the Fourier algorithm used only stores the positive half of the spectrum, a sine wave of peak amplitude A will yield a spectral line of amplitude A/2. The rms value for the sinewave is $A/\sqrt{2}$, so that to find the rms value in the spectrum, it is necessary to multiply the displayed amplitude by $\sqrt{2}$, i.e., 1.414. See the Integer Multiply command on page 5-10.

A complete tabulation of the values given by the Fourier transform, and the constants needed to correct these, are shown in the table below.

Fourier transform

AMPLITUDE VALUES FOR AN $A \sin \omega t$ INPUT

Function	Fourier Analyzer Gives	Mult By	To Get
Linear or Magnitude Spectrum	A/2	$\sqrt{2}$	RMS value ($A/\sqrt{2}$)
Linear or Magnitude Spectrum	A/2	2	Peak value A
Power Spectrum	$A^2/4$	2	Power ($A^2/2$)

Normalizing the Spectrum

The transform is taken without reference to the effective bandwidth set by the ADC sampling controls (for information on these controls, see page 4-5. Since the effective noise power bandwidth of the Fourier Analyzer is 1/T, where T is the sample record length, multiplication of the spectral values by T will normalize the above power spectrum to per unit bandwidth. To normalize a linear or magnitude spectrum to per $\sqrt{Hz}$, simply multiply the spectrum by $\sqrt{T}$. In this way, measurements of random noise and other non-coherent signals that can be defined as having a constant spectrum density, can be normalized to give equivalent readings independent of sample rate and bandwidth of the Fourier Analyzer.

However, if sinusoids and other coherent signals are normalized in this way, they will have amplitudes that depend on system bandwidth. Thus, if one is seeking equivalent measurements for coherent signals, it is best not to normalize.

CORR Correlation

Correlation is an operation done between two blocks of data, therefore by the rule discussed under the general form of the Keyboard command, page 3-10, one of the blocks must be block 0, and only the second block is named in the command. Results are stored and displayed in block 0. Correlation is performed on single precision blocks so that if any of the data blocks involved are double precision, they will be truncated before execution of the command.

If the correlation is between block 0 and another block, then the result is a cross correlation. If between 0 and itself, the result is an autocorrelation, as shown in Figure 5-2. The command structure is:

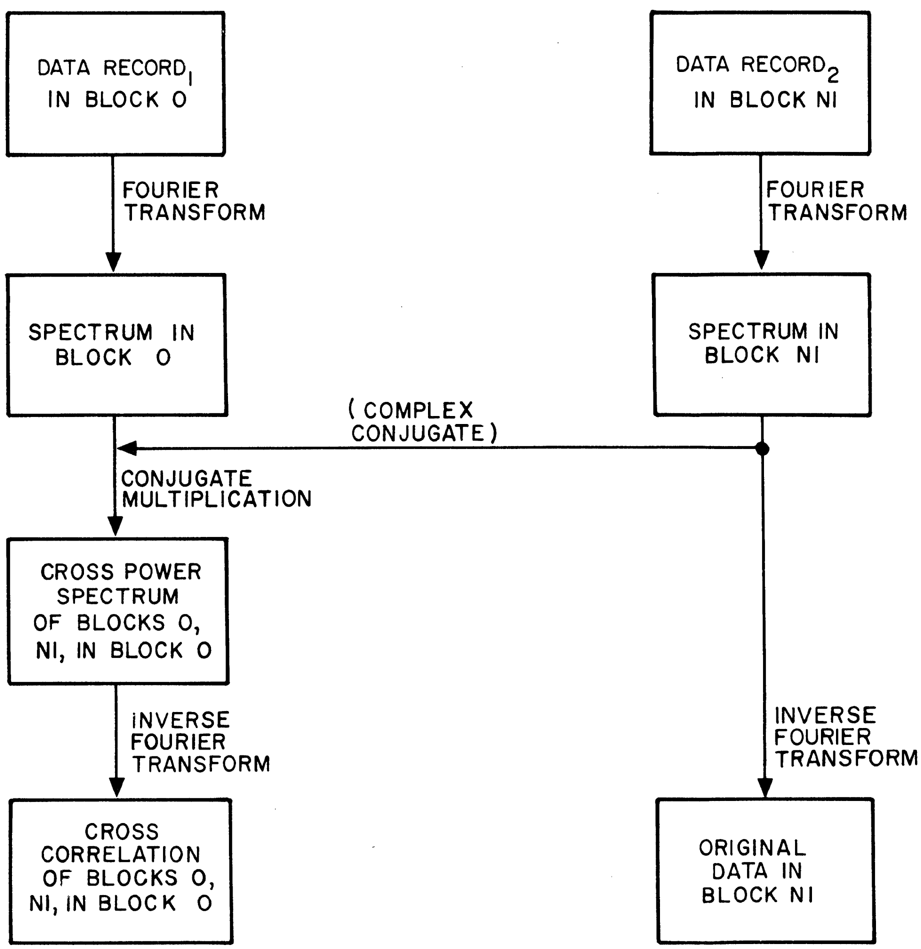
CORR (BLOCK N1) ENTER

where:

N1 is any data block. Default value = block 0 (in that case, an autocorrelation will result).

Correlation

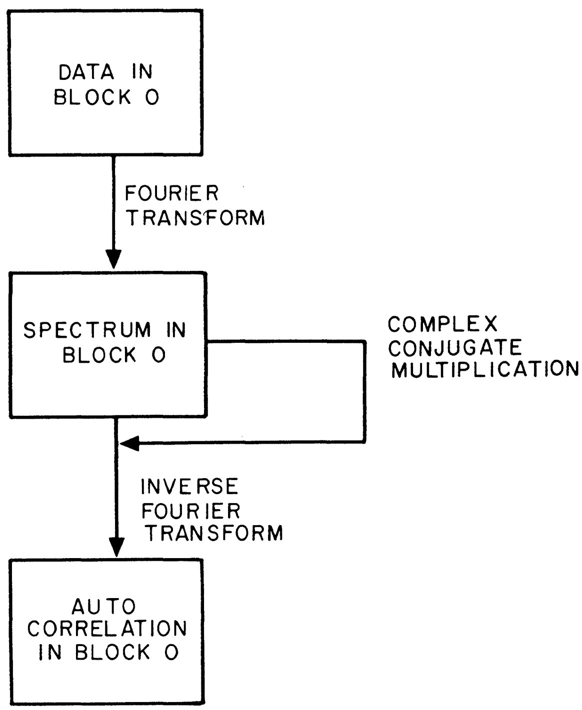
Figure 5-2(a). Flow Graph of Cross Correlation Computational Procedure



The means by which the Fourier Analyzer obtains the correlation function is as follows: first the system takes the Fourier transform of both blocks (cross correlation) or of block 0 (autocorrelation). The Fourier Analyzer does a conjugate multiply on both blocks or block 0 with itself. This yields the power spectrum. Finally, the inverse Fourier transform is taken on the power spectrum, to produce the correlation function. Flow graphs of these computations are shown in Figure 5-2b.

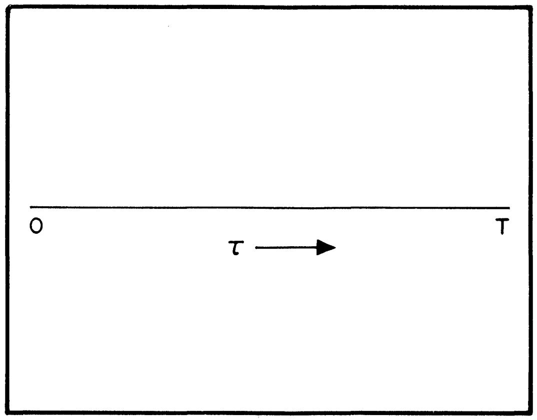
Note: when beginning a correlation with data in the frequency domain, it is necessary to multiply the result by the block size to achieve the proper scaling and calibration.

Figure 5-2(b). Flow Graph of Autocorrelation Computational Procedure

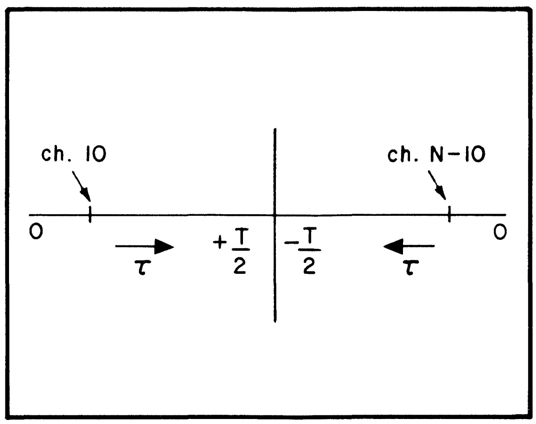


The final correlation function is in the time domain, but with the horizontal axis representing values of τ(lag) not t (time) as originally, over a range of $\pm T/2$. T is the length of the data window. This is shown in the three sets of axes on page 5-24. Note that the position of the origin can be switched with the ORIGIN switch on the Display Unit panel. The switch of origins is a display function, not a change in channel locations in the data block. With the ORIGIN switch on LEFT, the channels run from 0 on the left to N-1 on the right. Channel 10 is the tenth channel from the left origin, channel N-10 is the N-10th channel from the left origin. When the origin is set to CENTER, these channel numbers retain their same relative positions.

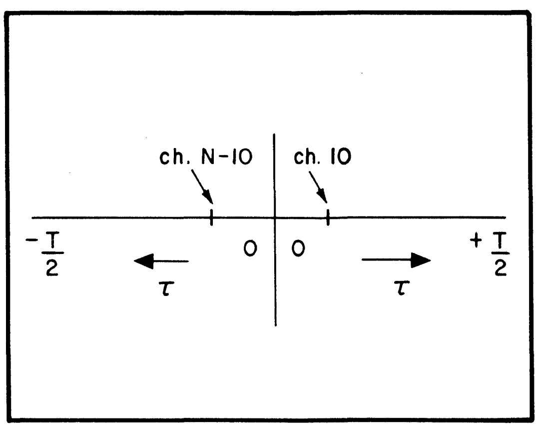
Correlation



NORMAL TIME
DOMAIN DISPLAY



AFTER CORRELATION,
ORIGIN SWITCH TO LEFT



AFTER CORRELATION,
ORIGIN SWITCH TO CENTER

Example 1: Set a block size of 128, then manually create a rectangular pulse in block 0, 11 channels wide, via the following commands:

CLEAR ENTER

KEY BOARD 0 SPACE 0

SPACE 1 0 ENTER

KEY BOARD — 4 SPACE 0 ENTER

3 0 0 0 0 ENTER

Take the autocorrelation of the pulse by pressing:

CORR ENTER

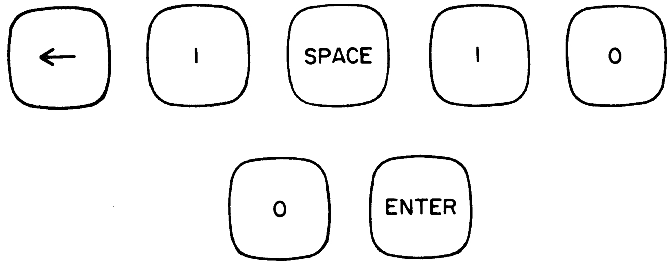
The result should be as shown in Figure 5-3(a). Switching the ORIGIN switch to CENTER should give the result shown in Figure 5-3(b).

Example 2: Set the same block size and create the same rectangular pulse as in Example 1.

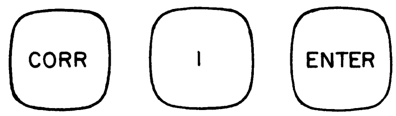
Transfer the pulse to block 1 by pressing:

STORE 1 ENTER

Shift the pulse in block 1, 100 channels to the left:



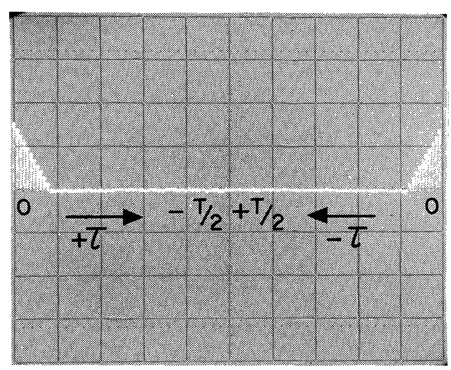
Take the cross correlation of block 0 and 1 by:



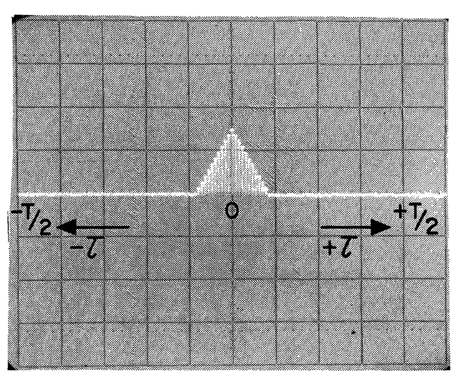
The result should be as shown in Figure 5-3(c), and indicates a lag between the two signals of 28 channels.

Correlation

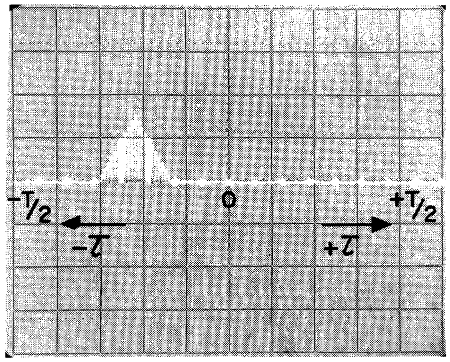
Figure 5-3. Autocorrelation (a and b) and Cross Correlation (c)



(a) ORIGIN switch to LEFT



(b) ORIGIN switch to CENTER



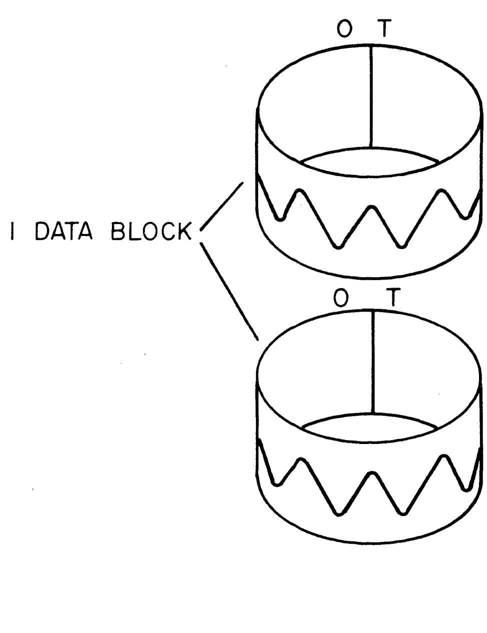
(c) After original pulse was shifted 100 channels
(8 point markers)

Wrap-around Error

In any correlation or convolution operation using the Fast Fourier Transform, the effect known as wrap-around error must be taken into account. It occurs whenever one of the signals to be correlated or convolved is 0 for less than half of the record length T.

Wrap-around error

To understand the cause of wrap-around error, we must realize that correlation or convolution in the Fourier Analyzer is done by going through the frequency domain. This means that whatever portion of the original signal the Fourier Analyzer takes in, it assumes the data block to be periodic, with the period having the record length T, as shown in Figure 5-4, line 3. Now when correlation or convolution is done, an error results because the original signal was in fact not periodic in this way. The very first increment of shift, as shown in Figure 5-4, line 4, introduces an error, because, for example, point (a) is being multiplied by point (b), rather than (a') by (b'), which should be the case, as shown in Figure 5-4, lines 4 and 5. The term "wrap-around error" comes from the fact that the repetitive records can be conceived as a single record, bent or wrapped around in a circle, as shown below:



The way to eliminate the problem is shown in Figure 5-4, lines 6 through 9. First we clear out T/4 channels on each end of one data block to be shifted. Now, when this record is shifted, it does not immediately run into the next repetition in the other record. In fact, it can be shifted a total of T/4 increments before this happens, as shown in Figure 5-4, line 7, and the correlation is thus valid over this range. But shifts from T/4 to T/2 of course do run into the next repetition, and hence are no good, just as, in the first case (Figure 5-4, line 4) the very first shift was no good. (The same applies to shifts in the opposite direction, since correlation involves a +T/2 and a -T/2 shift.) We now have a good correlation from 0 to |T/4|, bad from |T/4| to |T/2|. Therefore, in the final display, we simply clear out, or ignore, the outer ¼ of the correlation or convolution function.

The actual procedure is as follows: (See also sample correlation subroutine in Section VIII.)

1. In the case of autocorrelation, take the data into block 0, and STORE in block N1. (By the nature of the STORE command, this will now leave the data in both block 0 and block 1.) In the case of the cross correlation, simply introduce the data into blocks 0 and N1.
2. Next CLEAR out the first and last quarter of either of the blocks.
3. Correlate the blocks, using the CORRelation command.
4. When the resulting correlation function is displayed, the last quarters will also be in error. Either ignore these quarters, i.e., the channels running from +T/4 to +T/2 shifts, and from -T/4 to -T/2 shifts, or clear them out also. The half of the correlation that remains, will always be correct.

Wrap-around error

Figure 5-4. Wrap-around Error

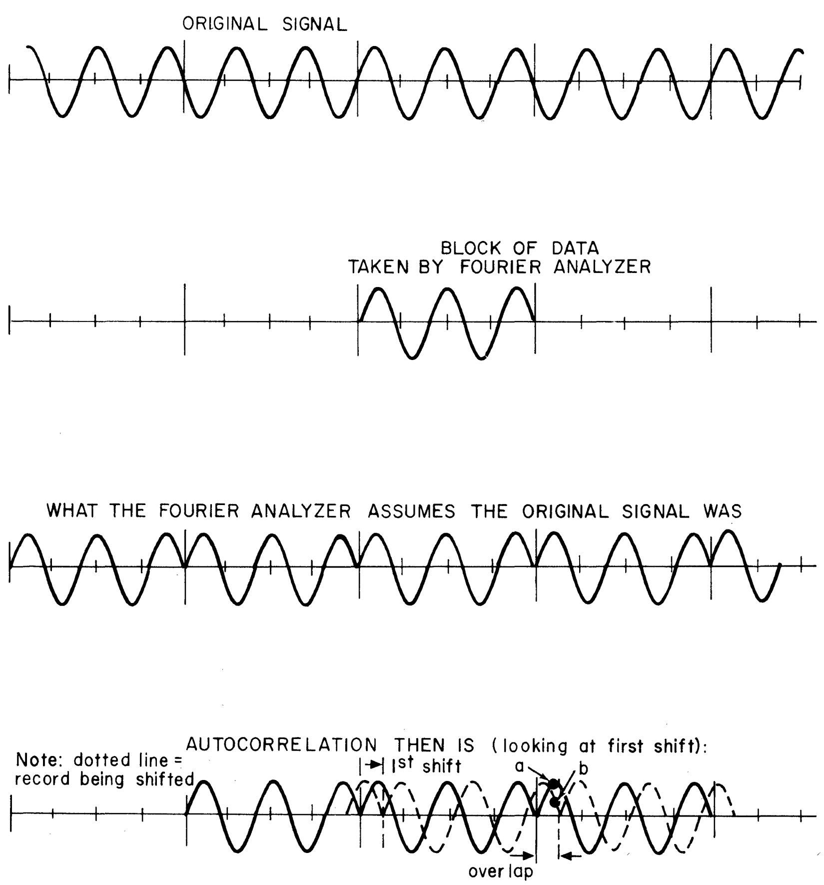
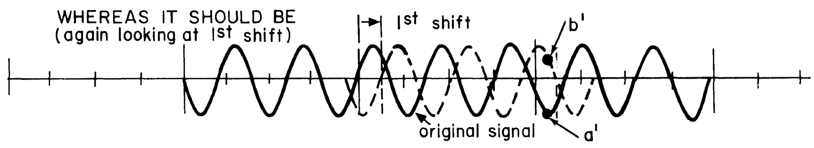
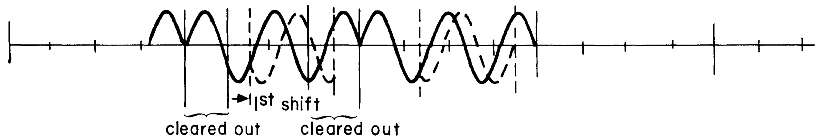


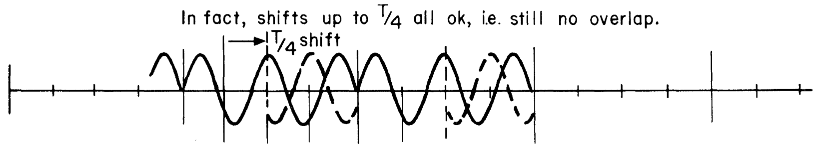
Figure 5-4. Wrap-around Error (Continued)



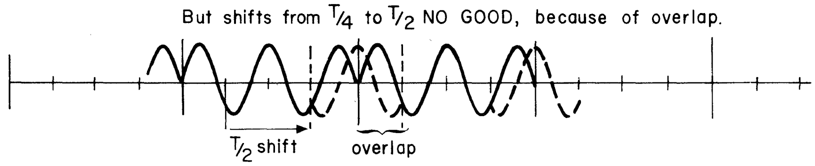
SOLUTION: First clear out T/4 channels on each side of one data block.
First shift then ok, i.e, no overlap with repeated waveform.



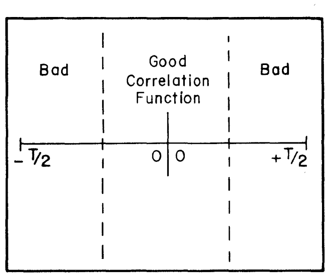
In fact, shifts up to T/4 all ok, i.e. still no overlap.



But shifts from T/4 to T/2 NO GOOD, because of overlap.



THEREFORE, IN FINAL DISPLAY, CLEAR OUT
OR IGNORE +(T/4 to T/2) AND −(T/4 to T/2) CHANNELS



Wrap-around error

CONV Convolution

Convolution like correlation, is an operation done between two blocks of data, therefore by the rule discussed under the general form of the Keyboard command, page 3-10, one of the blocks must be block 0, and only the second block is named in the command. Results are stored and displayed in block 0. Convolution is performed on single precision blocks so that if any of the data blocks involved are double precision, they will be truncated before command is executed.

If the convolution is between block 0 and another block, then the result is a cross convolution. If between 0 and itself, the result is an autoconvolution. The CONVolution command structure is:

CONV [BLOCK N1] ENTER

where:

N1 is any data block. Default value = block 0 (in that case, an autoconvolution will result).

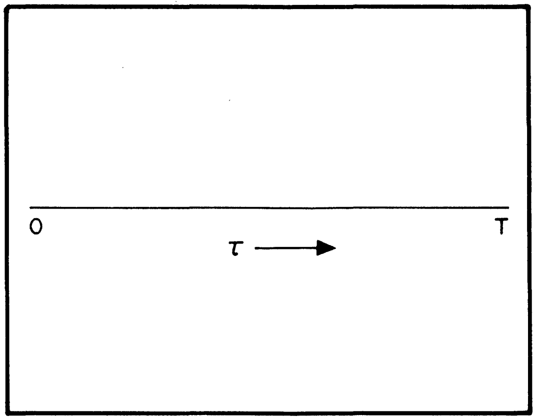
Convolution

The means by which the Fourier Analyzer obtains the convolution function is as follows: First the system takes the Fourier transform of both blocks (cross convolution) or of block 0 (auto convolution). The Fourier Analyzer does a block multiply on both blocks or block 0 with itself. This yields the power spectrum. Finally, the inverse Fourier transform is taken on the power spectrum, to produce the convolution function. Flow graphs of these computations are shown in Figure 5-5.

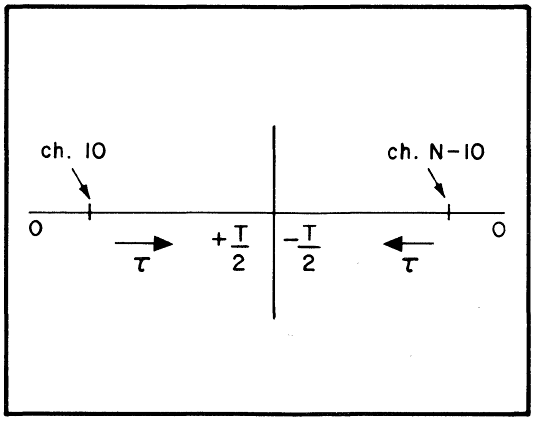
NOTE: When beginning a convolution with data in the frequency domain, it is necessary to multiply the result by the block size to achieve the proper scaling and calibration.

The final convolution function is in the time domain, but with the horizontal axis representing values of τ(lag), not t (time) as originally, over a range of $\pm T/2$. T is the length of the data window. This is shown in the three sets of axes below:

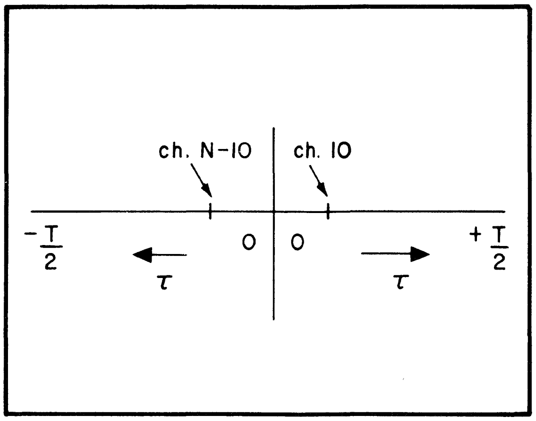
CONV



NORMAL TIME DOMAIN DISPLAY

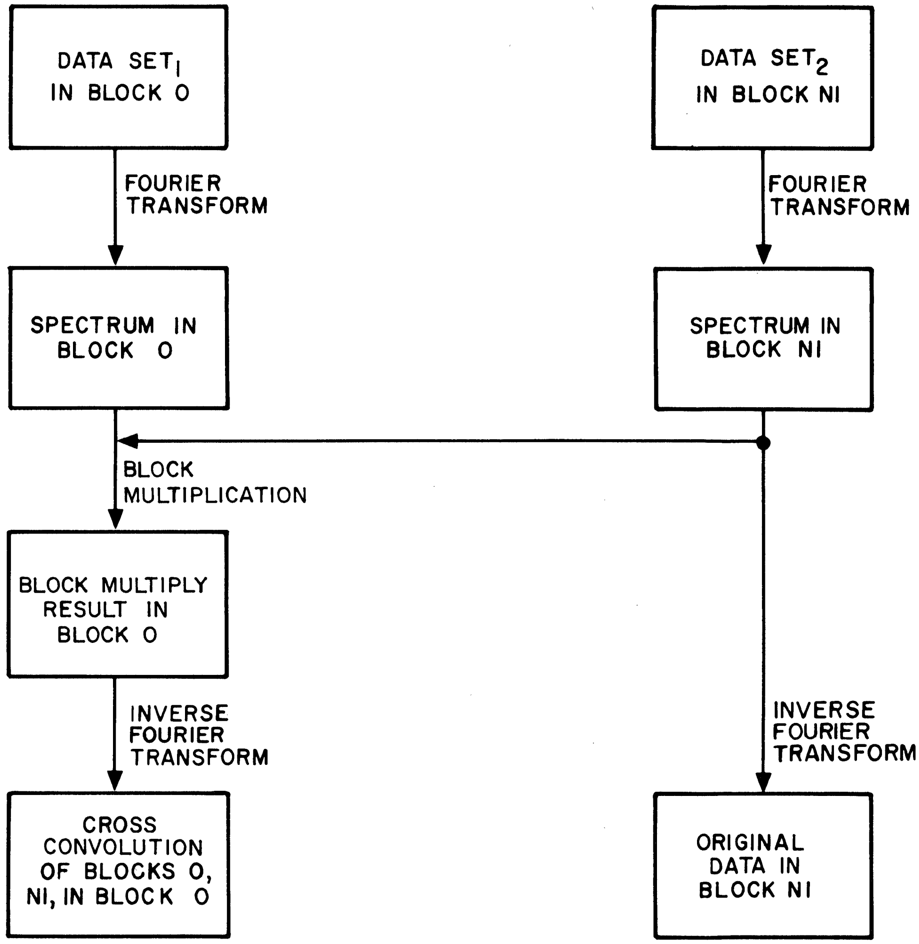


AFTER CONVOLUTION
ORIGIN SWITCH TO LEFT



AFTER CONVOLUTION
ORIGIN SWITCH TO CENTER

Figure 5-5(a). Flow Graph of Cross Convolution Computational Procedure

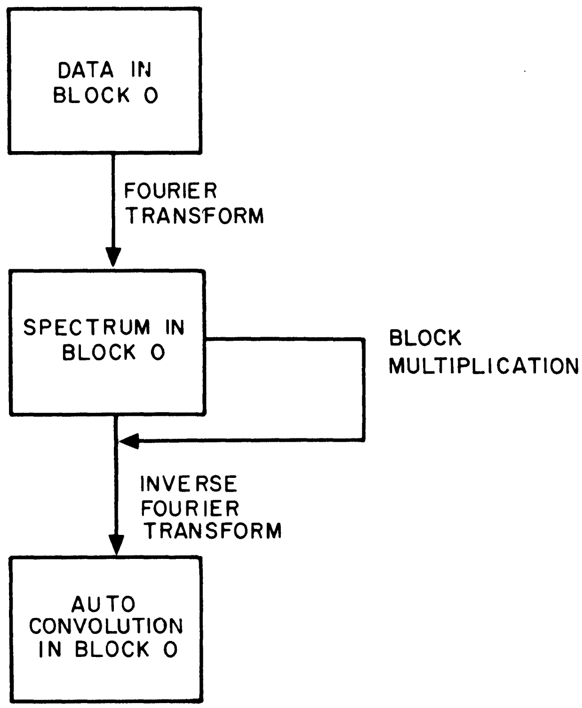


Convolution

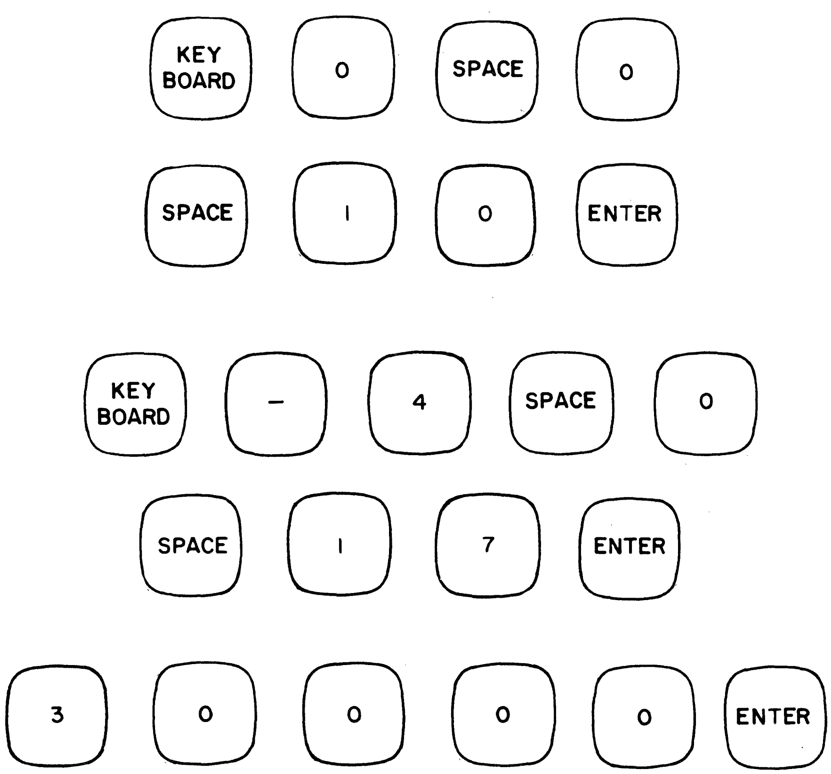
Note that the position of the origin can be switched with the ORIGIN switch on the Display Unit panel. The switch of origins is a display function, not a change in channel locations in the data block. With the ORIGIN switch to LEFT, the channels run from zero on the left to N-1 on the right. Channel 10 is the 10th channel from the left origin, channel N10 is the N-10th channel from the left origin. When the origin is set to CENTER, these channel numbers retain their same relative positions.

Be sure to see the remarks on wrap-around error beginning page 5-27. This error applies to both the convolution and correlation functions.

Figure 5-5(b). Flow Graph of Auto Convolution Computational Procedure



Example 1: Set a block size of 128, then manually create a rectangular pulse in block 0, 11 channels wide with frequency code 17, via the following commands:



Convolution

Take the auto convolution of the pulse by pressing:

CONV | ENTER

The results should be as shown in Figure 5-6(a). Setting the ORIGIN switch to CENTER should give the results shown in Figure 5-6(b).

Example 2: Set the same block size and create the same rectangular pulse as in Example 1. Then transfer the pulse to block 1 by pressing:

STORE | 1 | ENTER

Shift the pulse in block 1, 100 channels to the left:

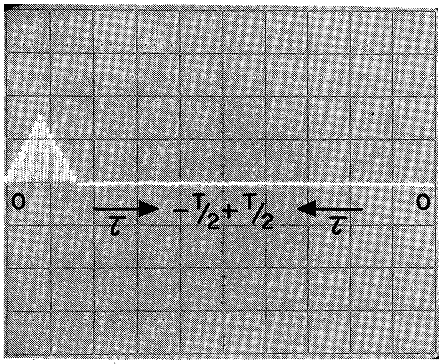
← | 1 | SPACE | 1 | 0 | 0 | ENTER

Take the cross convolution of blocks 0 and 1 by:

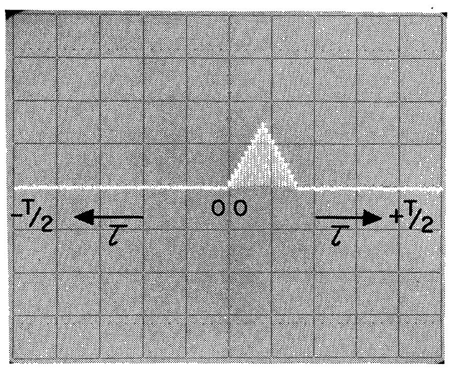
CONV | 1 | ENTER

The result should be as shown in Figure 5-6(c), and indicates a lag between the two signals of 28 channels.

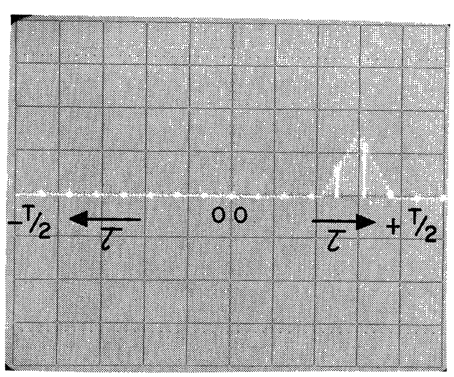
Figure 5-6. Autoconvolution (a and b) and Cross Convolution (c)



(a) ORIGIN switch on LEFT



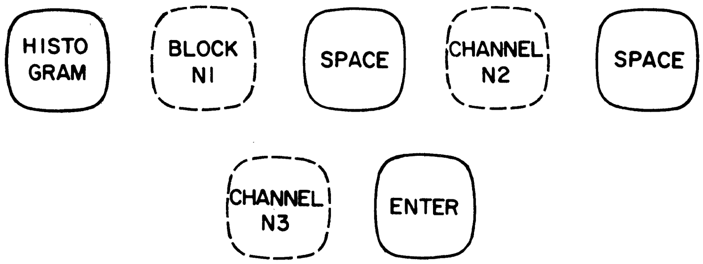
(b) ORIGIN switch on CENTER



(c) After original pulse was shifted 100 channels
(8 point markers)

HISTO GRAM Histogram

This command either histograms all of block 0 or a given range of channels of block 0. The form of the command is:



where:

N1 is the data block into which the histogram is to be entered. *N1 cannot be defaulted or equal to 0.*

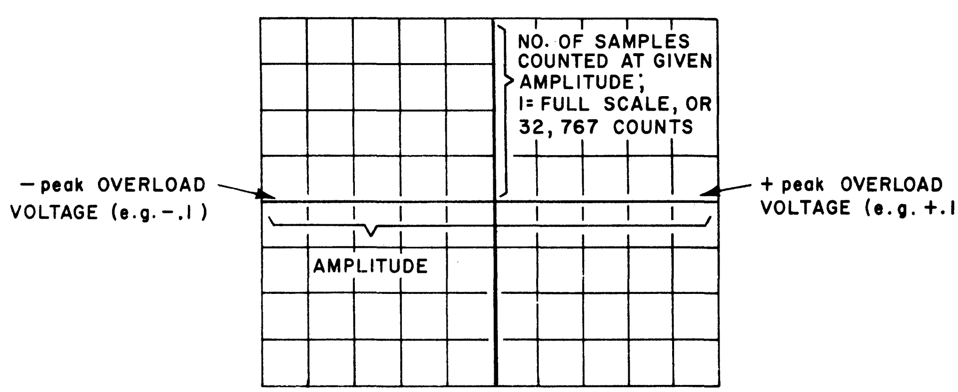
N2 is the channel where the histogram will be begin. Default value is 0.

Histogram

N3 is the channel where the histogram will end. Default value is N2. If N3 is defaulted then channel N2 will be beginning and end of histogram. If both N2 and N3 are defaulted the entire block 0 will be histogrammed.

Block 0 must be in the time domain for histogram operation or a WHAT? signal will result. If initially any value in block N1 is equal to 32,767 then the command will not be executed. After programming, the mode of block N1 will be made time-linear rectangular, its scale factor set to zero and block calibrator set equal to 1. Full scale on the histogram is indicated by the value 1.0 or 32,767 counts.

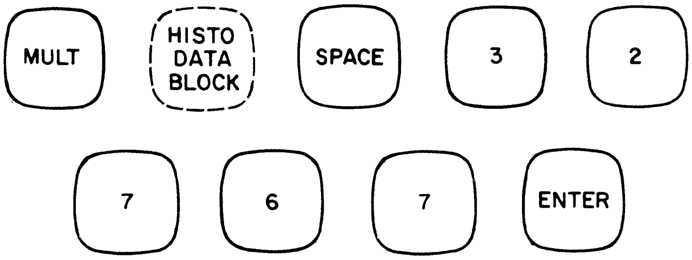
The histogram display is automatically set up as follows:



In the Computer memory, each voltage amplitude is an address. Counts are accumulated at each address until the number at any one address is 32,767. When and if that occurs, the histogram stops automatically, and is redisplayed with the correct scale factor. Further counts are ignored. Full scale on the

HISTO GRAM

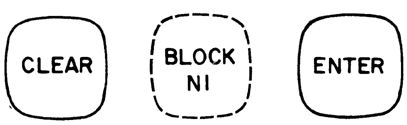
histogram is indicated by the value 1.0. To convert this to actual number of counts, multiply the data block by 32,767 by pressing:

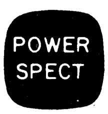


> **Note**
> A Time-Interval Histogram accessory is available for the Fourier Analyzer, consisting of an HP 10618A BCD-Binary Converter and an HP 5325B or a 5326A/B Counter, plus interfaces.

Power spectrum

Since the histogram command does not automatically clear the data block before accumulating a new set of values, the user must be certain to do this himself, unless additional values are to be collected on an old histogram. To clear the block, press:



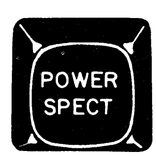


Power Spectrum

Power spectra are obtained in general by performing the following operations:

1. Taking N samples of the data (i.e., one record) and storing it in a data block.
2. Doing a Fourier transform on this data.
3. Conjugate multiplying the transform by itself (auto power spectra, i.e., input and output) or by another transform (cross power spectrum, i.e. input x output).

The POWER SPECTrum key is used to compute the input, output, and cross power spectra with a minimum of user programming. (Flow graphs of these computations are shown in Figures 5-7 and 5-8.) A comparison of a power



spectrum summation average program with and without the POWER SPECTrum command is shown below:

Power spectrum

Typical Program

```
 1 CL 3
 4 CL 4
 7 CL 5
10 L  1
13 RA 1
16 F  1   2
20 X< 1
23 *-
25 A+ 3
28 X> 3
31 X< 2
34 *-
36 A+ 4
39 X> 4
42 X< 2
45 *- 1
48 A+ 5
51 X> 5
54 #  1 100
59 :  3 100
63 :  4 100
67 :  5 100
71 .
```

Steps 20 through 51: Replaced by POWER SPECT command

Program using POWER SPECTrum Key

```
 1 CL 3
 4 CL 4
 7 CL 5
10 L  1
13 RA 1
16 F  1   2
20 SP 1   2
24 #  1 100
29 :  3 100
33 :  4 100
37 :  5 100
41 .
```

At conclusion of both programs:

Block 3 contains input power spectrum summation average
Block 4 contains output power spectrum summation average
Block 5 contains cross power spectrum summation average

It is evident from the above comparison that only half as many program steps are required using the POWER SPECTrum key, thus conserving program storage as well as reducing programming steps.

The POWER SPECTrum command format is:

[POWER SPECT] [N1] [SPACE] [N2] [ENTER]

where:

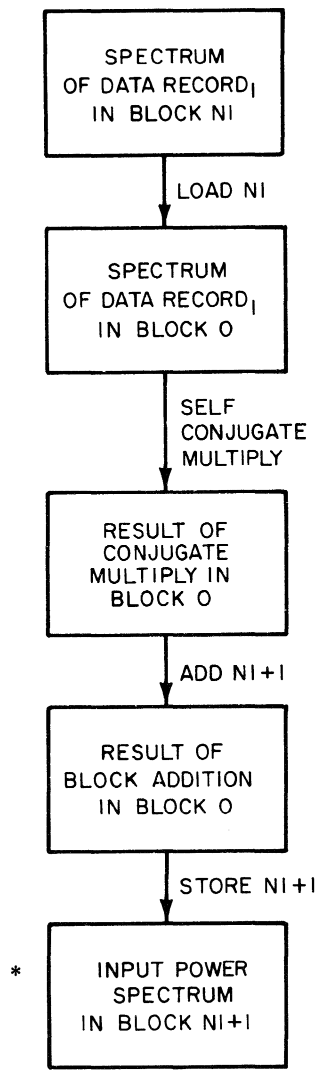
N1 is the first (input) data block and *cannot be defaulted* or equal zero for dual channel operation.

N2 is equal to 2 for dual channel input and default of N2 specifies single channel input.

Results	Location of Results	
	Single Channel Input	Dual Channel Input
Input Power Spectrum	N1 + 1	* N1 + 2
Output Power Spectrum	Not Executed	* N1 + 3
Cross Power Spectrum	Not Executed	* N1 + 4

*If block 0 is specified as N1 for dual channel input a WHAT? signal will result.

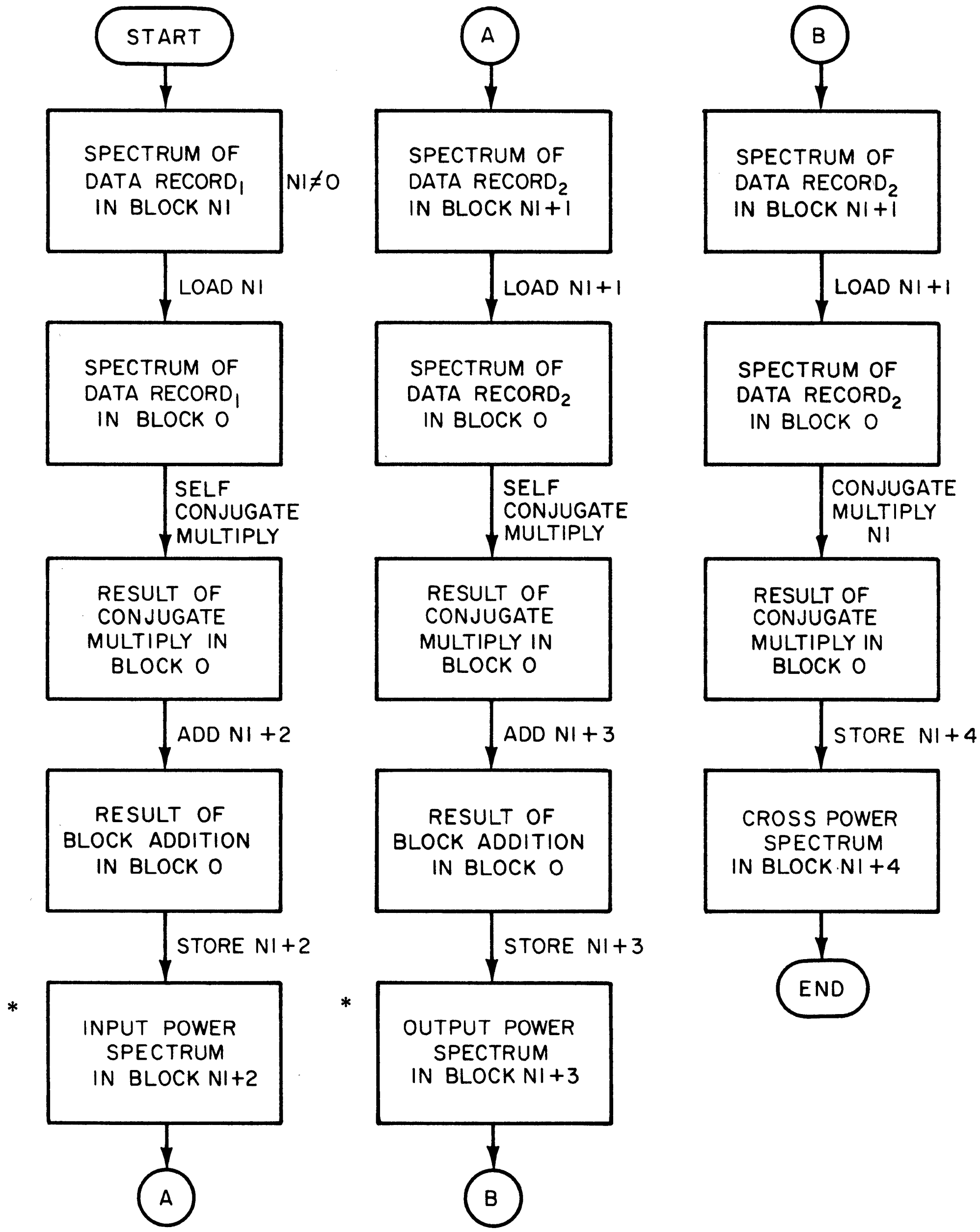
Figure 5-7. Flow Graph of Power Spectrum Computational Procedure for Single Channel Input



*Input power spectrum data block will be in double precision assuming block N1 is in the frequency domain.

Power spectrum

Figure 5-8. Flow Graph of Power Spectrum Computational Procedure for Dual Channel Input



*The input and output power spectrum data blocks will be in double precision assuming blocks N1 and N1+1 are in the frequency domain.

TRANS FCN

Transfer Function

A transfer function is a mathematical description of a system, be that system a filter, a jet engine, a vibrating airplane wing, an organ in the human body, or whatever. The transfer function can also be used to measure the relationship between any two signals. It can be defined as:

$$\text{transfer function} = \frac{\text{Fourier transform of output}}{\text{Fourier transform of input}}$$

or equivalently,

$$\text{transfer function} = \frac{\text{average cross power spectrum of input and output}}{\text{average power spectrum of input}}$$

The coherence function measures the degree of causality between any two signals. It can therefore be used to check the validity of the transfer function. When a transfer function is computed, we may not be aware of extraneous inputs or whether or not the system is linear. Both of these factors would introduce error in the computed transfer function. The coherence function ranges between 0 and 1. Zero means no coherence between input or output, or in other words, extraneous inputs and/or the system is non-linear; 1 means complete coherence between input and output, or in other words only one input and a linear system.

Transfer function

The equation for calculating the coherence function is

$$Y^2 = \frac{|\overline{G_{YX}}|^2}{\overline{G_{XX}} \cdot \overline{G_{YY}}}$$

where

$|G_{YX}|^2$ = square of the magnitude of the cross spectrum

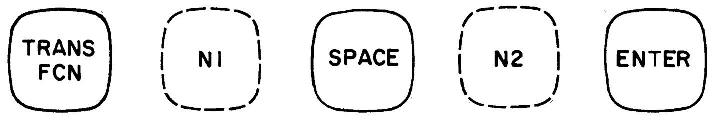
$\overline{G_{XX}}$ = input auto spectrum

$\overline{G_{YY}}$ = output auto spectrum

The TRANSfer FunCtioN key is used to compute the transfer and coherence functions with a minimum of user programming. It assumes dual channel operation, and that the auto and cross power spectrums have previously been

TRANS FCN

calculated using the POWER SPECTrum key. (For a flowgraph of transfer and coherence function, see Figure 5-9). The TRANSfer FunCtioN command structure is:



where:

N1 is the first input data block and *cannot be defaulted* or equal to zero since dual channel operation is assumed.

N2 is equal to 2 and specifies dual channel input, *default of N2 is not allowed.*

Results	**Location of Results**
Transfer Function	* N1
Coherence Function	* N1 + 1

Transfer function

*If block 0 is specified as N1, a WHAT? signal will result.

A comparison of a transfer and coherence function program with and without the POWER SPECTrum and TRANSfer FunCtioN commands is shown below.

Typical Program

```
 1 CL  3
 4 CL  4
 7 CL  5
10 L   1
13 RA  1
16 F   1   2
20 X<  1
23 *-
25 A+  3
28 X>  3
31 X<  2
34 *-
36 A+  4
39 X>  4
42 X<  2
45 *-  1
48 A+  5
51 X>  5
54 #   1 100
59 X<  5
62 :   3
65 X>  1
68 X<  5
71 *-
73 :   3
76 :   4
79 X>  2
82 .
```

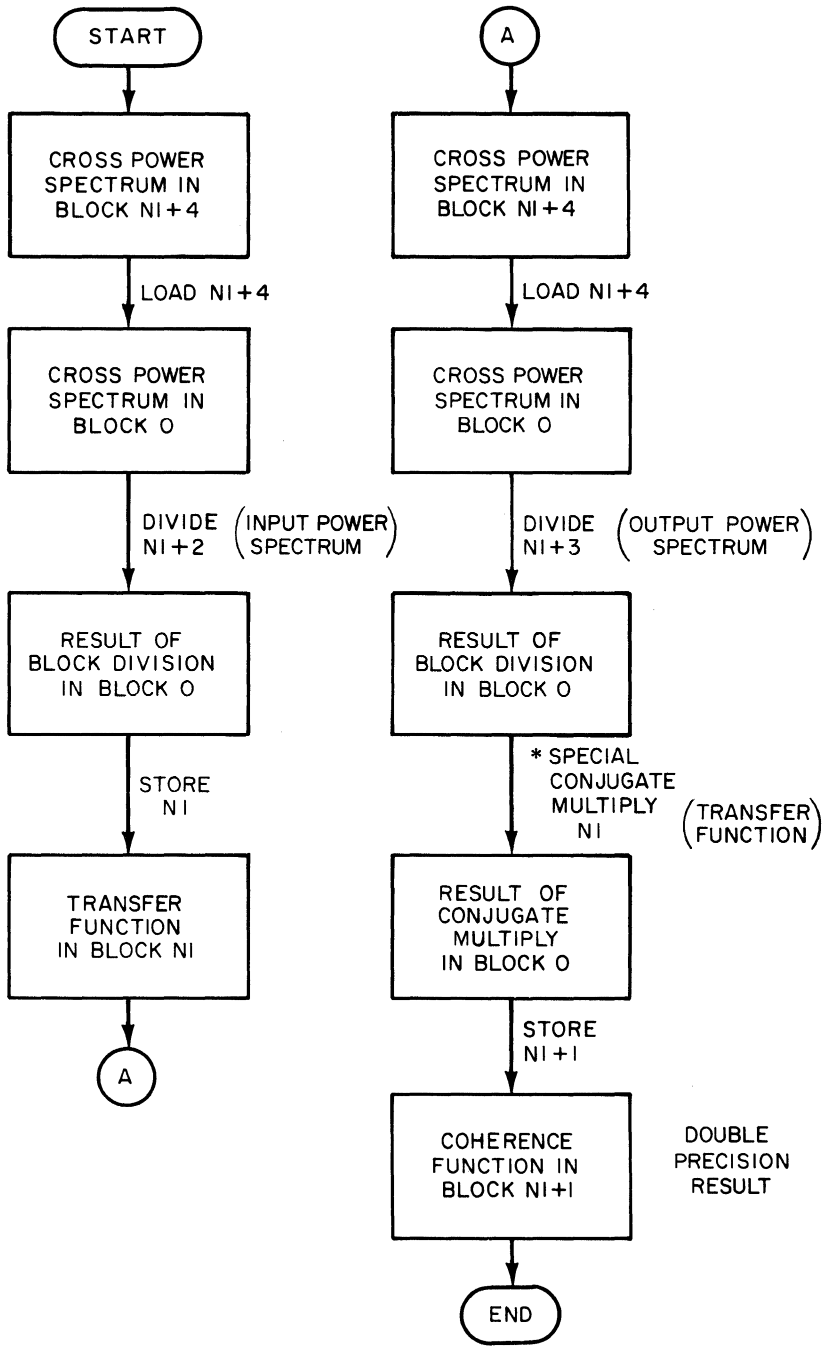
Program Using POWER SPECTrum and TRANSfer FunCtioN Keys

```
 1 CL  3
 4 CL  4
 7 CL  5
10 L   1
13 RA  1
16 F   1   2
20 SP  1   2
24 #   1 100
29 CH  1   2
33 .
```


It is evident from the comparison above that programming is considerably simplified, and program storage conserved, when using the POWER SPECTrum and TRANSfer FunCtioN keys.

Figure 5-9. Flow Graph of Transfer and Coherence Computational Procedures

Transfer function



*Since the coherence function is completely real, a special conjugate multiply is performed with the result being a double precision block.

SECTION VI

OUTPUT MODES

ABOUT THIS SECTION

This section provides operating instructions for the following output modes: print (Teleprinter), punched tape, scope display and plotter. The print and scope display outputs are standard on all Fourier Analyzers. Punched tape and plotter are options.

PRINTED DATA OUTPUT VIA TELEPRINTER

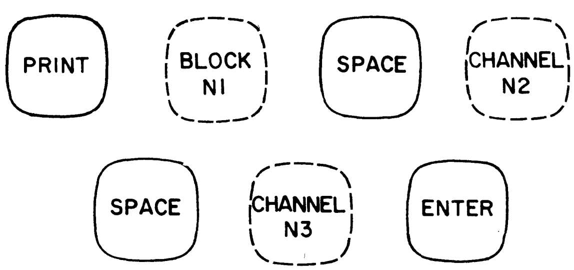
How Done

The 2752A Teleprinter is used to obtain a printed copy of the data in any block, or in any part of a data block. The data is printed in the decimal number system. If the system contains a High Speed Punch, this data can be punched on paper tape as well. Refer to the table on output device selection, page 6- 8 .

Print Command

The form of the command for printing out data is:

Print command



where:

N1 is the data block to be printed out. (See table below for default values.)

N2 is the first channel to be printed out.

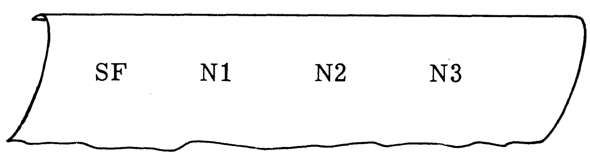
N3 is the final channel to be printed out.

Default Values for PRINT Command

Element	Meaning of Element	Default Value of Element
N1	data block to be printed out	data block 0
N2*	starting channel of print-out	whole data block is printed out
N3*	end channel of printout	N2
*If N2 and N3 are defaulted whole data block is printed.		

Format of Printed Out Data

Each printout begins with a line giving all the calibration factors associated with the data. The first item in the line is SF (meaning scale factors) and the form of the line is shown below:



Print command

where:

N1 is the amplitude scale factor for linear data (each word is multiplied by 10^{N1}); or the value in dB to be added to each data word, for logarithmic data.

N2 is the coordinate code from the table below.

Coordinate Code	Time	Frequency	Rectangular	Polar	Log	Linear	Single Precision	Double Precision
0	x		x			x	x	
2	x		x		x		x	
4		x	x			x	x	
5		x		x		x	x	
7		x		x	x		x	
12		x	x			x		x
14		x	x		x			x

N3 is the frequency code from the following table.

Freq. Code	F_{max} (MAX FREQ)	Δt (Δ TIME)	Freq. Code	Δf (Δ FREQ)	T (TOTAL TIME)
47	50 kHz	10 μsec	63	100 Hz	10 msec
46	25 kHz	20 μsec	62	50 Hz	20 msec
45	10 kHz	50 μsec	61	20 Hz	50 msec
44	5 kHz	100 μsec	60	10 Hz	100 msec
43	2.5 kHz	200 μsec	59	5 Hz	200 msec
42	1 kHz	500 μsec	58	2 Hz	500 msec
41	.5 kHz	1000 μsec	57	1 Hz	1000 msec
40	.25 kHz	2000 μsec	56	.5 Hz	2000 msec
39	0.1 kHz	5000 μsec	55	.2 Hz	5000 msec
15	50 Hz	10 msec	31	100 mHz	10 sec
14	25 Hz	20 msec	30	50 mHz	20 sec
13	10 Hz	50 msec	29	20 mHz	50 sec
12	5 Hz	100 msec	28	10 mHz	100 sec
11	2.5 Hz	200 msec	27	5 mHz	200 sec
10	1 Hz	500 msec	26	2 mHz	500 sec
9	0.5 Hz	1000 msec	25	1 mHz	1000 sec
8	0.25 Hz	2000 msec	24	.5 mHz	2000 sec
7	0.1 Hz	5000 msec	23	.2 mHz	5000 sec

Print command

Examples of typical printouts are given in Figure 6-1.

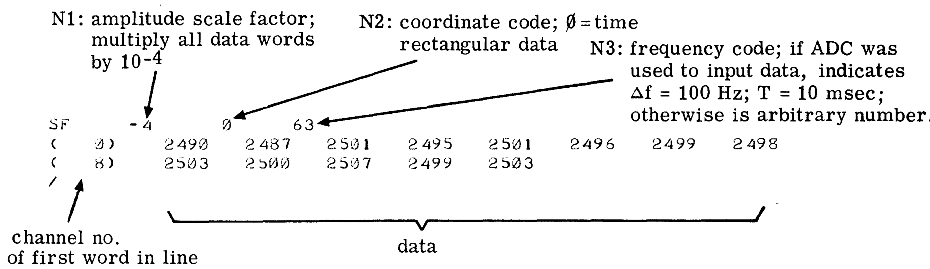
At the start of each printout line, the channel number of the first data word in the line is given in parentheses. If the printout is of time data, eight channels of data per line are printed. If the printout is of frequency data, only four channels per line are printed. For each frequency channel, first the real or magnitude word is printed (depending on coordinates), then the imaginary or phase word. The channel number at the start of each line will change by eight for each line of time data, and four for each line of frequency data. Log printouts are in hundredths of a dB. The scale factor (N1) must be added to each dB value. Phase is printed out in hundredths of a degree.

To stop a printout while it is in process, simply press the STOP key on the Keyboard. The Teleprinter will finish its current line and stop.

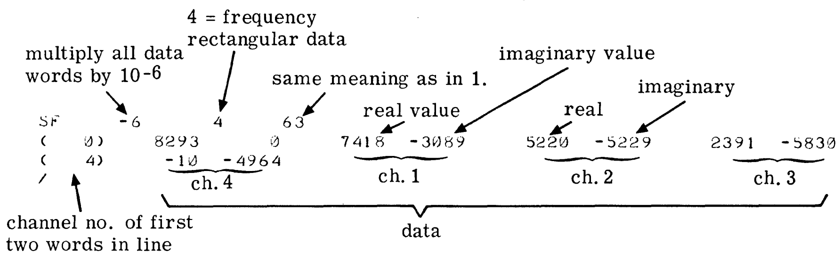
Print command

Figure 6-1. Examples of Printouts

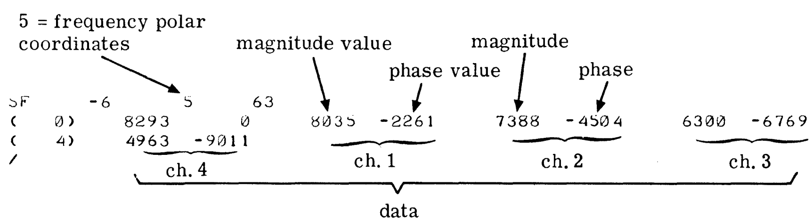
1. Time domain data



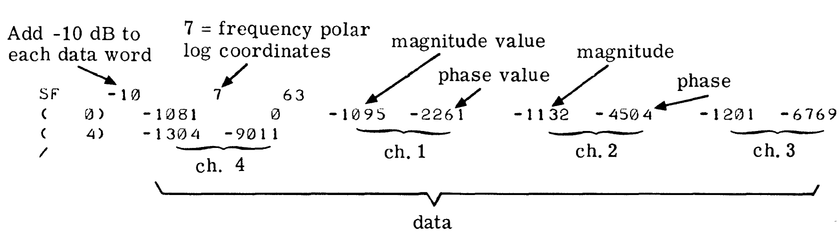
2. Frequency domain data, rectangular coordinates



3. Frequency domain data, polar coordinates (phase in .01 degree)



4. Frequency domain data, polar log coordinates (magnitude in .01 dB; phase in .01 degree)



PUNCHED TAPE OUTPUT

How Done

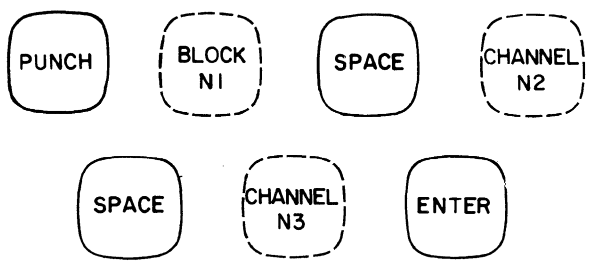
Paper tapes can be punched either by the Teleprinter or by the optional High Speed Punch. The optional punch has the advantage of speed in punching large amounts of data, while the Teleprinter has the attraction of printing out the same data being punched.

PUNCH

Punch Command

The command to punch a paper tape is given below. If the system has a High Speed Punch, place power switch to ON. If not, set the Teleprinter switch to LINE, press the ON button of the Teleprinter punch, and press to light switch 4 on the Computer.

The command form is:



Punch command

where:

N1 is the data block to be punched out.

N2 is the first channel to be punched out.

N3 is the last channel to be punched out.

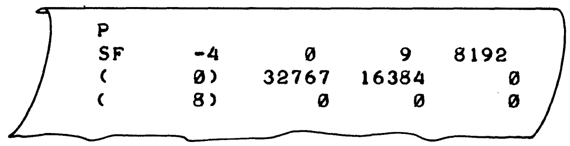
Following table has default values.

Element	Meaning of Element	Default Value of Element
N1	data block to be punched out	data block 0
N2*	first channel to be punched out	whole data block is punched out
N3*	last channel to be punched out	N2
*If N2 and N3 are defaulted, entire data block will be punched out.		

Data Format of Punched Tape

The data format on any tape produced by the PUNCH command (via High Speed Punch or Teleprinter) is shown in Figure 6-2.

Figure 6-2. Data format on punched tape



SF — stands for "scale factors"

-4 — is "k" in the expression "10^k". All data words are multiplied by 10^k, giving the number system called "floating point on a block basis". Thus, in the above example, all data words are multiplied by 10^{-4} = 0.0001.

Ø — is coordinate code from the following table. In above example, data is in time domain, rectangular coordinates.

Punch command

Coordinate Code	Time	Frequency	Rectangular	Polar	Log	Linear	Single Precision	Double Precision
0	x		x			x	x	
2	x		x		x		x	
4		x	x			x	x	
5*		x		x		x	x	
7*		x		x	x		x	
12		x	x			x		x
14		x	x		x			x

*With polar data, the coordinate code can be 16, 32, or 48 larger than shown to indicate polarity of dc and F_n points. That is, binary 16 greater indicates F_n negative, 32 indicates dc negative, and 48 indicates both points negative.

9 — frequency code from following table, which expresses data sampling parameters in terms of SAMPLE MODE and MULTIPLIER switch settings on the ADC. In the above example, pertinent data sampling parameters were F_{max} of 0.5 Hz, Δt of 1000 msec.

8192 is block calibrator which multiplies all data words in block. To convert calibrator into actual physical values see explanation on page 4-12.

Figure 6-2. Data Format on Punched Tape (Continued)

Freq. Code	F_{max} (MAX FREQ)	Δt (Δ TIME)	Freq. Code	Δf (Δ FREQ)	T (TOTAL TIME)
47	50 kHz	10 μsec	63	100 Hz	10 msec
46	25 kHz	20 μsec	62	50 Hz	20 msec
45	10 kHz	50 μsec	61	20 Hz	50 msec
44	5 kHz	100 μsec	60	10 Hz	100 msec
43	2.5 kHz	200 μsec	59	5 Hz	200 msec
42	1 kHz	500 μsec	58	2 Hz	500 msec
41	.5 kHz	1000 μsec	57	1 Hz	1000 msec
40	.25 kHz	2000 μsec	56	.5 Hz	2000 msec
39	0.1 kHz	5000 μsec	55	.2 Hz	5000 msec
15	50 Hz	10 msec	31	100 mHz	10 sec
14	25 Hz	20 msec	30	50 mHz	20 sec
13	10 Hz	50 msec	29	20 mHz	50 sec
12	5 Hz	100 msec	28	10 mHz	100 sec
11	2.5 Hz	200 msec	27	5 mHz	200 sec
10	1 Hz	500 msec	26	2 mHz	500 sec
9	0.5 Hz	1000 msec	25	1 mHz	1000 sec
8	0.25 Hz	2000 msec	24	.5 mHz	2000 sec
7	0.1 Hz	5000 msec	23	.2 mHz	5000 sec

Punch command

32767 16384.. are data words. Data word system is as follows: 0 stands for 0; 32767 stands for 1.

To convert data words into actual physical values:

Use the following formula:

$$\frac{\text{data word}}{32{,}767} \times \frac{\text{block calibrator}}{32{,}767} \times 10^{k}$$

Thus, in the above example, actual physical value of first data word, 32767 is

$$\frac{32767 \times 8192}{(32767)^2} \times 10^{-4} = .25 \times 10^{-4}$$

Actual physical value of second data word is:

$$\frac{16384 \times 8192}{(32767)^2} \times 10^{-4} = .125 \times 10^{-4}$$

OUTPUT DEVICE SELECTION

The PUNCH command assumes that data will be punched on the optional High Speed Punch. If this device is not part of the system, the PUNCH command will apply to the Teleprinter punch, but only if bit 4 of the Computer DISPLAY REGISTER is lighted. The LIST and PRINT commands are also affected by the state of bit 4. The following table shows the effect of bit 4 on the output device selection for these commands.

Command	Bit 4 = 0	Bit 4 = 1
PUNCH	High Speed Punch	Teleprinter Punch
LIST	Teleprinter	High Speed Punch
PRINT	Teleprinter	High Speed Punch

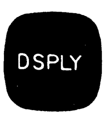
Note: Bit 3 affects *input* conditions, see Section IV for similar table.

SCOPE DISPLAY OUTPUT

The 5460A Display Plug-in and the H51-180AR/DR Oscilloscope form the Display Unit of the Fourier Analyzer. The purpose of the Display Unit is to convert digital data in the Computer memory to analog signals that may be displayed on the 8 x 10 cm screen of the scope, or may be sent to an external X-Y scope, or plotted on a standard X-Y plotter. Control of system operations connected with the display—such as digital expansion and log horizontal scale—are provided on the Display Unit.

Display command

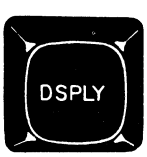
It is important to note that any operation called for by the Display Unit does not change the form or coordinates of the data in the memory, but only modifies the data before it is transmitted to the Display Unit. Thus, display operations do not cause computational errors to build up. Only operations initiated by the Keyboard cause changes in data form in the memory.



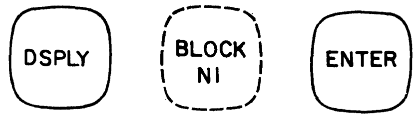
Display Command

There are three types of display commands in the Fourier Analyzer: **block display**, a repeat calibrated display in which the entire block is displayed; **partial block display**, a repeat calibrated display in which only a given number of channels are displayed across the entire face of the scope; and a **coded display**, where combinations of single/repeat, calibrated/uncalibrated displays are possible.

If block N1 is in double precision, the least significant bits will not be shown, i.e., the imaginary part of block N1 will appear equal to zero.



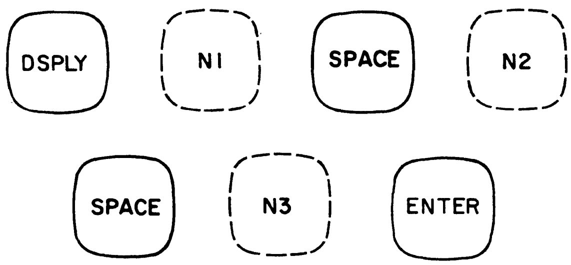
The command for block display is:



where:

N1 is the data block. Default value = block 0.

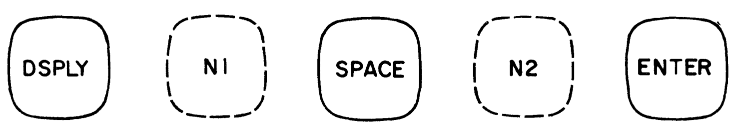
The command for partial block display is:



where:

N1 is the data block. *No default allowed.*
N2 is the first channel to be displayed. *No default allowed.*
N3 is the last channel to be displayed. *No default allowed.*

The command for coded display is: (This command always displays an entire data block.)



where:

N1 is the data block. *No default allowed.*

N2 is display code. Default value = 0 (calibrated repeat).

Display Code (N2 Value)	Meaning of Code
0	Calibrated repeat sweep display
4	Calibrated single sweep display
8	Uncalibrated repeat sweep display
12	Uncalibrated single sweep display

The Fourier Analyzer operates on an automatic display concept, that is, the result of an operation is displayed. There is one exception: the ANALoG IN command, where the block displayed depends on the block named in N2 of the command, or in the REPEAT mode on the setting of the DISPLAY/INPUT switch of the ADC.

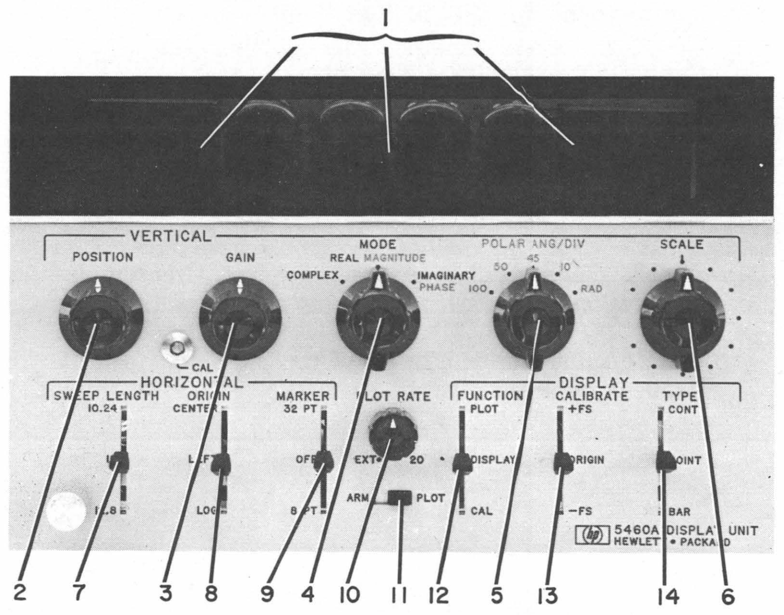
If a repeat sweep display command is executed from the program memory, the program will display and stop, until ordered to continue by pressing the CONTINUE key or by another command. However, if a single sweep display command is used the display will take one sweep of the block specified and then continue on with the program.

Display Unit Controls

Front panel controls of the 5460A Display Unit are explained in Figure 6-3.

Display command

Figure 6-3. 5460A Display Unit



1. **Scale Factor:**

for linear scales - the value of each vertical division on the scope. For example, a scale factor of 5 x 10 -005 means that each vertical division is 5 x 10^{-5} or 0.00005 volts, or volts squared (the user must know the units of the display he is looking at).

The scale factor for a linear amplitude scale will always be correct if the GAIN control is in the CAL position. The Fourier Analyzer will automatically adjust the digital output to give the best possible resolution with no display overflow. When the scale factor is not correct during certain special situations (such as the analog-in REPEAT mode), the scale factor display will be blanked out.

for log scales - the value, in dB referenced to 1 volt, of the top line of the scope display; in other words, the offset.

domain and coordinate readout - on the right of the scale factor display, the domain of the data displayed (time or frequency) and the coordinates are given.

FREQ: means that the horizontal axis has units of frequency, with dc on the left edge (the left edge is channel zero), and maximum frequency (F_{max}) on the right. Each point is then Δf apart, the second point from the left being 1Δf, the third 2Δf, etc. *When the FREQ light is out, it means that the horizontal axis has units of time.*

RECT: indicates that vertical axis is real (cosine) or imaginary (sine), depending on setting of MODE switch.

POLAR: indicates that vertical axis is magnitude or phase, depending on setting of MODE switch.

LOG: indicates that vertical axis is logarithmic. The scale factor display in this case gives the value of the top line of the display in dB with reference to 1 volt. (For example, a scale factor display of -020 in LOG means that the top line of the display is -20 dB.) Each vertical division is equal to 10 dB.

2. **POSITION:** adjust vertical position of display. Used as part of calibration procedures.

3. **GAIN:** expands display vertically. When in CAL position, scale factor will always be kept correct by Fourier Analyzer. This control allows the vertical scale to be expanded up to 2 times. It functions for all types of displays, and is a continuous expansion of the scope screen display only (i.e., does not change scale factors and has no effect on external scope or plotter). This control should be generally left in the CAL position so the display will always be calibrated in accordance with the scale factor display.

4. **MODE:**

COMPLEX: displays a complex, or Nyquist, plot of a spectrum, in which the horizontal axis is the real part, and the vertical axis the imaginary part of the spectrum. Each point then represents a frequency. The distance to each point from the origin is the magnitude, and the angle of the magnitude from the positive horizontal axis is the phase. Since this display requires an equal scale along each axis, the 8 x 10 cm scope display is automatically changed to an 8 x 8 cm display when the MODE switch is put in the COMPLEX position. The center is then 1 cm to the left of the center vertical axis on the scope screen.

REAL/MAGNITUDE: displays real part of spectrum if scale factor display reads POLAR. Displays log magnitude if scale factor display reads LOG. For time functions, this position will display the real part of the time series.

IMAGINARY/PHASE: displays imaginary part of spectrum if scale factor display reads FREQ and RECT. Displays phase if scale factor display reads POLAR. In this case, the vertical scale is determined by the setting of the POLAR ANG/DIV switch, with the numbers representing degrees per division. The numeric part of the scale factor display will be blanked out when the display reads POLAR. For time functions, this position will display the real part of the time series, since the imaginary part does not exist.

5. **POLAR ANG/DIV:** sets number of degrees (angle) per vertical division when scale factor display reads POLAR and MODE switch is in PHASE position.

6. **SCALE:** This is a digital expander of the vertical scale. In the upright position, makes scale factor display automatic, so that Fourier Analyzer sets the scale factor for best resolution without overflow. When switch is turned clockwise, scale expands. When counterclockwise, scale compresses.

This switch has two principal uses: first, it expands the display vertically over a 500:1 (2.3 decade) range, for examination of small details near zero, when large values are also present. Second, it overrides the automatic scale factoring of the Fourier Analyzer when taken off the vertical position so that two different displays, whose scale would be otherwise automatically changed, can be photographed or plotted on the same scale. When display is logarithmic, each position off the vertical adds +4 dB to the display. Thus the switch can be used to set up a non-fractional decade scale (i.e., a top line value of -16 dB can be switched to -20 dB, etc.). At the same time, the display is shifted up 4 dB to maintain calibration. This enables the user to look at the data more than 80 dB below maximum value.

The SCALE switch does not function during a phase display. It does affect the display on external plotters and scopes. Keep in mind that the external device plots or displays what is on the scope screen.

7. **SWEEP LENGTH:**

10: sweep is completed in 10 horizontal divisions of scope screen. This will, however, produce a fractional number of channels per division, which the 10.24 and 12.8 positions avoid (e.g., with data block size of 512, number of channels per division = 512/10 = 51.2).

10.24: expands sweep to 10.24 horizontal divisions. Part of the sweep will now be off screen, but number of channels per division on screen can be made non-fractional in this, or failing this, the 12.8 position (e.g., with data block size of 512, number of channels per division = 512/10.24 = 50).

12.8: expands sweep to 12.8 horizontal divisions. Part of sweep will not be off screen, but number of channels per division in this, or failing this, the 10.24 position, can be made non-fractional (e.g., with data block size of 512, number of channels per division = 512/12.8 = 40).

Note: sweep expansion has the same effect on an external plotter as it does on the scope.

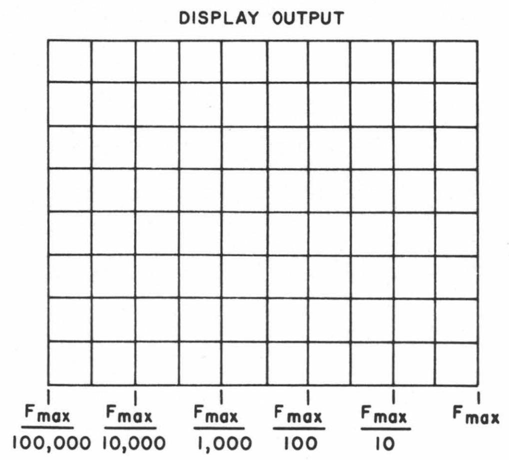
8. **ORIGIN:**

LEFT: puts horizontal origin on the left edge of the scope screen, with channel 0 at the left edge, and channel N-1 at the right edge. Sweep moves from channel 0 to channel N-1.

CENTER: channel 0 is in center of screen, channel N/2 at the right edge, channel N/2 + 1 at left edge, channel N-1 is immediately left of channel 0. Sweep moves from channel 0 to the right edge, then in from the left edge to channel N-1. This mode of sweep is useful in certain cases where display of correlation or convolution is desired. This mode does not function in the frequency domain, where the origin is aways on the left edge.

LOG: sets a logarithmic horizontal scale. The conversion for this scale is done digitally, so there are no conversion errors. The number of decades contained in a display is determined by the block size. For example, a block size of 128 has 64 frequency points and thus has less than 1 decade of log horizontal display. A block size of 512 has 256 frequency points and thus has 2½ decades of log horizontal display.

The horizontal log scale is illustrated below.



9. **MARKER:**

32 PT: gives an intensity marker every 32 channels on the horizontal axis, counting origin as channel 0.

OFF: no markers.

8 PT: gives a marker every 8 channels on the horizontal axis, counting origin as channel 0.

10. **PLOT RATE:**

EXT: allows "seek command" from Fourier Analyzer to tell external X-Y plotter to plot a point. Will not give next point until "completed plot" signal for previous point received from plotter.

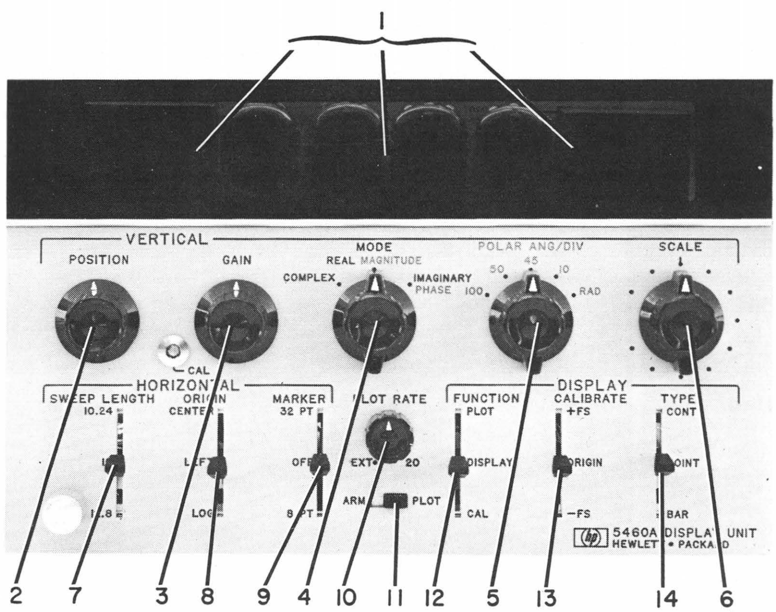
"internal" (i.e., off EXT position): external X-Y plotter plots points at rate set by Fourier Analyzer. This control varies that rate from 5 points/sec (counterclockwise position) to 300 points/sec (clockwise position). Fourier Analyzer does not wait for "plot completed" signal from external plotter before sending next point.

11. **ARM-PLOT:**

ARM: outputs first channel to X and Y outputs of external X-Y plotter, and, if the SERVO signal is being used to control the plotter servo enable, it turns on the servo. If the plotter servo drive is manually controlled, it should be turned on when switch is set to ARM. Plotter moves to first point to be plotted.

PLOT: outputs X and Y points to be plotted by external plotter. Note: ARM position *must* precede PLOT position.

Figure 6-3. 5460A Display Unit (Continued)



12. FUNCTION:

PLOT: makes display output from Fourier Analyzer available to external X-Y plotter, via connectors at rear of Display Unit. Shuts off scope display.

DISPLAY: makes display output from Fourier Analyzer available at scope screen. Turns off servos in external X-Y plotter.

CAL: puts calibration dot at one of three positions on scope screen, as set by CALIBRATE switch.

13. CALIBRATE:

ORIGIN: allows calibration dot to be centered on horizontal axis, left edge of screen, using VERTICAL POSITION control on Display Plug-in panel, and HORIZONTAL POSITION control on scope panel.

+FS: allows calibration dot to be centered at mid top line of screen, using screwdriver adjustment on vertical GAIN CAL control.

-FS: allows calibration dot position to be checked (should be at mid-bottom line of screen) after ORIGIN and +FS adjustments. This position can be used for GAIN control screwdriver adjustment, and the +FS used for check,if desired.

No horizontal gain calibration is provided. The basic stability of the display will not allow the horizontal gain to drift an amount greater than the resolution of the oscilloscope. In addition, the horizontal markers eliminate the need for precise horizontal gain (to identify a frequency point, one simply has to count the dots and multiply by Δf).

Calibration examples are on page 6-13.

14. TYPE:

CONT: display will be a continuous line. Greatest usefulness is in time domain and log displays.

POINT: display will be a series of points. Greatest usefulness is for linear displays in frequency domain.

BAR: display will be a series of vertical bars. Useful for linear displays in frequency domain.

Example of Calibrating the Scope for Normal Use

To calibrate the scope for normal plus and minus readings (as opposed to full scale positive-only readings, covered in the next example), make the following settings:

FUNCTION to CAL
CALIBRATE to ORIGIN

Now use the VERTICAL POSITION control (on Display Unit panel) to situate the calibration dot exactly midway on the left vertical line (i.e., on the 0 point on the vertical axis). Then use HORIZONTAL POSITION control (on scope panel) to situate calibration dot exactly on vertical axis. Then set:

DISPLAY CALIBRATE to +FS

Use the screwdriver adjustment on the GAIN control CAL position to set the dot exactly on the top line of the scope. The dot should be on the center vertical axis. Then set:

DISPLAY CALIBRATE to -FS

The dot should be on the bottom line of the scope, on the center vertical axis.

Example of Calibrating the Scope for Positive Readings Only

In many cases, power spectra for one, the function displayed on the scope will be positive only. Thus, a full plus and minus scale will be wasteful of the resolution powers of the scope. To calibrate the scope for full-screen positive display only, set:

FUNCTION to CAL
CALIBRATE to ORIGIN

Now use the POSITION control to situate the calibration dot in the lower left corner of the screen. Use the HORIZONTAL POSITION control (on scope panel) to make any horizontal adjustments necessary to put the dot exactly on the corner. Now set:

DISPLAY CALIBRATE to +FS

Scope Calibrate

and use the GAIN control to situate the dot exactly on the top line of the screen. It should be on the center vertical axis. Any positive display will now be spread out over the full screen.

Example of Using Markers to Determine a Channel Number

It is desired to determine in what channel a particular point lies. First set following display conditions:

SWEEP LENGTH to 10

ORIGIN to LEFT

MARKER to OFF

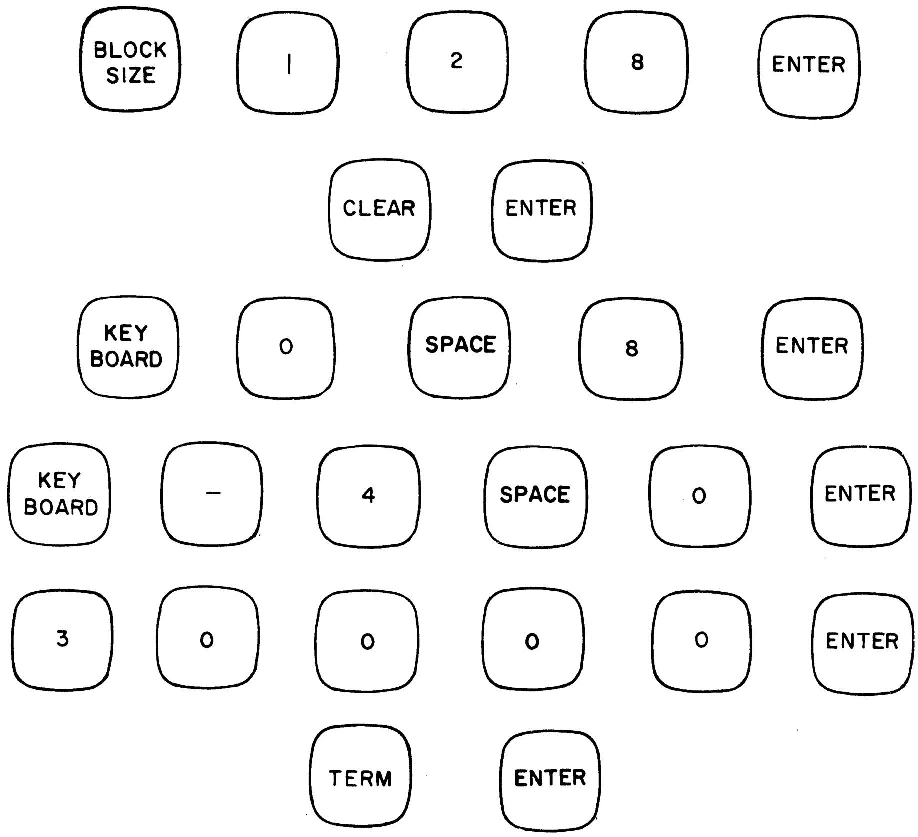
FUNCTION to DISPLAY

TYPE to POINT

MODE to REAL MAGNITUDE

SCALE to straight up position

We will manually enter data into the 0 data block, using a data block size of 128. To do this, press the following keys on the Keyboard:

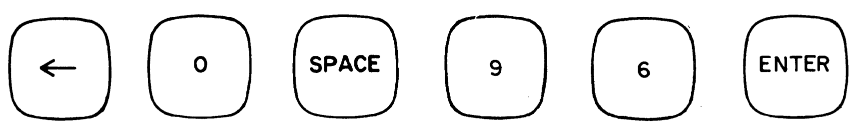


Channel markers

Note that this data--which would be a single point--is in the time domain (FREQ light is off), and has RECT coordinates.

To determine what channel the point is in, set the MARKER switch to 8 PT. Note that the data point is more intensified (as is every 8th point thereafter, across the screen). Therefore, we know that the data point is in channel 8.

It may be desired to move the data point a certain number of channels. Assume we want to move it 96 channels to the left. Press:



Then set MARKER switch to 32 PT position. Now the data point is slightly beyond the first 32-point marker. We can count up from that marker, or switch to 8 PT, and see that the data point is in channel 40.

EXTERNAL X-Y SCOPE OUTPUT

The Fourier Analyzer has provision to drive an external X-Y oscilloscope for remote or large screen display purposes. X and Y signals are derived from the X OUTPUT and Y OUTPUT connectors at the rear of the Display Mainframe. The format of these signals is the same as on the scope screen. The 0 voltage point is at the origin of the display (i.e., left edge, midway between top and bottom of screen). Positive full scale and negative full scale for the vertical (+2 and -2 volts) are at the top and bottom of the display screen. Horizontal full scale is given by an X OUTPUT of +5 volts. The output impedance for X OUTPUT and Y OUTPUT is 4.7 kΩ. To use these signals with the HP 1300A X-Y Oscilloscope, it is necessary to attenuate them so they will be within its range. This can be accomplished by adding a 1200Ω load across each output, or by using an HP 10604A load on each output. The connections for hooking up the 1300 X-Y scope to the Display Mainframe are given in the table below. A blanking signal is available at the UNBLANKING output. This signal supplies a -12.6 volt signal behind 10 kΩ to blank off the scope during retrace.

1300 X-Y Scope Hook-Up

Fourier Analyzer Display Unit	1300A Oscilloscope
X OUT with 1200Ω load	X INPUT (DC)
Y OUT with 1200Ω load	Y INPUT (DC)
UNBLANKING	SWEEP BLANK

Scope output

The position of the external display, and the horizontal and vertical gain, are unaffected by the POSITION and GAIN controls of the Fourier Analyzer Display Unit. They are controlled only by the external scope controls. Calibration of the external scope is accomplished by the regular Fourier Analyzer calibration procedure (see page 6-13). Fourier Analyzer markers, and display modes other than POINT, do not affect the external display.

PLOTTER OUTPUT

How Done

Plotter output is obtained with the HP 7004 X-Y Recorder, or with any other plotter that can handle the Display Unit output described below under "Using Other Plotters." As long as the vertical GAIN control on the Display Unit is in the CAL position, whatever is displayed on the scope will be plotted. All display controls except two affect the plotter: these are the GAIN and POSITION controls.

The following discussion applies primarily to the HP 7004 or 7034 X-Y Recorder. The Display Unit transmits points to the Plotter at a rate set by the PLOT RATE control (variable between 5 and 300 points per second). The Plotter pen traces an inked line between the points transmitted by the Display Unit. The pen moves at a variable rate according to the setting of the Display Unit PLOT RATE switch.

To begin a plot, place Display Unit FUNCTION switch to PLOT and place ARM/PLOT switch to ARM and back to PLOT. Plotter will then trace whatever is on the display. Refer to next paragraph for detailed setup.

Obtaining a Plot

To obtain a plot with the HP 7004B, the Plotter must be installed as described in the Plotter manual. Then, the procedure is as follows:

1. Insert graph paper in Plotter. Place POWER switch and CHART switch to ON, SERVO switch to OFF.
2. Be sure the ON-OFF switch on the 17173A Null Detector is ON. This is required on the 7004 to make the servo enable/disable function.
3. Be sure a continuous plotting pen is installed in the arm.
4. Set the PLOT RATE control to straight up position.
5. Set RECORDER switch on plotter to NO MUTE.
6. Be sure the vertical GAIN control on the Display Unit is in the CAL position.
7. Place 7004 POWER switch to ON and SERVO switch to ON.
8. Set the FUNCTION switch on the Display Unit to PLOT. This causes the scope to finish its current sweep and stop. The servo-disable line from the Display Unit to the plotter is open, turning on the Plotter servos.
9. Set ARM-PLOT switch to ARM. The pen should now move to the first point on the plot and stop. Place PEN switch to ON.
10. Set ARM-PLOT switch to PLOT. The pen should now plot the display. Set PLOT RATE control on Display Unit for speed desired. At the conclusion of the plot the pen should lift from the paper.

Plotter output

To return to the display mode, set the FUNCTION switch on the Fourier Analyzer to DISPLAY. This will ground the disable line, turning off the Plotter servos.

Calibration for Plotting

Before beginning a plot or a series of plots, the plotter should be calibrated. The following procedure applies to the HP 7004 or 7034 Plotter with 17173 Null Detector.

1. Be sure the ON-OFF switch on the 17173 Null Detector is ON.
2. Set the RECORDER switch to the NO MUTE position.
3. Set the FUNCTION switch on the Display Unit to CAL.
4. Set CALIBRATE switch to ORIGIN.
5. Set the X ZERO and Y ZERO controls on the Plotter to position pen in the same location as the dot on the scope.

6. Now set CALIBRATE switch to +FS and set the Plotter X VERNIER and Y VERNIER to match the dot position.

7. Set CALIBRATE to -FS, and check pen and dot position. Adjust plotter controls if necessary.

Note: the Display Unit puts out .500 ± .1% volts/display division.

Using Other Plotters

Plotters other than the HP models can be operated from the Display Unit by wiring a plug for connector J112 on the back of the Display Unit, according to the table below.

J112 Pin No.	Function
1	+X OUTPUT
2	-X OUTPUT
3	Not used
4	Pen Control No. 1
5	Servo disable
6	Completed plot
7	Ground
8	+Y OUTPUT
9	-Y OUTPUT
10	Not used
11	Pen Control No. 2
12	Not used
13	Seek
14	Ground

Plotter output

In normal operation the disable line is grounded when the FUNCTION switch on the Display Unit is in the DISPLAY position. When the FUNCTION switch is in the CAL or PLOT position the disable line is ungrounded (open circuit). This allows the servos of an external plotter to be muted when the Display Unit is in the display mode--that is, prevents the pen from attempting to follow the display. Pen controls 1 and 2 are shorted when the PLOT-ARM switch moves from the ARM to the PLOT position and a plot output is being delivered. The sense of contact closure can be reversed by moving the OC-CC jumper on Display Unit A7 card 05460-60004 to OC or CC, whichever is appropriate.

SECTION VII

PROGRAMMING AND EDITING OPERATIONS

ABOUT THIS SECTION

This section gives a brief explanation of how a program is set up on the Fourier Analyzer, and also gives details on the Programming and Editing keys. Some additional keys that pertain to the programming function are discussed at the end of the section.

WHAT IS A PROGRAM

A program is a sequence of commands that the Fourier Analyzer will perform automatically. Power spectra and averaging functions are two prime examples of programming applications.

A typical printout (called a listing) of a program on the Teleprinter might appear as follows.

		List			*Meaning*
1	L	1			Label 1
4	CL	1			Clear block 1
7	L	2			Label 2
10	RA	0	1		Analog into block 0, display block 1
14	A+	1			Add block 0 to block 1
17	X>	1			Store block 0 into block 1
20	#	2	100	0	Repeat steps from label 2, 100 times. 0 steps had been executed when this listing was run./End program.
25	•				

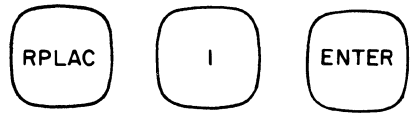
Program setup

The following paragraphs give an outline of the kind of procedure one uses to set up and run a program on the Fourier Analyzer. Details on each of the Key commands are provided in this section. A detailed, step-by-step explanation of how to set up and run a program is given in Section III, beginning page 3-26. Examples of some typical programs will be found in Section VIII.

SETTING UP A PROGRAM

There are 100 *element* locations in the program memory (200 with 16K machines), which, when you begin a program, may be empty, but most likely will be partially filled with commands from a previous program. Therefore, setting up a program consists of editing the contents of the program memory. Hence, every program entry must begin with an editing command--that is, a command involving one of the Editing keys on the Keyboard.

In general, you will enter a program with a RePLACe command, for example:



This starts to enter the program at line 1 (i.e., at the beginning of the program memory). If this line is not specified (i.e., N1 defaulted) the entire program memory will be cleared. The BUSY light then comes on, indicating the machine is waiting for program commands. The BUSY light must be on when a program is being entered. Subsequent lines (also called steps) that you enter will automatically displace downward in memory any previous lines.

After, or during the setting up of a program, the steps can be listed via a LIST command. Such a listing is shown on the previous page. The line numbers are automatically assigned by the Fourier Analyzer: the number being the number of elements to the end of the previous line (these are the same command elements that have been mentioned several times throughout this manual).

RUNNING A PROGRAM

To start a program it is necessary to move the internal pointer to the starting point. (This pointer moves down the lines as the program runs.) The starting point may be either a label or a line number. To start a program at a given line number, one uses a POINT command to set the pointer to a given line number, followed by a press of the CONTINUE key. The program then starts running at the line specified, displays the result and stops. To start the program at a given label number, one uses a JUMP-to-a-label command; the program runs through from that label, displays results and stops.

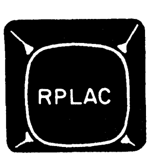
The program may be stopped at any time by pressing the STOP key and restarted by pressing the CONTINUE key. To find out where the pointer is after the program is stopped, an interrogate command (? key) is used: the line number is then printed out on the Teleprinter. The interrogate command is useful if a WHAT? signal should appear during the running of a program. The CONTINUE key will restart the program after an interrogation. The pointer will be set to line 0 when any of the following commands are executed: RePLACe, DELETe, and INSeRT.

Replace

Replace

This command replaces a single line or a range of lines with another line or range of lines. *It is the command most often used to write a new program, because it automatically displaces old program steps as the new ones are entered.* The first line to be replaced (for example, line 1 when beginning a new program) is given in this command; the end of the replacement sequence is indicated by a TERM ENTER command.

All lines beyond the replacement range are moved up or down in number, according to whether more or less elements were replaced; in other words, the program memory recompacts itself for most efficient use of memory space. The pointer is automatically set to 1.



The command form is:

[RPLAC] [LINE N1] [SPACE] [LINE N2] [ENTER]

where:

> N1 is the first line number to be replaced. If N1 is defaulted the entire program memory would be cleared.
>
> N2 is the final line number to be replaced. Default value = N1, in which case, only line N1 will be replaced.

If there is a program in memory and nothing beyond it, and you wish to enter another program immediately after it, the RePLACe command must call for the replacement of the *last line in the existing program* and then this line must be re-entered, as is, and the new program entered following it. The reason here is that the automatic restacking feature of the RePLACe command will not operate on a blank line (as would be the case if the command were to replace line 11 and on). The command in that case would go back until it found a full line (line 10) and replace that. Thus, the last line of the previous program must be the one replaced, and then re-entered prior to entering the new program.

Example 1. We will first look at a listing of a sample program. This listing was obtained by the command:

[LIST] [ENTER]

The listing is as follows:

	List			*Meaning*
1	RA	Ø	1	Analog in block 0, display block 1
5	F	Ø		Fourier transform block 0
8	*-			Conjugate multiply block 0
1Ø	A+	1		Add block 1 to block 0
13	.			End program

Now it is desired to replace line 8 with commands to clear block 0, channel 0; perform an interval-centered Hanning function in block 0; then do a conjugate multiply on the same block. The command sequence to accomplish all this is as follows:

[RPLAC] [8] [ENTER]

(Note that the Fourier Analyzer now goes to BUSY state.)

[CLEAR] [0] [SPACE] [0] [ENTER]

[HANN] [ENTER]

[* MULT] [ENTER]

[TERM] [ENTER]

(Note that the Fourier Analyzer now goes to the READY state.)

Now we call for a printout of the entire program via the command:

Replace

[LIST] [ENTER]

and the resulting printout would be:

```
 1 RA     Ø    1
 5 F      Ø
 8 CL     Ø    Ø
12 H1
14 *-
16 A+     1
19 .
```

Example 2. Now, in the preceding program it is desired to replace lines eight through fourteen with a single interval-centered HANNing command. The sequence of commands is:

RPLAC 8 SPACE 1 4 ENTER

(BUSY light now comes on.)

HANN ENTER

TERM ENTER

Now a listing, obtained by pressing LIST ENTER, would look as follows:

```
 1 RA      Ø      1
 5 F       Ø
 8 H1
1Ø A+      1
13 •
```

INSRT **Insert**

This permits the addition of any number of lines between any two given lines in a program, or following an END command. The pointer is automatically set to line 1.The command form is:

INSRT [LINE N1] ENTER

where:

N1 is the number of the line after which the new material will be inserted.

While the additional material is being inserted the BUSY light will be on. After the material is in, a TERM ENTER command must be given. The Fourier Analyzer then goes to READY. Program steps beyond those inserted are shifted farther down in memory.

DELET Delete

This is used to delete one step or a range of steps from a program. All lines below the delete move up to recompact the program. The pointer is automatically set to line 1. The command form is:

DELET LINE N1 SPACE LINE N2 ENTER

where:

N1 is the number of the first line to be deleted. If N1 is defaulted the entire program memory will be cleared.

N2 is the number of the last line to be deleted. Default value = N1 or in other words, if N2 is defaulted, then only line N1 will be deleted.

No TERM ENTER command is needed after the DELETe command.

NOTE: It is neither necessary nor desirable to delete all elements from program memory before entering a new program. See RePLACe command.

LIST List

This causes any program, or any part of a program, to be printed out on the Teleprinter.

To list an entire program from line 1. Use the command form:

LIST ENTER

The listing will proceed to the first END command, or if there is none, to the first blank line, or if the entire program memory is full, to the last line (line 99).

To list a single line: Use the command form:

LIST LINE N1 ENTER

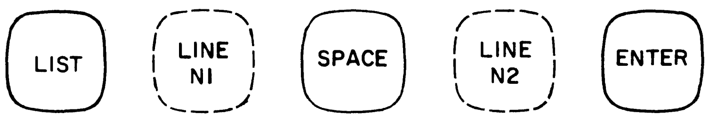
DELET

LIST

where:

N1 is the number of the line to be listed.
No default allowed.

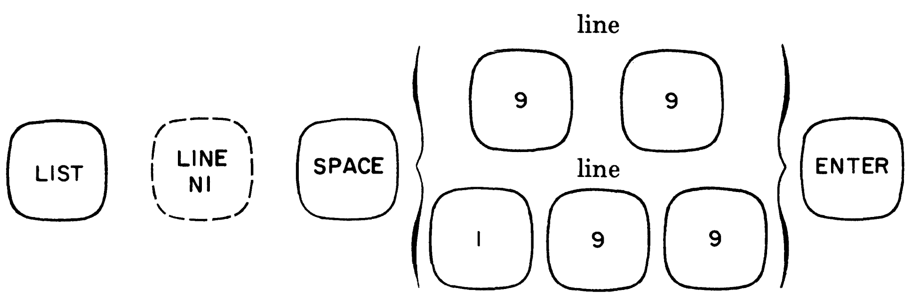
To list a range of lines: Use the command form:



where:

N1 is the first line number to be listed. *No default allowed.*
N2 is the last line number to be listed. *No default allowed.*

To list from a certain line to the end of a program: Use the command form:



where:

N1 is the first line to be listed. *No default allowed.*

99 is the number of the last line in 8K program memory, 199 is the number of the last line in 16K program memory. The listing will proceed to the next END statement and stop there, or, failing such a statement, to the first blank line, or failing that, to the last line—i.e., line 99 or 199.

Punching a Program on Paper Tape

There are two ways to punch a program on paper tape: Method 1 leaves the initial editing command (e.g., RePLACe) off the tape, so that to re-enter the tape, the command must be given separately; method 1 is recommended for systems with the optional Photoreader. Method 2 puts the editing command on the tape; method 2 is recommended for systems without the optional Photoreader and Punch.

Method 1 (No RePLACe command on tape)

1. Set Teleprinter switch to LINE. (If system has optional High Speed Punch, see note below).

2. Turn off Teleprinter punch (switch on box at left side of Teleprinter).

3. Enter the LIST command for the type of listing desired (see above) *but do not press ENTER.*

4. Set Teleprinter switch to LOCAL and turn on the Teleprinter punch.

5. Press the HERE IS key on the Teleprinter about three times to generate adequate tape leader. Turn off punch, set switch to LINE, turn on punch.

6. Now press the ENTER key. The LIST command will be executed and simultaneously punch the program or portion of program on the paper tape. Turn off punch and set Teleprinter switch to LOCAL.

7. Turn on punch and press the HERE IS key to generate tape trailer.

   NOTE: If the optional High Speed Punch is part of the system, press pushbutton 4 of the Computer DISPLAY REGISTER and apply the Teleprinter instructions above to the High Speed Punch. See also table on page 6-8.

Method 2 (RePLACe command is on tape)

1. Set the Teleprinter switch to LOCAL.

2. Turn on the Teleprinter punch (switch on box at middle of left side of Teleprinter).

3. Press the HERE IS key on the Teleprinter about three times to generate adequate tape leader.

4. Type the following command on the Teleprinter.

   (/) (TAPE R) (Line N1) (RETURN) (LINE FEED)

Punching A program

where:

   N1 is the line number at which you want to re-enter the program; this will normally be line 1.

5. Turn off the Teleprinter punch.

6. Set Teleprinter to LINE.

7. Give the equivalent LIST command on the Fourier Analyzer Keyboard for that given in step 4 above, *but do not press ENTER.*

8. Turn on Teleprinter punch.

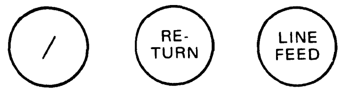
9. Press ENTER.

10. When the program has been punched out, turn off the Teleprinter punch.

11. Set the Teleprinter switch to LOCAL.

12. Turn the Teleprinter punch back on. (This repeated turning on and off of the punch is to prevent unnecessary punching of the tape.)

13. Type the following on the Teleprinter:

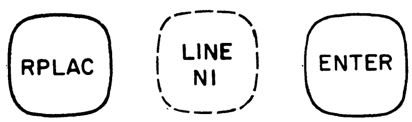


14. Generate adequate leader by pressing the HERE IS key three times.

Loading a program

To re-enter a program punched via Method 1, use the following procedure:

1. Set Teleprinter tape reader to OFF (switch is at left front of Teleprinter) and right front switch to LINE. If system has optional Photoreader, see note below.
2. Place paper tape in reader, lowering plastic cover in place.
3. Give the appropriate RePLACe command on the Fourier Analyzer Keyboard to enter the program into program memory.



where:

N1 is the starting line at which the program is to be entered. Remember if you are entering this program immediately behind another program, then you must immediately re-enter the last line of that program after giving the RePLACe command, or it will be deleted.

4. Turn on Teleprinter reader. The tape should now read in.

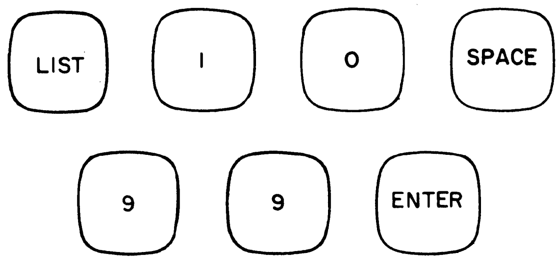
5. When the tape stops, press the TERM key on the Fourier Analyzer Keyboard.

   NOTE: A punched program can also be re-entered on the optional Photoreader by energizing this device instead of the Teleprinter reader, then press to light pushbutton 3 of Computer Switch Register before pressing RPLAC Key. See also table on page 4-15.

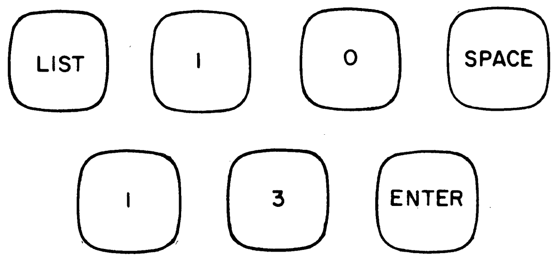
To re-enter a program punched via Method 2, use the following procedure:

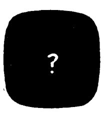
1. Set Teleprinter tape reader to OFF (switch at left front of Teleprinter) and right front switch to LINE.
2. Place paper tape in reader, lowering plastic cover in place.
3. Turn on Teleprinter reader. The tape should now read in, and stop when completed. The Keyboard should automatically go to READY state.

LIST Example 1: It is desired to list all steps in a program starting at line 10 (8K memory). The command is:



LIST Example 2: It is desired to list lines 10 through 13 of a program. The command is:

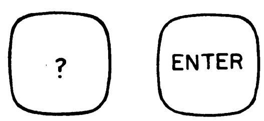




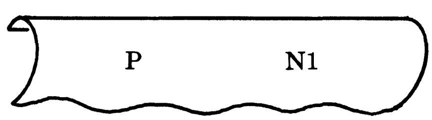
Interrogate

This command is used to determine where the pointer is at any given time. The pointer follows down the list of steps in a program as they are executed. The interrogate command can be used for example, after a WHAT? signal in order to determine which line produced that signal and caused the program to stop. The command form is simply:

Interrogate

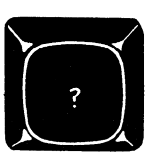


The Fourier Analyzer then prints out on the Teleprinter the pointer location as follows:



where:

N1 is the pointer location, i.e., line number.



TERM Terminate

This command is used to indicate to the Fourier Analyzer the end of a replace or insert editing function, and also to indicate the end of a Keyboard data entry. It returns the Keyboard to the READY mode. The command is:

TERM ENTER

JUMP Jump

When a program reaches this command, it immediately goes to a label specified by N, and proceeds from that point. This command can also be used to start a program running, simply by specifying the label (N) at the beginning of the program. The command form is:

JUMP [LABEL N1] ENTER

where:

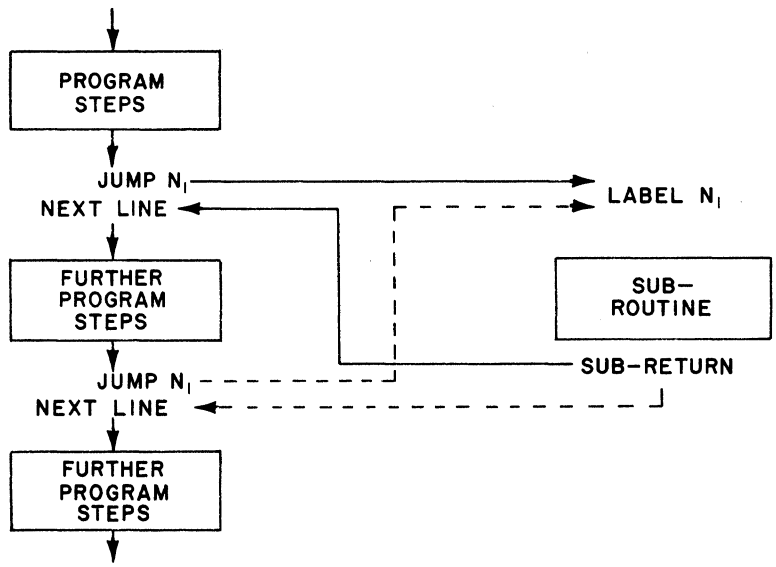
N1 is a label integer, -32,767 through +32,767.
No default allowed.

The JUMP command must be executed as part of a program or to start a program running. It has no meaning as an isolated command.

Example. If, at a certain point in a program, we wished to go back to label 2, and have the program proceed from there, the command would be:

JUMP 2 ENTER

The JUMP command also allows a short sequence of steps (i.e., a subroutine) to be executed from any point in a program. The subroutine must have a label N1 to identify its beginning, and a SUBReTuRN command to identify its conclusion. When a program reaches a JUMP command, it goes from that line to the first line in the subroutine, executes all the subroutine steps, then returns to the line following the one from which it entered the subroutine. The following diagram illustrates this:



Note: Subroutines cannot call on other subroutines.

LABEL Label

This command labels a line in the program irrespective of the line number. The line can then be gotten to via the JUMP and COUNT commands. The form of the command is:

Label

LABEL N1 ENTER

where:

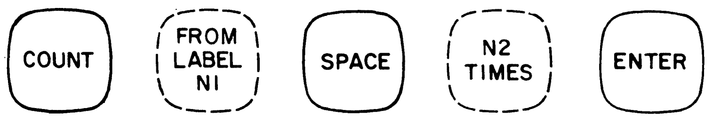
N1 is an integer, -32,767 through +32,767. *No default allowed.*

The LABEL command must be executed as part of a program. It has no meaning as an isolated command.

LABEL

Count

This causes a portion of the program whose starting point is label N1, to be repeated N2 times. In other words, this command is used to form a loop. The command form is:

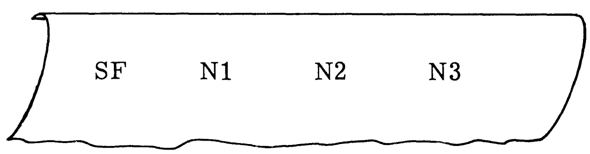


where:

N1 is the label of the starting point (must be -32,767 through +32,767). *No default allowed.*

N2 is the number of times the portion of the program is to be repeated. N2 must be an integer from 1 through +32,767. *No default allowed.*

When a program that includes a COUNT command is listed on the Teleprinter, the COUNT line will have the following form:



where:

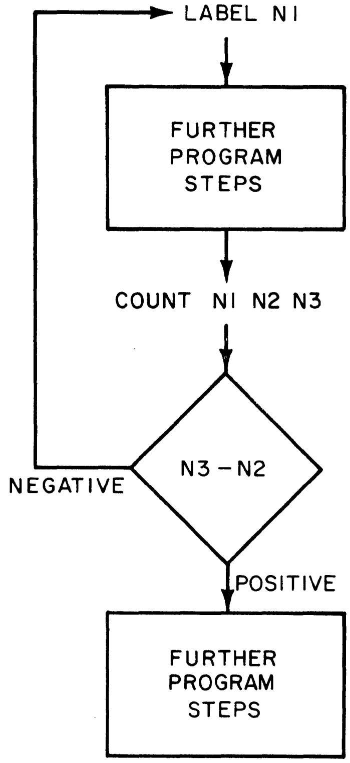
N1 is the label (an integer -32,767 through +32,767).

N2 is the number of times the portion of the program is to be repeated.

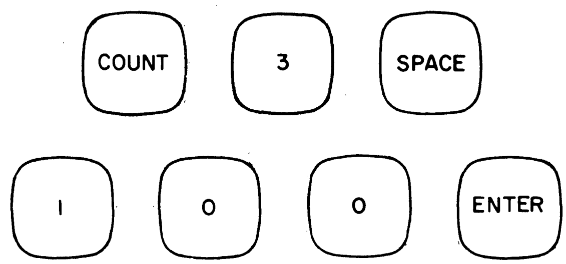
N3 is the number of times the loop had been repeated *at the time the LIST command was given.* In most cases this will be 0, i.e., one does not normally stop a program during execution to ask for a listing. But if this were done, then N3 would state the number of loops that had been completed at the time the LIST command was given.

Note: the COUNT does not automatically reset to 0 after being stopped by a listing, as above. To reset the count to 0, push RESTART or *re-enter the COUNT command* (using a RePLACe command first), then start the program again from the top.

The COUNT command must be executed as part of a program. It has no meaning as an isolated command. A flowgraph of a COUNT command is shown as follows.



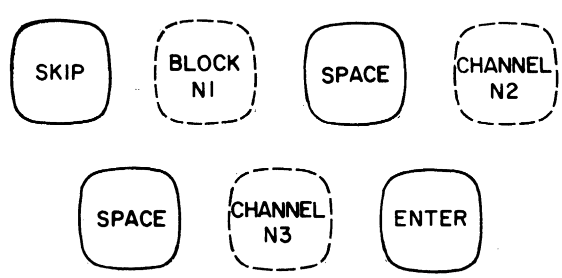
Example. If the user wanted to repeat the steps of a program from label 3, 100 times, the following would be placed at the bottom of the list of steps, immediately prior to the END command:

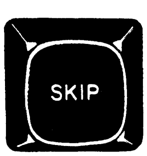


Skip

Skip

This command allows a skip of the next instruction in the program memory if any of the data points in block N1, between channels N2 and N3, are negative. The command form is:





where:

N1 is the data block examined; default value = block 0

N2 is the first channel examined; default value = channel 0

N3 is the last channel examined; default value = channel N2

If both N2 and N3 are defaulted, the entire block is examined.

The SKIP command is only executed as part of a program. If the user attempts to execute the command directly from the Keyboard a WHAT? signal will result.

Example. It is desired to perform an auto spectrum summation on data blocks which have positive dc values.

List				*Meaning*
1	L	1		Label 1
4	CL	1		Clear block 1
7	L	2		Label 2
10	RA	0	1	Analog into block 0, display block 1
14	F	0		Fourier transform of block 0
17	IF	0	0	Skip next program step if channel 0 of block 0 is negative
22	J	3		Jump to label 3
25	J	2		Jump to label 2
28	L	3		Label 3
31	*-			Self-conjugate multiply
33	A+	1		Add block 1 to block 0
36	X>	1		Store block 0 into block 1
39	#	2	100	Count 100 repetitions from label 2
44	.			End program

SUB RTRN

Subroutine Return

This command must be placed at the conclusion of any subroutine in order to return the program control to where it entered the subroutine, i.e., the step immediately following JUMP. A diagram of the subroutine format is shown under the JUMP command, page 7-12. The form of the SUB ReTuRN command is simply:

SUB RTRN   ENTER

END

End

This is used to indicate the conclusion of a program. This command should always be used in case there are residual steps from earlier programs still in memory following the new program. The command form is:

END   ENTER

The END command must be executed as part of a program. It has no meaning as an isolated command.

Sub-return
End
Block size

BLOCK SIZE

Block Size

This command establishes the number of channels in each data block. Permissible block sizes are shown above the indicator lights along the top of the Control Unit Keyboard. Maximum block size permitted depends upon size of computer memory (see *data block*, page 3-1).

SUB RTRN

END

BLOCK SIZE

If a block size that is not allowed is commanded, the system will give a WHAT? signal. After a BLOCK SIZE command is executed, block 0 is always displayed. The command form is:

[BLOCK SIZE] [BLOCK N1] [ENTER]

where:

N1 is the number of channels, as 64, 128, 256, 512, etc.

When block size is decreased, i.e., original block broken into smaller blocks, the new blocks will assume the same data qualifiers (scale factor, block calibrator, frequency code) as the original blocks. However, a coded BLOCK SIZE command can be given which prevents the new blocks from assuming the same data qualifiers. The format for this command is:

[BLOCK SIZE] [BLOCK N1] [SPACE] [0] [ENTER]

where:

N1 is number of channels

0 is a code number for the command.

POINT Pointer

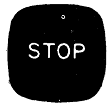
The command sets the pointer to any line number (not label number) in the program. Then, when the CONTINUE key is pressed, the program will restart at that line number. (Don't press Edit keys RPLAC, INSRT, or DELET meanwhile since these return the pointer to line 1). The purpose of this is to restart the program at any point not specified by a label. Line numbers are obtained on a printout of the program via the LIST command. The command form is:

[POINT] [LINE N1] [ENTER]

where:

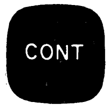
N1 is the line number at which the program is to restart. Default value = line 1, in which case the program restarts at the beginning of the program memory.

Note that the CONTINUE key must be pressed to start the program.



Stop

&

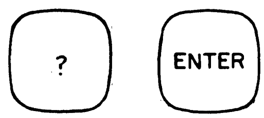


Continue

When a program is running, pressing the STOP key will cause the program to stop after it completes the present program line. The STOP command executes automatically: no ENTER keystroke is needed. It differs from the RESTART command in that it always permits the operation to be completed. RESTART interrupts in the middle of an operation, and thus leaves meaningless data in memory. Neither STOP nor RESTART, however, have any effect on the Fourier master program, that is, they do not turn off the Fourier Analyzer.

Pressing the CONTINUE key (no ENTER keystroke required) causes the program to resume from the line at which the pointer is set. (Not the following line, but that line.)

To find out at which line the pointer is set, use the interrogate command:

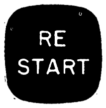


The STOP and CONTINUE keys operate in conjunction with the STEP-RUN switch on the Keyboard. When in the STEP position, the program proceeds one step at a time; when in RUN, it automatically proceeds through to the end.

STEP-RUN Switch

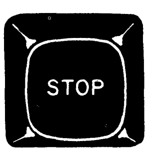
In the STEP position, this switch causes the program to proceed one step at a time. The step is completed, the appropriate display of results is given, and then the program stops. To make the program continue, the CONTINUE key must be pressed. The next step is then performed, results are displayed, and the program again stops. Thus, the STEP position is useful in debugging a program. In the RUN position, the switch causes the program to continue through all its steps automatically.

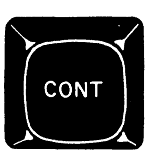
Stop
Continue
Restart

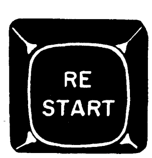


Restart

Pushing RESTART key (no ENTER necessary) restarts the entire Fourier program. If it is pushed while a program is running, it interrupts the completion of the program. This may leave meaningless data in memory. RESTART resets all drivers, sets the loop counters of all COUNT commands in the program memory to zero, and extinguishes the WHAT? light. The system then proceeds to display the block currently acted upon and waits for Keyboard command.







SECTION VIII

SAMPLE PROGRAMS

ABOUT THIS SECTION

This section provides a number of typical sample programs, plus a discussion of the considerations that went into the development of each. A Teleprinter listing of each program is also included. Because of the length of some of the programs, the illustrated key format will no longer be used: instead the commands will be listed in tabular form.

For details on how programs are entered into the Fourier Analyzer, see Section III, beginning page 3-26.

SPECTRUM AVERAGING PROGRAMS

The following remarks apply to the first four sample programs: These are the auto spectrum summation average program, cross power spectrum summation average, auto spectrum stable average, and auto spectrum exponentially decaying average. All these programs consist of power spectra (or linear spectra) averaged in different ways.

Power Spectrum

To compute a power spectrum of a random process, it is necessary to generate a positive quantity which can be averaged. The resulting average will then be a measure of the energy in each frequency of the spectrum band.

The typical random process yields a Fourier transform which will have positive and negative real, and positive and negative imaginary values, randomly distributed across the spectrum. Thus if the transform is averaged without a time synchronization, the results will average out to zero. To obtain the required positive quantities, therefore, the Fourier transform must be conjugate multiplied, yielding a positive quantity at each frequency which is the energy at that frequency. If we sum a number of such spectra, and then divide each frequency channel by that number, we achieve the desired average. **Power spectrum**

The Computer uses 16-bit data words (15 bits plus sign) thus providing a range of 0 to 65,534 numbers (-32,767 to +32,767). 65,534 is about 4.8 decades. If we compute the linear spectrum, we have 20 dB/decade times 4.8 or 96 dB dynamic range. With the power spectrum we have 10 dB/decade times 4.8 or only 48 dB dynamic range. However, since the result of a power spectrum is in double precision, the same dynamic range is maintained.

If the function is a Gaussian process then the real and imaginary components of the Fourier transform are a Gaussian random quantity, each frequency component having a value expressed by: $a+ib$, where a is the value of the real component and b the value of the imaginary. When we square this expression by conjugate multiplication, we arrive at an expression of the form $a^2 + b^2$, whose statistics are Chi-squared with a degree of freedom for each squared term, or in other words, with two degrees of freedom. Thus each time record yields one estimate of a power spectrum with two degrees of freedom. For the *first* estimate, the variance on the expected value of each spectral line is equal to the mean value, which is a very large statistical variation. However, the variation decreases as the number of estimates increases. That is:

$$\sigma = \frac{1}{\sqrt{K}} \times 100$$

where:

σ is one standard deviation in percent.

K is the number of estimates.

The same programming strategy is employed for all four spectrum averaging programs. This strategy consists in reading data into block 0, forming the power spectrum there, and reserving block 1 to store the accumulating sums of the estimates. Then the contents of block 1, which are 0 the first time, are added to block 0. Thus the current record is always added to the sum of the past records. The result is stored back into block 1, and a new record obtained in block 0. The current record is worked on in block 0 because in all arithmetic operations involving two blocks, one of them must be block 0.

A feature of the analog input command is that it permits the user to take in data in one block (or two in the case of dual channel input) and display a different block. In the case of spectrum averaging programs, this permits the user to observe the sum accumulating in the storage block, which is of more interest than observing the input block, since this would merely show each record of a random process.

When running the summation average programs, the user will observe the sums accumulating in the storage block. If the program is stopped in mid-run, a calibrated average up to that point cannot be read, since the division takes place at the end. However, the user can compute the calibrated average by first listing the COUNT line of the program which will also give the number of sums up to that point. Then the block can be manually divided by that number to give a calibrated average.

The stable and the decaying average programs provide a calibrated average at every repetition, and thus no additional operations are required when the program is stopped in mid-run.

Linear spectrum

Linear Spectrum

To accumulate a linear spectrum rather than a power spectrum, one uses a polar coordinate command in place of the conjugate multiply command. The result will be the magnitude values in the real part of the data block, and the phase in the imaginary. The magnitudes then sum as repeated spectral estimates are taken and stored. The phase values will randomly sum and so must be disregarded. The major disadvantage of a linear spectrum (as opposed to a power spectrum) is that it takes longer to form. This is because the polar coordinate operation takes longer than conjugate multiplication.

Summation Average

The summation average is the simplest and fastest average, because it consists solely of summations, followed by a division at the end.

$$A_n = \frac{\sum_{0}^{n-1} I_i}{n}$$

where:

A_n = the average after n estimates

I_i = the i^{th} estimate

n = the number of estimates

However, as mentioned above, it has the disadvantage of not being calibrated at all times. Also, the display changes very rapidly as the summation grows. The stable average on the other hand, is always calibrated, and the display does not change so rapidly. But it is also slower than the summation average.

Spectrum Stable Average

The spectrum stable average is computed from the following algorithm:

$$A_n = A_{n-1} + \frac{I_n - A_{n-1}}{n}$$

where:

A_n = the average after n estimates

A_{n-1} = the average after n-1 estimates

I_n = the n^{th} estimate

n = the number of estimates

Although the sample programs which employ this algorithm are for spectrum averaging, the algorithm can be applied to any averaging process, for example time averaging. The algorithm is derived as follows. The previous, i.e., n-1 average, A_{n-1} is:

$$A_{n-1} = \frac{\sum_{0}^{n-1} I_i}{n-1}$$

Then, multiplying:

$$(A_{n-1})(n-1) = \sum_{o}^{n-1} I_i$$

If we now add the nth estimate, I_n, to both sides, we get:

$$(A_{n-1})(n-1) + I_n = \sum_{o}^{n-1} I_i + I_n$$

Dividing both sides by n:

$$\frac{(A_{n-1})(n-1) + I_n}{n} = \frac{\sum_{o}^{n-1} I_i + I_n}{n}$$

But the right hand side is A_n, the average after n estimates. And, rewriting the left hand side, we get:

$$\frac{(A_{n-1})\not{n}}{\not{n}} - \frac{A_{n-1}}{n} + \frac{I_n}{n} = A_n$$

Further rewriting:

$$A_n = A_{n-1} + \frac{I_n - A_{n-1}}{n}$$

The program can thus be stopped at any time, and the average in block 1 will always be calibrated. The price of this convenience is the additional computer time required to do the subtracting, division, and addition after each estimate.

Exponentially Decaying Average

In a stable average, all ensemble estimates contribute equally to the final average. This is satisfactory if the signal as a whole is not changing with time. But if it is changing compared to one sample record, we may want to see what the average is over a short period; in other words we may want to look at an average that approaches the final value exponentially as $(1 - e^{-n/k})$. That is:

$$A_n = A_f (1 - e^{-n/k})$$

where:

A_n = the average after n estimates

A_f = the final average

n = the number of estimates

k = the weighting factor

It is clear that A_n is a close approximation to A_f only as n grows large compared to k.

The derivation of the above equation from the running average algorithm is as follows:

$$A_n = A_{n-1} + \frac{I_n - A_{n-1}}{n}$$

Therefore

$$A_1 = A_0 + \frac{I_1 - A_0}{k} = \left(\frac{k-1}{k}\right) A_0 + \frac{I_1}{k}$$

And

$$A_2 = \left(\frac{k-1}{k}\right) A_1 + \frac{I_2}{k} = \left(\frac{k-1}{k}\right)^2 A_0 + \frac{k-1}{k^2} I_1 + \frac{I_2}{k}$$

But A_0, the average after 0 estimates, is 0. Therefore, by induction,

$$E\left[A_n\right] = E\left[\frac{1}{k} \sum_{i=1}^{n} I_i \left(\frac{k-1}{k}\right)^{n-i}\right]$$

where:

E means "the mean value of..."

I_i = the i^{th} estimate

$$E\left[A_n\right] = \frac{1}{k} \sum_{i=1}^{n} E\left[I_i \left(\frac{k-1}{k}\right)^{n-i}\right]$$

$$E\left[A_n\right] = \frac{1}{k} \sum_{i=1}^{n} \left(\frac{k-1}{k}\right)^{n-i} E\left[I_i\right]$$

$$E\left[A_n\right] = E\left[I_i\right] \left(\frac{k-1}{k}\right)^{n} \frac{1}{k} \sum_{i=1}^{n} \left(\frac{k-1}{k}\right)^{-i}$$

But since

$$\sum_{i=0}^{n-1} ar^i = a\left(\frac{r^n - 1}{r - 1}\right)$$

and letting $a = 1,\ r = \left(\frac{k-1}{k}\right)^{-1} = \frac{k}{k-1}$,

$$\sum_{i=1}^{n} \left(\frac{k-1}{k}\right)^{-i} = \frac{\left(\frac{k}{k-1}\right)^{n+1} - \frac{k}{k-1}}{\frac{k}{k-1} - 1}$$

$$E\left[A_n\right] = E\left[I_i\right] \frac{1}{k} \left(\frac{k-1}{k}\right)^{n} \left[\frac{\left(\frac{k}{k-1}\right)^{n+1} - \frac{k}{k-1}}{\frac{k}{k-1} - 1} \right]$$

$$E\left[A_n\right] = E\left[I_i\right] \frac{1}{k} \left[\frac{1 - \left(\frac{k-1}{k}\right)^{n}}{1 - \frac{k-1}{k}} \right]$$

$$E\left[A_n\right] = E\left[I_i\right] \left[1 - \frac{k-1}{k}^{\,n}\right]$$

For k very large,

$$E\left[A_n\right] = E\left[I_i\right] \left(1 - e^{-n/k}\right)$$

For sufficiently large n, $E[A_n]$ approaches A_n, $E[I_i]$ approaches A_f. Thus:

$$A_n = A_f \left(1 - e^{-n/k}\right)$$

The exponentially decaying average like the stable average will always produce a calibrated result if the program is stopped in mid-run.

The effective number of averages contained in the ensemble is k. Thus, each spectrum in an exponentially decaying average has 2k degrees of freedom.

AUTO SPECTRUM SUMMATION AVERAGE PROGRAM

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
LABEL 0 ENTER			Establishes initial label point
CLEAR 1 ENTER		Cleared	Initializes block 1 (removes old data)
LABEL 1 ENTER		Sum of past power spectra $G_{xx}(f)$. 0 first time.	Establishes target point for ensemble average
ANALOG IN 0 SPACE 1 ENTER	Current time record	"	Reads current record into block 0, displays block 1, where sums are accumulating
F ENTER	Fourier transform of current record	"	Fourier transforms block 0
POWER SPECT ENTER	(undefined)	Sum of current plus past $G_{xx}(f)$'s	Forms $G_{xx}(f)$ of current record and stores sum of current plus past $G_{xx}(f)$'s in block 1 for next pass
COUNT 1 SPACE N2 ENTER	"	"	Loop back to target label 1 N2 times, for N2 sums
÷ 1 SPACE N2 ENTER	"	Average of N2 $G_{xx}(f)$'s	Forms average of N2 $G_{xx}(f)$ sums
END ENTER	"	"	Ends program

Auto spectrum averaging

LISTING OF AUTO SPECTRUM
SUMMATION AVERAGE PROGRAM

(N2 = 100)

```
 1 L        Ø
 4 CL       1
 7 L        1
1Ø RA       Ø       1
14 F
16 SP
18 #        1     1ØØ      Ø
23 :        1
26 .
```

Auto spectrum averaging

LINEAR SPECTRUM SUMMATION AVERAGE PROGRAM

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
LABEL 0 ENTER			Establishes initial label point
CLEAR 1 ENTER		Cleared	Initializes block 1 (removes old data)
LABEL 1 ENTER		Sum of past linear spectra. 0 first time.	Establishes target point for ensemble average
ANALOG IN 0 SPACE 1 ENTER	Current time record	"	Reads current record into block 0, displays block 1, where sums are accumulating
F ENTER	Fourier transform of current record	"	Fourier transforms block 0
POLAR ENTER	Current linear spectrum in polar coord.	"	Convert linear spectrum to polar coordinates (magnitude and phase)
+ 1 ENTER	Sum of current plus past linear spectra	"	Forms sum of current plus past linear spectra in block 0
STORE 1 ENTER	"	Sum of current plus past linear spectra	Stores sum of current plus past linear spectra in block 1 for next pass
COUNT 1 SPACE N2 ENTER	"	"	Loop back to target label 1 N2 times, for N2 sums
÷ 0 SPACE N2 ENTER	Average of N2 linear spectra sums	Sum of current plus past linear spectra	Forms average of N2 linear spectra
END ENTER	Average of N2 linear spectra sums	"	Ends program

LISTING OF LINEAR SPECTRUM
SUMMATION AVERAGE PROGRAM

```
  1 L      0
  4 CL     1
  7 L      1
 10 RA     0      1
 14 F
 16 TP
 18 A+     1
 21 X>     1
 24 #      1     10     0
 29 :      0     10
 33 .
```

Linear spectrum averaging

CROSS SPECTRUM SUMMATION AVERAGE PROGRAM

Program Commands	Contents of Block 0	Contents of Block 1	Contents of Block 2	Purpose of Command
LABEL 0 ENTER				Establishes initial label point
CLEAR 2 ENTER			Cleared	Initializes block 2 (removes old data)
LABEL 1 ENTER			"	Establishes target point for cross power spectrum $G_{yx}(f)$ sums
ANALOG IN 0 SPACE 1 ENTER	Current time record from channel A	Current time record from channel B	Sum of past $G_{yx}(f)$'s. 0 first time.	Data input
F 0 SPACE 1 ENTER	Fourier transform of channel A time record	Fourier transform of channel B time record	"	Obtain Fourier transform of data
MULT* 1 ENTER	$G_{yx}(f)$ of channel A and channel B records	"	"	Obtain cross power spectrum
+ 2 ENTER	Sum of current plus past $G_{yx}(f)$'s	"	"	Adds current $G_{yx}(f)$'s to sum of past $G_{yx}(f)$'s
STORE 2 ENTER	"	"	"	Stores result of current pass for next pass
COUNT 1 SPACE N2 ENTER	"	"	"	Loop back to target label 1 N2 times for N2 sums

Cross spectrum averaging

CROSS SPECTRUM SUMMATION AVERAGE PROGRAM (Cont'd)

Program Commands	**Contents of Block 0**	**Contents of Block 1**	**Contents of Block 2**	**Purpose of Command**
÷ 0 SPACE N2 ENTER	Average of N2 $G_{yx}(f)$'s	Fourier transform of channel B time record	Sum of past $G_{yx}(f)$'s. 0 first time.	Form average of $G_{yx}(f)$'s
END ENTER	"	"	"	Ends program

Cross spectrum averaging

LISTING FOR CROSS SPECTRUM
SUMMATION AVERAGE PROGRAM

(N2 = 50)

```
 1 L     0
 4 CL    2
 7 L     1
10 RA    0    2
14 F     0    1
18 *-    1
21 A+    2
24 X>    2
27 #     1    50    0
32 :     0    50
36 .
```

AUTO SPECTRUM STABLE AVERAGE PROGRAM

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
LABEL 0 ENTER			Establishes initial label point
CLEAR 1 ENTER		Cleared	Initializes block 1 (removes old data)
LABEL 1 ENTER		True average of past power spectra $G_{XX}(f)$'s. 0 first time.	Establishes target point for stable average
ANALOG IN 0 SPACE 1 ENTER	Current time record	"	Reads current record into block 0, displays block 1, where average is accumulating
F ENTER	Fourier transform of current record	"	Fourier transforms block 0
MULT* ENTER①	$G_{XX}(f)$ of current record	"	Forms $G_{XX}(f)$ of current record
- 1 ENTER	Difference between current $G_{XX}(f)$ and average of past $G_{XX}(f)$'s	"	Subtract average of past $G_{XX}(f)$'s (in block 1) from current $G_{XX}(f)$ in block 0
÷ 0 SPACE 0 ENTER	Result of dividing above difference by no. of $G_{XX}(f)$'s averaged so far	"	Divides the variance between current $G_{XX}(f)$ and past $G_{XX}(f)$'s by the loop counter, i. e., no. of $G_{XX}(f)$'s averaged so far

① Replace this command with POLAR ENTER if you want to accumulate the average of voltage rather than power spectra.

Stable average

AUTO SPECTRUM STABLE AVERAGE PROGRAM (Cont'd)

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
+1 ENTER	Sum of above division result and average of past $G_{XX}(f)$'s	True average of past power spectrums $G_{XX}(f)$'s. 0 first time.	Adds division result to average of past $G_{XX}(f)$'s to obtain new average
STORE 1 ENTER	Sum of above division result and average of past $G_{XX}(f)$'s (i. e., new average)	same as block 0 (i. e., new average)	Stores result of current pass for next pass
COUNT 1 SPACE N2 ENTER	"	"	Loop back to label 1 N2 times, for N2 averages
END ENTER	"	"	Ends program

LISTING OF AUTO SPECTRUM
STABLE AVERAGE PROGRAM

(N2 = 100)

Stable average

```
 1 L     0
 4 CL    1
 7 L     1
10 RA    0     1
14 F
16 *-
18 A-    1
21 :     0     0
25 A+    1
28 X>    1
31 #     1   100     0
36 •
```

AUTO SPECTRUM EXPONENTIALLY
DECAYING AVERAGE PROGRAM

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
LABEL 0 ENTER			Establishes initial label point
CLEAR 1 ENTER		Cleared	Initializes block 1 (removes old data)
LABEL 1 ENTER		True average of past power spectrums $G_{XX}(f)$'s. 0 first time.	Establishes target point for ensemble average
ANALOG IN 0 SPACE 1 ENTER	Current time record	"	Reads current record into block 0, displays block 1, where average is accumulating
F ENTER	Fourier transform of current record	"	Fourier transforms block 0
MULT* ENTER ①	$G_{XX}(f)$ of current record	"	Forms $G_{XX}(f)$ of current record
-1 ENTER	Difference between current $G_{XX}(f)$ and average of past $G_{XX}(f)$'s	"	Subtract average of past $G_{XX}(f)$'s (in block 1) from current $G_{XX}(f)$ in block 0
÷ 0 SPACE K ENTER	Result of dividing above difference by time constant, K	"	Divides the variance between current $G_{XX}(f)$ and past $G_{XX}(f)$'s by time constant, K

① Replace this command with POLAR ENTER if you want to accumulate the average of voltage rather than power spectra.

Exp. decaying average

AUTO SPECTRUM EXPONENTIALLY
DECAYING AVERAGE PROGRAM (Cont'd)

Program Commands	**Contents of Block 0**	**Contents of Block 1**	**Purpose of Command**
+1 ENTER	Sum of above division result and average of past $G_{XX}(f)$'s	True average of past power spectrums $G_{XX}(f)$'s. 0 first time.	Adds division result to average of past $G_{XX}(f)$'s to obtain new average
STORE 1 ENTER	Sum of above division result and average of past $G_{XX}(f)$'s (i.e., new average)	same as block 0 (i.e., new average)	Stores result of current pass for next pass
JUMP 1 ENTER	"	"	Loop back to Label 1. Note: program keeps running until STOP key is pressed
END ENTER	"	"	Ends program

LISTING OF SPECTRUM
EXPONENTIALLY DECAYING AVERAGE PROGRAM

(K = 20)

Exp. decaying average

```
 1 L    0
 4 CL   1
 7 L    1
10 RA   0    1
14 F
16 *-
18 A-   1
21 :    0    20
25 A+   1
28 X>   1
31 J    1
34 .
```

TIME ENSEMBLE AVERAGE

If a signal is buried in noise, and a time synchronization is available so the user knows the periodicity of the signal, the time averaging program is an efficient, very fast means of recovering the desired signal either in the time or spectrum domain. This is because the Fourier transform and subsequent conjugate multiply (or conversion to polar coordinates in the case of a linear spectrum) is only done once, namely after the time average is completed. During the time average the signal always has the same value in each ensemble, but the noise having random plus and minus values will average out.

TIME ENSEMBLE AVERAGE PROGRAM

Note: A time synchronization is required for this program.

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
LABEL 0 ENTER			Establishes initial label point
CLEAR 1 ENTER		Cleared	Initializes block 1 (removes old data)
LABEL 1 ENTER		Past sum. 0 first time.	Establishes target point for summation
ANALOG IN 0 SPACE 1 ENTER	Current time record	"	Reads current record into block 0, displays block 1, where sums are accumulating
+1 ENTER	Sum of current record plus past sums	"	Forms sum of current record plus past sum in block 0
STORE 1 ENTER	"	Sum of current record plus past sum	Stores new sum in block 1 for next pass

Time ensemble averaging

TIME ENSEMBLE AVERAGE PROGRAM (Cont'd)

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
COUNT 1 SPACE N2 ENTER	New sum	New sum	Loop back to target label 1 N2 times for N2 sums
÷ 0 SPACE N2 ENTER	Average of N2 sums	"	Forms average of N2 sums
END ENTER	"	"	Ends program

LISTING OF TIME ENSEMBLE AVERAGE PROGRAM

(N2 = 50)

```
 1 L      0
 4 CL     1
 7 L      1
10 RA     0     1
14 A+     1
17 X>     1
20 #      1    50     0
25 :      0    50
29 .
```

Time ensemble averaging

NORMALIZING A HISTOGRAM TO UNIT AREA

In the program to normalize a histogram to unity, we see how imaginative use of the keys can create very flexible operations. The program runs as follows: The histogram is first integrated so that the total number of counts, n, will be in the last channel. Then the entire block is shifted so that the last channel is in the first channel and the remainder of the block is then cleared out. Then the entire block is again integrated, resulting in a function which is flat all the way across the block. Thus we have the number n in all channels of the block. We then divide the original histogram, which has remained in another block, by this block which means that the number of counts in each channel, n_i, are divided by n:

$$\text{Normalized histogram} = \sum_{0}^{N} \frac{n_i}{n} = \frac{1}{n} \sum_{0}^{N} n_i$$

This technique of integrating, shifting, and clearing can be used to normalize a spectrum or any other function to the unity value for its integral.

NORMALIZING A HISTOGRAM TO UNIT AREA
(Probability Density Function)

Note: This program assumes histogram is already in block 0.

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
STORE 1 ENTER	Histogram	Histogram	Stores histogram in block 1
∫ 1 ENTER	"	Integral of histogram	To obtain, in last channel of block 1, the value for the total no. of counts in the histogram
←1 SPACE N-1 ENTER, where N is the block size	"	Integral of histogram with last channel now in first channel	
CL 1 SPACE 1 SPACE N-1 ENTER	"	Total no. of counts in channel 0, rest of block cleared	Prepare to divide histogram by total no. of counts

Normalizing histogram

NORMALIZING A HISTOGRAM TO UNIT AREA (Cont'd)

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
∫ 1 ENTER	Histogram	Total no. of counts is in all channels	Prepare to divide histogram by total no. of counts
÷ 1 ENTER	Normalized histogram	Total no. of counts is in all channels	Obtain normalized histogram in block 0
END ENTER	Normalized histogram	Total no. of counts is in all channels	Ends program

Normalizing histogram

LISTING OF NORMALIZING HISTOGRAM PROGRAM
(Block Size = 512)

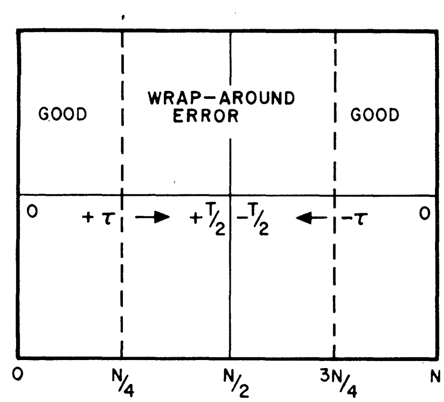
```
 1  X>     1
 4  $      1
 7  ←      1     512
11  CL     1       1     512
16  $      1
19  :      1
22  .
```

AUTO AND CROSS CORRELATION

The auto and cross correlation sample subroutine provides for the elimination of wrap-around error. This error is discussed under the correlation command, beginning page 5-27.

The subroutine is designed as follows: For a cross correlation, the data is read into block 0 and block 1. For an auto correlation, the single set of data is read into block 0, then stored in block 1. By the nature of the STORE command the initial set of data remains also in block 0.

As the first step in prevention of wrap-around error, the first and last quarters of block 1 are cleared out. Then block 0 and block 1 are cross-correlated, providing a cross correlation if the blocks are different, and an auto correlation if the blocks are the same. The result if displayed would be set up as follows:



It can be seen that there is still wrap-around error in the middle of the block from +N/4 to 3N/4 (N is the block size), and so as the final step in the subroutine, we clear these channels. Now the correlation is valid for 0 to ± T/4. The user can set the ORIGIN switch to CENTER in order to review the results more conveniently.

Auto & cross correlation

To achieve a higher degree of certainty, a number of correlations can be averaged via one of the averaging algorithms mentioned under the discussion of the spectrum averaging programs. The greater certainty appears because the averaging effectively makes the length of the integral in the function greater than T.

AUTO[1] AND CROSS-CORRELATION SUBROUTINE

(Eliminates wrap-around error; for discussion of this error, see page 5-22.)

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
LABEL 1 ENTER			Establishes initial label point
ANALOG IN ENTER [1]	Current time record from channel A	Current time record from channel B	Input data
CLEAR 1 SPACE 0 SPACE N/4 ENTER (where N = block size)	"	Current time record from channel B with first 1/4 block cleared	Clear first 1/4 block of time record from channel B
CLEAR 1 SPACE 3/4(N) SPACE N ENTER	"	Current time record from channel B with first and last 1/4's of block cleared	Clear last 1/4 block of time record from channel B
CORR 1 ENTER	Auto or cross correlation	Current time record from channel B with first and last 1/4's of block cleared	Obtain correlation function

Auto & cross correlation

[1] For auto-correlation, substitute the following two commands for the analog input step.

ANALOG IN ENTER	Current time record from channel A		Input data
STORE 1 ENTER	"	Current time record from channel A	Store block 0 in block 1 for later CORR command

AUTO AND CROSS-CORRELATION SUBROUTINE (Cont'd)

Program Commands	Contents of Block 0	Contents of Block 1	Purpose of Command
CLEAR 0 SPACE N/4 SPACE 3/4 (N) ENTER	Auto or cross correlation with channels N/4 to 3/4(N) cleared	Current time record from channel B with first and last 1/4's of block cleared	Clear wrap-around error from final function
Note: By placing the ORIGIN switch on CENTER, the display will show the + lags to the right of the vertical axis, and the - lags to the left. The last quarter block on each side was cleared by the above command.			

LISTING OF CROSS CORRELATION SUBROUTINE

(Block Size = 512)

```
 1 L     1
 4 RA    0     0
 8 CL    1     0    128
13 CL    1   384    512
18 CR    1
21 CL    0   128    384
26 .
```

LISTING OF AUTO CORRELATION SUBROUTINE

```
 1 L     1
 4 RA    0     0
 8 X>    1
11 CL    1     0    128
16 CL    1   384    512
21 CR    1
24 CL    0   128    384
29 .
```

Auto & cross correlation

TRANSFER FUNCTION AND COHERENCE FUNCTION

Transfer Function

A sample program is provided to measure the transfer and coherence functions between two sets of signals, X and Y[1]. The ensemble estimates are summed as in the summation average programs, although if the user desires, he can use other averaging techniques. In this way the average of the input auto spectrum ($\overline{G_{XX}}$), the output auto spectrum ($\overline{G_{YY}}$), and the cross spectrum ($\overline{G_{YX}}$) are formed. Then the transfer function is calculated:

$$H(f) = \frac{\overline{G_{YX}}}{\overline{G_{XX}}}$$

The reason why we average the individual lower spectra first, rather than averaging the entire function is that if a random noise process is used for excitation, not all frequencies may be present in any one estimate of the spectrum, and hence large computational errors may result, causing overflows in the division process. However, if we average (sum) the spectra first, then divide, there is little chance of these errors occurring.

Coherence Function

The coherence function tells the degree of similarity between input and output. Its equation is:

$$\gamma^2 = \frac{\left|\overline{G_{YX}}\right|^2}{\overline{G_{XX}} \cdot \overline{G_{YY}}}$$

It is computed by taking summation averages of the cross spectrum ($\overline{G_{YX}}$), then forming the square of its magnitude ($|\overline{G_{YX}}|^2$). This is done by multiplying $\overline{G_{YX}}$ by its self complex conjugate. Then $|\overline{G_{YX}}|^2$ is divided by the input auto spectrum average ($\overline{G_{XX}}$), then by the output auto spectrum average ($\overline{G_{YY}}$).

Transfer & coherence function

[1]Roth, Peter, "Digital Fourier Analysis", Hewlett-Packard Journal, June 1970, pp. 7—8.

Transfer/Coherence Function Program

Two sources of computational error should be kept in mind when computing transfer and coherence functions. Both tend to occur at those frequencies where the value of the power spectrum becomes very small, for example, at the high end of the filter roll-off curve. The first source of errors occurs as follows: As the power spectrum values get very small, they are in effect truncated from the right because the Computer uses a data word of finite length, i.e., 16 bits. Therefore the numerator or denominator of the transfer or coherence function may be reduced to numbers only 4 or 5 bits long.

As a result, the accuracy of these numbers is also diminished. The combined transfer and coherence function program on the following pages illustrates the requirement that six data blocks are necessary to calculate both transfer and coherence functions simultaneously. The listing of this combined program is shown below.

LISTING OF COMBINED TRANSFER AND COHERENCE FUNCTION PROGRAM

(N2 = 10)

```
 1 L     0
 4 CL    3
 7 CL    4
10 CL    5
13 L     1
16 RA    1    3
20 H1    1
23 H1    2
26 F     1    2
30 CL    1    0
34 CL    2    0
38 SP    1    2
42 #     1   10    0
47 CH    1    2
51 •
```

COMBINED TRANSFER AND COHERENCE FUNCTION PROGRAM

Program Commands	Contents of Block 0	Contents of Block 1	Contents of Block 2	Contents of Block 3	Contents of Block 4	Contents of Block 5	Purpose of Command
LABEL 0 ENTER							Establishes initial point for start of program
CLEAR 3 ENTER				Cleared			Initializes block 3 (Removes old data)
CLEAR 4 ENTER					Cleared		Initializes block 4 (Removes old data)
CLEAR 5 ENTER						Cleared	Initializes block 5 (Removes old data)
LABEL 1 ENTER				Sum of past G_{XX} (f)'s. 0 first time.	Sum of past G_{YY} (f)'s. 0 first time.	Sum of past G_{YX} (f)'s. 0 first time.	Establishes target point for power spectrum summation
ANALoG IN 1 SPACE 3 ENTER		Current time record from Channel A (input, X)	Current time record from Channel B (output, Y)	"	"	"	Data input
*HANN 1 ENTER		Current Channel A record multiplied by sine bell window	"	"	"	"	To make input A time record more periodic in sampling window *optional
*HANN 2 ENTER		"	Current Channel B record multiplied by sine bell window	"	"	"	To make input B time record more periodic in sampling window *optional
F1 SPACE 2 ENTER		Fourier transform of Channel A record	Fourier transform of Channel B record	"	"	"	Obtain Fourier transform of data
*CLEAR 1 SPACE 0 ENTER		Fourier transform of Channel A record minus dc value	Fourier transform of Channel B record	Sum of past G_{XX} (f)'s. 0 first time.	Sum of past G_{YY} (f)'s. 0 first time.	Sum of past G_{YX} (f)'s. 0 first time.	Clears dc value from Channel A record to improve spectrum dynamic range *optional

COMBINED TRANSFER AND COHERENCE FUNCTION PROGRAM

Program Commands	Contents of Block 0	Contents of Block 1	Contents of Block 2	Contents of Block 3	Contents of Block 4	Contents of Block 5	Purpose of Command
*CLEAR 2 SPACE 0 ENTER		Fourier transform of Channel A record minus dc value	Fourier transform of Channel B record minus dc value	Sum of past G_{XX} (f)'s. 0 first time.	Sum of past G_{YY} (f)'s. 0 first time.	Sum of past G_{YX} (f)'s. 0 first time.	Clears dc value from Channel B record to improve spectrum dynamic range *optional
POWER SPECT 1 SPACE 2 ENTER	Sum of current plus past G_{YX} (f)'s.	"	"	Sum of current plus past G_{XX} (f)'s.	Sum of current plus past G_{YY} (f)'s.	Sum of current plus past G_{YX} (f)'s.	Calculates current power spectra (input, output, and cross) adds past spectra and stores results for next pass
COUNT 1 SPACE N1 ENTER	"	"	"	"	"	"	Loop back to target label 1 N1 times
TRANS FCN 1 SPACE 2 ENTER	Coherence function $\gamma^2 = \frac{\left\lvert G_{YX}(f) \right\rvert^2}{G_{XX}(f) \cdot G_{YY}(f)}$	Transfer function $H(f) = \frac{\overline{G_{YX}(f)}}{G_{XX}(f)}$	Coherence function $\gamma^2 = \frac{\left\lvert G_{YX}(f) \right\rvert^2}{G_{XX}(f) \cdot G_{YY}(f)}$	"	"	"	Calculates transfer and coherence functions and stores results in block 1 and 2.
END ENTER	"	"	"	"	"	"	End program

SECTION IX

WRITING USER PROGRAMS FOR THE FOURIER ANALYZER

ABOUT THIS SECTION

This section provides information so the user can write his own Assembler language or FORTRAN programs for special processing of his data, in addition to that available on the Fourier Analyzer Keyboard. Pertinent addresses and data word formats are given so the user can write his own ADC and Display Unit drivers. A special command, called the Y command, is described which permits the user to interface his own program with the Fourier program. Full details on HP Assembler language can be found in the Assembler/BCS Training Manual or the Computer Micromanual.

USER PROGRAM LIBRARY

The Fourier Analyzer User Program Library (UPL) maintains a library of user programs that is available to HP Fourier Analyzer users. The UPL acquires new programs, distributes programs to users, and initiates action on reported program errors. To this end, the UPL welcomes all new programs that expand the capabilities of the Fourier Analyzer System. Detailed information on contributing programs to the UPL may be found in the Fourier Analyzer UPL Contributor's Guide.

Through use of the Fourier System Overlay Generator program (described later in this section) and the Keyboard USER PROG key, the user may load and call any of the UPL programs *without any specialized knowledge of software or computers*. For the user who is not interested in writing his own programs as described in this section, the UPL offers an alternate opportunity to add new programs to his system.

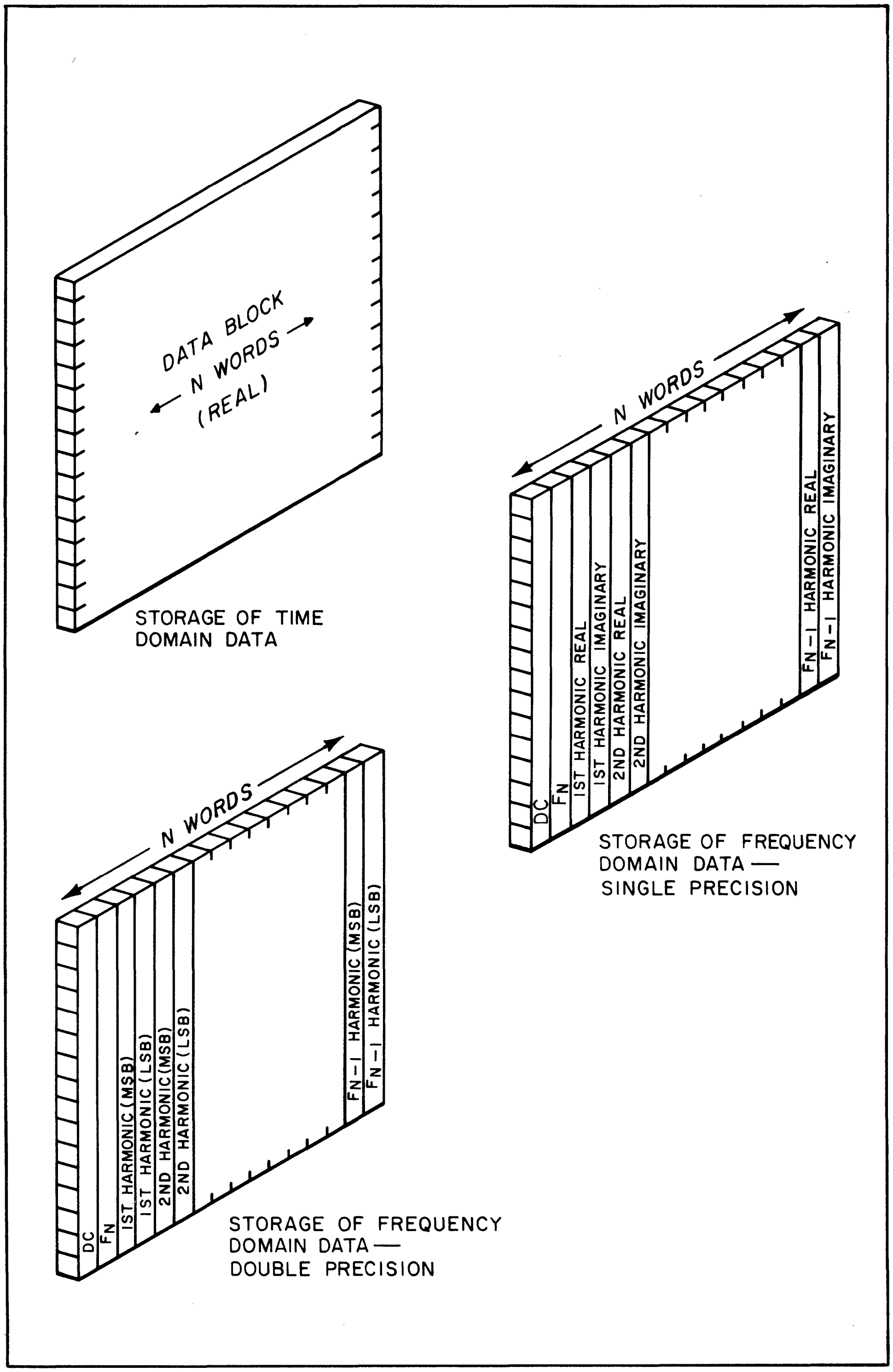
DATA STORAGE AND ORGANIZATION

Data is stored in the Computer memory as 16-bit words in two's complement format. Each data block contains a quantity of data words equal to the present block size. Data blocks are sequentially located beginning with data block 0 and running through data block (K/N) -1, where K is the size of the data memory in decimal and N is the block size in decimal.

The organization of data words within a block depends on the domain and coordinates of the data block. Figure 9-1 shows the composition of a time domain data block and a single precision and double precision frequency data block. If a frequency data block is in polar coordinates the magnitude of each harmonic is contained where the real part was previously and the phase value is stored in the location formerly containing the imaginary part.

Data organization

Figure 9-1. Data Storage Format



How Data Storage is Established

Data storage and data associated global words, are established by program NAM DTADO. To discuss details of this program, a listing of DTADO for the standard 16K Fourier System is shown below.

```
PAGE 0002 #01  NAM DTAD0          A-05451-94002-1

     00000                NAM DTAD0
                          ENT DTAD0,DRE.,BSMAX,BSMIN,NMAX0,NMAX
                          ENT SIZE,SIZM1,MSZ,POW
                          ENT SCAL0,CALB0,FREQ0
                          ENT SCAL
                          ENT MSZH
                          ENT ZIFFR
                          ENT DTAD
                          ENT FREQD
*
***  16K SYSTEM
* FOURIER DATASTORAGE AND DATA QUALIFIERS
*
*
*
     04000          MAX   EQU 2048
     00100          MIN   EQU 64
     20000          SUM   EQU 8192
     00200          MXBL  EQU 128
     00002          TMXBL EQU 2
     00014          LOG   EQU 12
     00100          SCTR  EQU 64
*
*
     00000 000000  DTAD  BSS SUM
     20000 000000R DTAD0 DEF DTAD
     20001 000000R DRE.  DEF DTAD
     20002 010000  BSMAX ABS MAX+MAX LARGEST DATABLOCK
     20003 000100  BSMIN ABS MIN      SMALLEST DATABLOCK
     20004 000200  NMAX0 ABS MXBL     LARGEST NUMBER OF DATABLOCKS
     20005 000002  NMAX  ABS TMXBL    # OF DATABLOCKS FOR GIVEN SIZE
*
     20006 010000  SIZE  ABS MAX+MAX
     20007 007777  SIZM1 ABS MAX+MAX-1 SIZE - 1
     20010 170000  MSZ   ABS -MAX-MAX NEGATIVE SIZE
     20011 174000  MSZH  ABS -MAX
     20012 000014  POW   ABS LOG      BLOCKSIZE POWER
*
     20013 000000  SCAL0 BSS MXBL
     20213 000000  CALB0 BSS MXBL
     20413 000000  FREQ0 BSS MXBL
     20613 020013R SCAL  DEF SCAL0
     20614 020613R FREQD DEF SCAL
     20615 177700  ZIFFR ABS -SCTR
*
```

The significance of each label is explained as follows:

To alter the data allocation only the first seven EQU statement need be changed.

MAX	EQU	(MAXIMUM BLOCKSIZE)/2
MIN	EQU	MINIMUM BLOCKSIZE
SUM	EQU	TOTAL DATA STORAGE
MXBL	EQU	LARGEST NUMBER OF DATABLOCKS (SUM/MIN)
TMXBL	EQU	NUMBER OF BLOCKS FOR PRESENT SIZE (SUM/2X(MAX))
LOG	EQU	LOG TO THE BASE 2 OF BLOCKSIZE (LOG2(MAX+MAX))
SCTR	EQU	MXBL/TMXBL

The EQU statements then set up the following labels for largest blocksize.

DTAD	BSS STATEMENT ESTABLISHING TOTAL DATA SPACE (EXAMPLE 8192)
DTADO	STARTING ADDRESS OF DATA SPACE (AND THAT OF BLOCK 0)
DRE.	STARTING ADDRESS OF BLOCK N1. SET UP BY SUB-ROUTINE LOCAT.
BSMAX	LARGEST BLOCKSIZE (EXAMPLE 4096)
BSMIN	SMALLEST BLOCKSIZE (EXAMPLE 64)
NMAXO	LARGEST NUMBER OF DATABLOCKS (EXAMPLE 128)
NMAX	NUMBER OF DATABLOCKS FOR PRESENT BLOCKSIZE. (EXAMPLE 2) SET UP BY COMMAND SUBROUTINE BLOK
SIZE	PRESENT BLOCKSIZE (EXAMPLE 4096) SET UP BY COMMAND SUBROUTINE BLOK
SIZM1	SIZE-1 (EXAMPLE 4095) SET UP BY COMMAND SUBROUTINE BLOK
MSZ	-SIZE (EXAMPLE -4096) SET UP BY COMMAND SUBROUTINE BLOK
MSZH	-(SIZE/2) (EXAMPLE -2048) SET UP BY COMMAND SUBROUTINE BLOK
POW	LOG2 (SIZE) (EXAMPLE 12) SET UP BY COMMAND SUBROUTINE BLOK

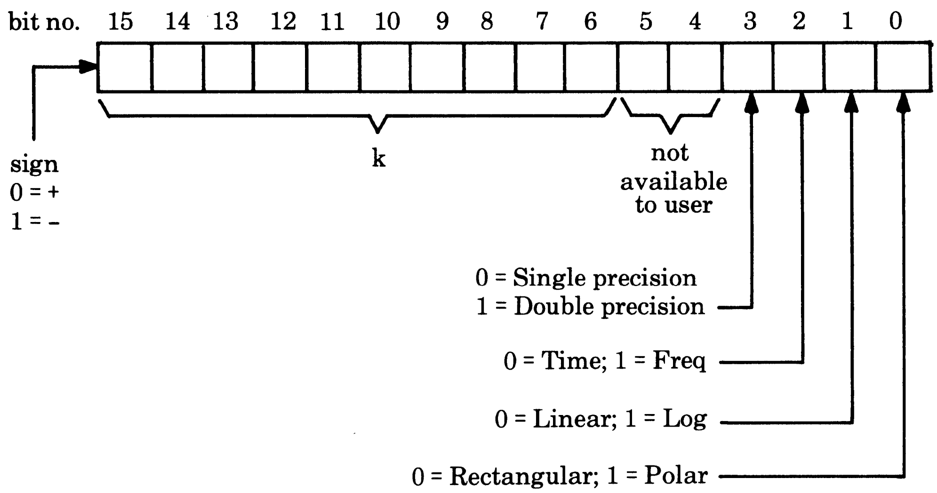
The following three BSS statements always have to be continuous, together they establish the data qualifier storage (see following paragraphs). Each BSS statement is BSS (NMAX0).

SCAL0	BSS STATEMENT ESTABLISHING SCALE FACTOR — MODE NUMBER STORAGE, ALSO CONTAINS SCALE FACTOR OF BLOCK 0 (EXAMPLE 128)
CALB0	BSS STATEMENT ESTABLISHING CALIBRATOR STORAGE, ALSO CONTAINS CALIBRATOR OF BLOCK 0 (EXAMPLE 128)
FREQ0	BSS STATEMENT ESTABLISHING FREQUENCY FACTOR STORAGE, ALSO CONTAINS FREQUENCY FACTOR OF BLOCK 0 (EXAMPLE 128)
FREQD	FINAL ADDRESS+1 OF DATA QUALIFIER STORAGE
ZIFFR	COUNTER USED TO DETERMINE LOCATION OF BLOCK N DATA QUALIFIERS. SETUP BY COMMAND SUBROUTINE BLOK. INITIALLY SET EQUAL TO -(NMAX0/NMAX). (EXAMPLE -64)

Note: Subroutines LOCAT & BLOK are described in 5451A System Software Manual.

Scale Factor Word

With each data block, three Computer words are associated: one contains the scale factor and coordinate code, another contains the block calibrator, and a third contains the frequency code. All scale factors, calibrators, and frequency codes are contained in the data qualifier storage described in the previous paragraph. The format of the scale factor word is as follows:



The scale factor is the "k" which immediately follows the SF at the beginning of any punched tape input or output. It is the floating point power, in radix 10, of the corresponding data block. Thus, for example, if the block has a scale factor of 10^{10}, then bits 7 and 9 would be 1's, and the remainder of bits 6 through 15 would be 0's. The coordinate code is given in bits 0, 1, 2, and 3 of the scale factor word. Note that these bits spell out in binary the coordinate code which we have seen throughout this manual:

Coordinate Code	Time	Frequency	Rectangular	Polar	Log	Linear	Single Precision	Double Precision
0	x		x			x	x	
2	x		x		x		x	
4		x	x			x	x	
5		x		x		x	x	
7		x		x	x		x	
12		x	x			x		x
14		x	x		x			x

Thus, time rectangular linear coordinates are expressed by a coordinate code of 0, or 0's in bits 0, 1, 2, and 3; frequency rectangular linear coordinates are expressed by a coordinate of 4, or a 1 in bit 2 and 0's in bits 0, 1, and 4. Regardless of what shifting of the scale factor the user might do, he should be certain he does not in any way truncate the 10 bits or improperly modify the coordinate code.

Block Calibrator Word

The calibrator of each data block is a 16-bit binary number which multiplies all N data words in the block. For data blocks in linear coordinates, the block calibrator is always a positive number whose value should always be kept between 1 (32,767) and .1 (3,277). For log coordinate data blocks the calibrator can be between 0 (log 1) and -6554 (log .1).

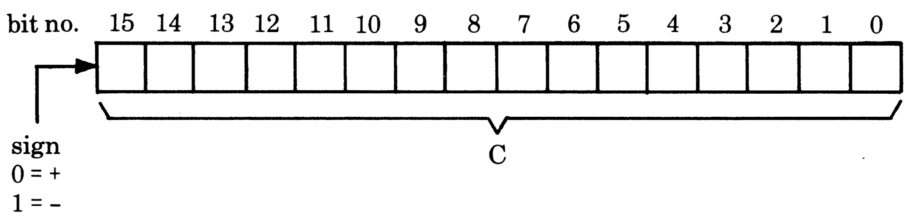
The calibrator and the scale factor of a data block constitute the complete calibration of that data block. Every data word D_i of a data block has a value of:

Linear domain: $C \times D_i \times 10^k$

Log Domain: $\text{Log}(C) + \text{Log}(D_i) + k$

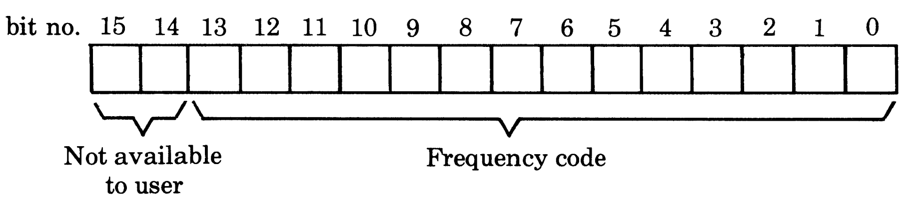
where:

C = Block Calibrator
k = Scale Factor
D = Data Word



Frequency Code Word

The frequency code word is a 16-bit word with a format as follows.



Data qualifiers

The frequency codes have been seen at various places throughout this manual, for example page 6-7. The code has its principal application when data has been taken in by the ADC. In this case, it gives the SAMPLE MODE and MULTIPLIER switch settings. The user, however, can enter any number from 0 through +16,383 for his own identification purposes when he is inputting data via paper tape or via manual Keyboard entry. The frequency code stays with the data block throughout all operations on it, and is never modified.

WRITING ADC DRIVERS

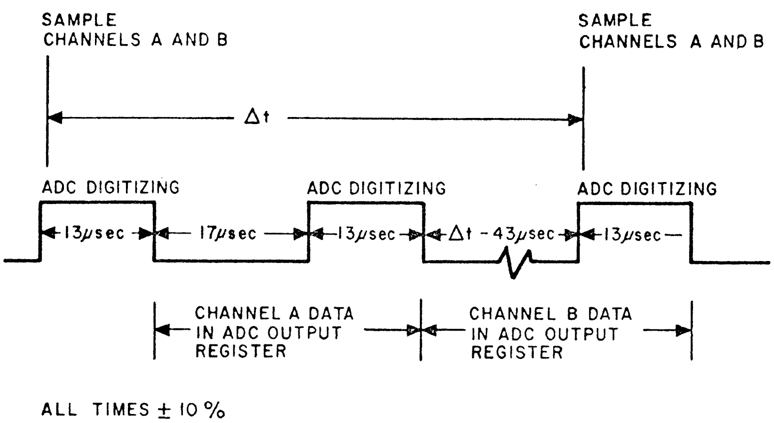
The user can write his own driver to utilize the ADC. The following information will assist him.

The ADC interfaces with a microcircuit card in the Computer. The ADC is armed by issuing Set Control, Clear Flag commands to the ADC I/O channel. After the Set Control command is given and after the trigger circuit is triggered, the ADC will begin to send data points back to the Computer. In the LINE mode the ADC is triggered by the next line frequency zero crossing. In the INTERNAL mode it is triggered by the A channel input as determined by the SLOPE and LEVEL control settings on the ADC panel. In the FREE RUN mode the Set Control signal also triggers the ADC so that in this mode a data response comes back within 12 μsec of Set Control. In the EXTernal mode, the EXTernal trigger starts the process.

The Set Control command sets the ADC Encode. The microcircuit card is strapped so that the device flag from the ADC does not clear the Encode signal to the ADC. This means that once the ADC starts sending data, the only way it can be turned off is by a Clear Control command. When a Clear Control is received, the ADC does a self-calibration and cannot be turned on again in less than 50 μsec.

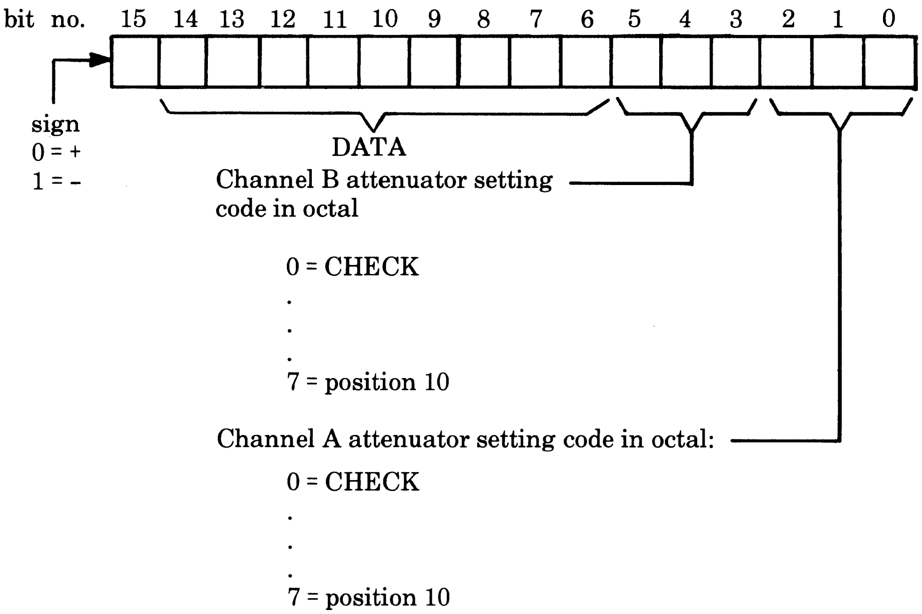
In the single channel mode, data points enter the microcircuit card register on the ADC at the rate set by the ADC sampling control. In the dual channel mode, data points enter the channel in pairs, first the A channel point, then the B channel point, then A, etc. The A word remains in the register 30 μsec ± 3, as shown in Figure 9-2. The B word enters the register and remains until the next A word. Thus, the distance between A words is the sample interval, Δt, as set by the ADC control, but the distance between the A word and the B word is a fixed quantity independent of the sampling rate.

Figure 9-2. ADC Sampling Ratios



ADC Data Word

The ADC data word is a 10-bit word in 2's complement form, the first bit being the sign.



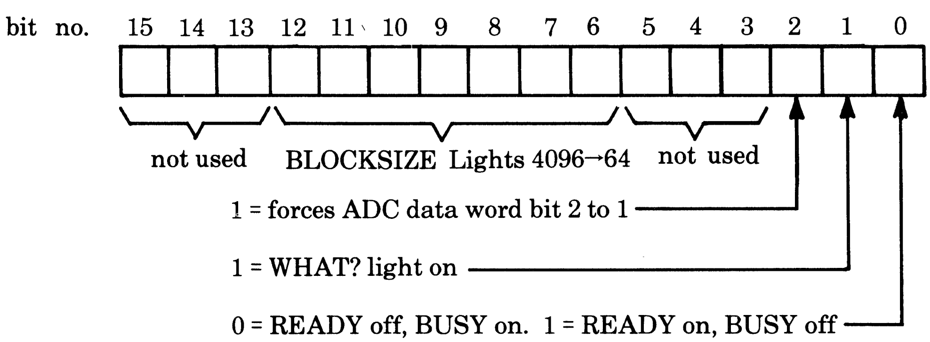
The attenuator setting code in bits 0 through 5 is transmitted with every ADC word, single or dual channel.

The ADC is adjusted so that 0 volts is on the edge between 0 and -1; that is, 0 volts has equal likelihood of producing all bits equal 0 or all bits equal 1 in the ADC word. A plus full-scale-overload is indicated by a 0 sign bit and all 1's in the data; a negative full scale overload is indicated by a 1 sign bit and all 0's.

ADC Code Word

The Keyboard lights (Blocksize 64-4096, READY, BUSY, WHAT?) are controlled by placing a code word in the ADC I/O channel. The format for this code word is as follows.

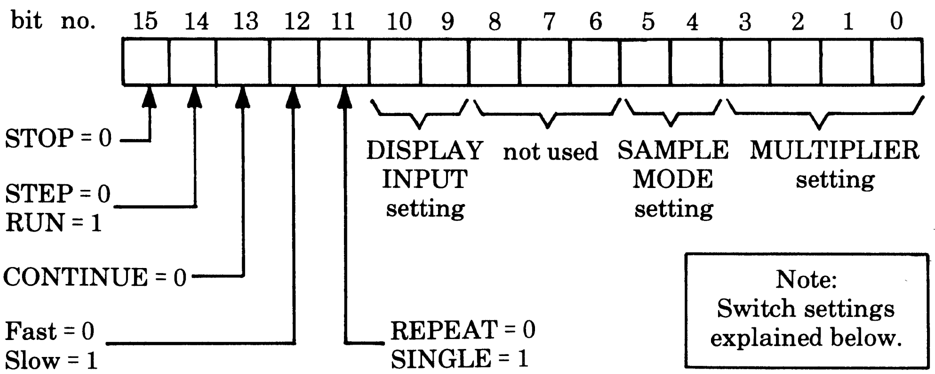
ADC driver

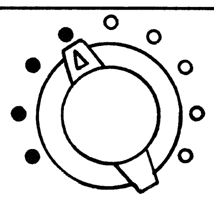


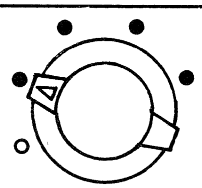
It is good practice to output blocksize (SIZE) to the ADC before ADC *attenuator setting code* is picked up from the ADC to ensure that bit 2 of the ADC data word is not forced to 1.

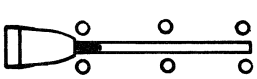
Keyboard Interface Word

The Keyboard switches STEP/RUN and REPEAT/SINGLE, pushbuttons STOP, CONTINUE, and the ADC SAMPLE MODE, MULTIPLIER, and DISPLAY INPUT controls are interfaced to the Computer using a Microcircuit card in the Keyboard I/O channel. This information is obtained by issuing an LIA (Load Into A) command to the Keyboard I/O channel. The Keyboard Interface Word from the I/O channel has the following format.



MULTIPLIER Setting	Bit 3	Bit 2	Bit 1	Bit 0
 50/100/10	0	0	0	0
25/50/20	0	0	0	1
10/20/50	0	0	1	0
5/10/100	0	0	1	1

SAMPLE MODE Setting	Bit 5	Bit 4
 (Δ TIME) kHz/μs	0	1
(Δ TIME) Hz/ms	1	1
(Δ FREQ) Hz/ms	0	0
(Δ FREQ) mHz/sec	1	0

DISPLAY INPUT Setting	Bit 10	Bit 9
 A/A	0	0
A/DUAL	1	0
B/DUAL	1	1

Keyboard interface

Bit 12 = 0 if SAMPLE MODE is at	Δ FREQ kHz/μs Δ TIME Hz/ms	Fast Rate	
AND			
MULTIPLIER is at	5/10/100 10/20/50 25/50/20 50/100/10		
Bit 12 = 1 if SAMPLE MODE is at	Δ FREQ Hz/ms Δ TIME mHz/sec	Slow Rate	

(MULTIPLIER = don't care)

Bit 13 = 0 if CONTINUE has been pressed and READY light is on.

Bit 15 = 0 if STOP has been pressed and BUSY light is on.

Bit 13 = 1 if BUSY light is on.

Bit 15 = 1 if READY light is on.

Obviously it is necessary to test for CONTINUE button depressed only when READY light is on, and conversely test for STOP button depressed only when BUSY light is on. In other words, after setting bit 0 of ADC Code Word to 1 (READY light on), test for bit 13 = 0 (CONTINUE pressed) in Keyboard Interface Word; after setting bit 0 of ADC Code Word to 0 (BUSY light on), test for bit 15 = 0 (STOP pressed) in Keyboard Interface Word.

WRITING DISPLAY UNIT DRIVERS

The user can also write his own Display Unit drivers. The following information will be of assistance.

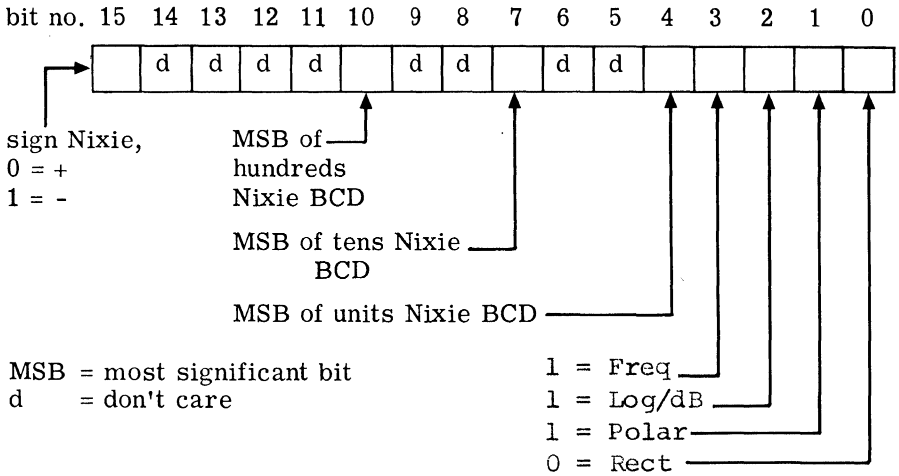
The Display Unit is interfaced through a microcircuit card. To output a word to the microcircuit card from the Computer, the Set Control, Clear Flag commands are given. Then 25 μsec are required to process the word and return the device flag, after which the Display Unit is ready for the next word.

Display Unit Word — From Computer

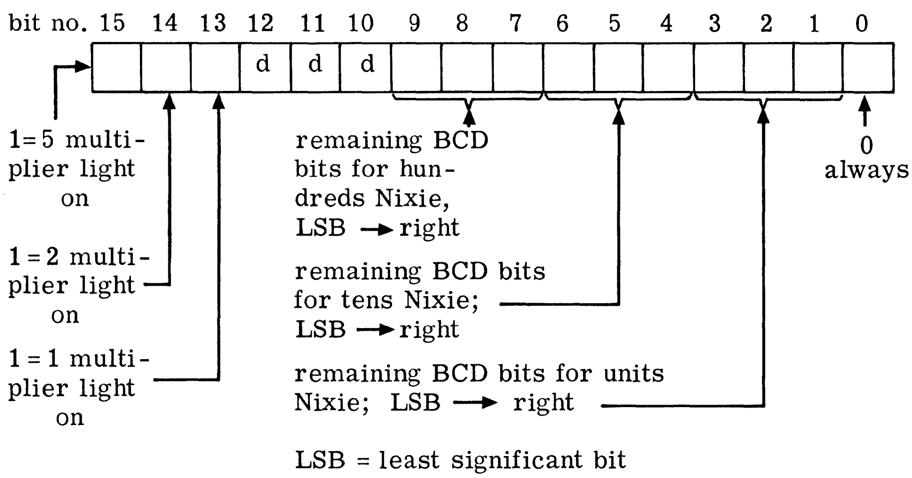
Display driver

Display Unit words coming from the Computer have the format shown on the next page. The first data word is identified by a 1 in bit 0. The first and second words contain scale factor information. The scale factor digits displayed on the Nixie tubes are defined in BCD: the most significant bits are in word 1, the remaining in word 2. The third word is then the first X point and the fourth is the first Y point; X's and Y's are then sent alternatively until the end of the data block. Each data word to the Computer is an 11-bit word in 2's complement form, left adjusted. The first bit is the sign; 00 is located in the center of the screen. When the ORIGIN switch is used to shift the origin to the left of the screen, this is done via software.

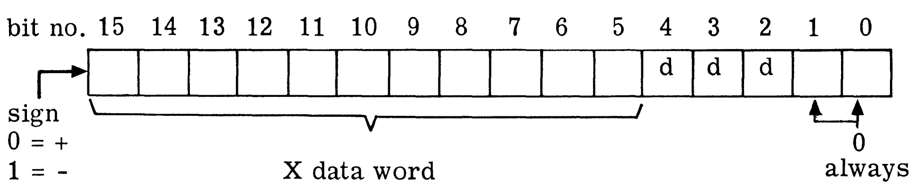
WORD 1



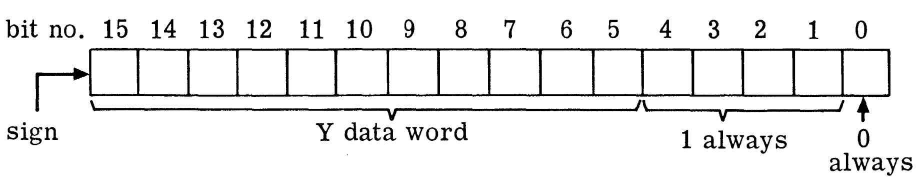
WORD 2



WORD 3

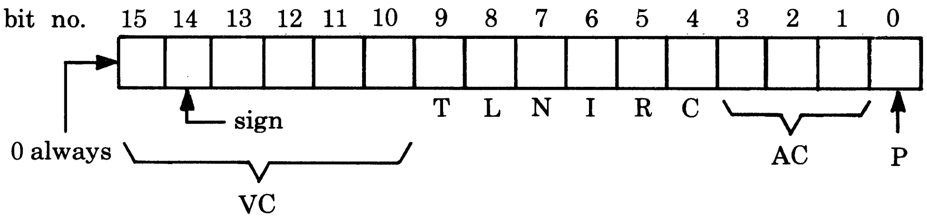


WORD 4

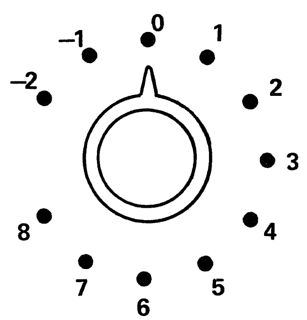


Display Unit Word — To Computer

When the Display Unit sends back the device flag it also sends a data word back indicating the Display Unit control settings. The format of this word is as shown below:



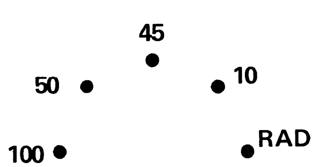
VC = VERTICAL SCALE switch change. Binary number equal to number of upscale or downscale step changes on SCALE switch.



VERTICAL SCALE Position	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10
–2	0	1	0	0	1	0
–1	0	1	0	0	0	1
0	0	0	0	0	0	0
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	0	0	1	0	0
5	0	0	0	1	0	1
6	0	0	0	1	1	0
7	0	0	0	1	1	1
8	0	0	1	0	0	0

If T = 1, origin at center of screen
If L = 1, horizontal log display
If N = 1, complex display
If I = 1, imaginary part display
If R = 1, real part display
If C = 1, origin at left of screen
If P = 1, plot

AC = POLAR ANG/DIV phase angle scale code.



POLAR ANG/DIV Position	Bit 3	Bit 2	Bit 1
100°/div	0	1	0
50°/div	1	0	0
45°/div	0	0	0
10°/div	0	0	1
RAD/div	1	1	0

Y COMMAND FORMAT

The Y command feature of the Fourier Analyzer gives the user the opportunity to write his own relocatable programs (Y commands) in FORTRAN or HP Assembly language, merge them with the 5451A System and execute them using the USER PROG key. The Y command format is:

[USER PROG] [N] [SPACE] [N1] [SPACE] [N2] [ENTER]

where:

> N is a positive number expressed as four digits, leading zeroes included (e.g. ØØØ1)
> N1 and N2 are optional program parameters that can have values from -32,767 to +32,767.

NOTE

> The HP User Program Library programs will be numbered in the range
>
> CØØØ thru C799
>
> where C is a classification code number from Ø thru 9. Therefore, for compatibility with the UPL system it is recommended that the user number his programs in the range C8ØØ thru C999.

To show how Y command programs can be written, a listing and description of a sample Assembly language user program follows:

```
ASMB,R,B,L
      NAM YNNNN    WHERE NNNN IS A 4 DIGIT NUMBER EXAMPLE Y0100
      ENT YNNNN
*
      EXT INS1     CONTAINS FIRST PARAMETER ENTERED : N1
      EXT INS2     CONTAINS SECOND PARAMETER ENTERED : N2
      EXT INSMX    CONTAINS NUMBER OF PARAMETERS ENTERED
*                  =0 : NO PARAMETERS ENTERED
*                  =1 : ONLY N1 ENTERED
*                  =2 : BOTH N1 & N2 ENTERED
      EXT SIZE     CURRENT VALUE OF BLOCKSIZE
      EXT DTAD0    STARTING ADDRESS OF DATA STORAGE & OF BLOCK 0
*
*
YNNNN NOP          ENTRY TO USER PROGRAM (Y COMMAND)
*
*
*
* BODY OF USER PROGRAM
*
*
*
*
EXIT  LDA YNNNN,I   MUST EXIT FROM USER PROGRAM (Y COMMAND)
      JMP 0,I            IN THIS MANNER
      END
```

A similar listing for a FORTRAN written Y command is shown as follows.

```
FTN,B,L
      SUBROUTINE YNNNN (I,JBUF,KBUF,L,MBUF)
C** WHERE NNNN IS A 4 DIGIT NUMBER EXAMPLE Y0100
C** I CONTAINS NUMBER OF PARAMETERS ENTERED
C** =0 : NO PARAMETERS ENTERED
C** =1 : ONLY N1 ENTERED
C** =2 : N1 & N2 ENTERED
C
C** JBUF(1) CONTAINS FIRST PARAMETER ENTERED : N1
C** JBUF(2) CONTAINS SECOND PARAMETER ENTERED : N2
C
C** KBUF(1) CONTAINS VALUE OF FIRST DATA POINT IN DATA STORAGE
C
C** L CONTAINS CURRENT VALUE OF BLOCKSIZE
C
C** MBUF(1) CONTAINS VALUE OF FIRST SCALE FACTOR IN
C            DATA QUALIFIER STORAGE
C
C
      DIMENSION JBUF(2),KBUF(1),MBUF(1)
C*
C*
C*
C*  BODY OF USER PROGRAM (YCOMMAND)
C*
C*
C*
      RETURN
      END
```

Since the 5451A System software consists of relocatable subroutines written in HP Assembly language, they are directly callable by user's program (Y command) if it is also written in Assembly language.

These relocatable subroutines do not pass parameters to each other when calling, therefore they are not directly FORTRAN callable.

To write effective user programs (Y commands) with minimum effort and to fully utilize the 5451A relocatable subroutines, the user should thoroughly familarize himself with the 5451A System Software Manual, which contains a complete description of the system software.

How To Write a User Program (Y Command)

a. Write the Y command in HP Assembly language or FORTRAN as a relocatable program. The program must have entry point YNNNN, where NNNN is a 4-digit positive number (e.g. Ø8ØØ) excluding numbers of resident Y commands.

Exit from the program must be done as shown in the following example:

Writing Y commands

```
ASMB,R,B,L
      NAM Y0800
      ENT Y0800
*
Y0800 NOP
*
*
      LDA Y0800,I        MUST EXIT FROM Y COMMAND
      JMP 0,I                 IN THIS MANNER
      END
```

b. To obtain the optional program parameters, N1 and N2, entered from the Keyboard when the Y command is called, three external labels are used. Their contents are as follows:

No. of Parameters Entered	INSMX	INS1	INS2
2	2	N1	N2
1	1	N1	Undefined
0	0	Undefined	Undefined

It is necessary to test label INSMX before using INS1 and INS2. The following example shows how INSMX may be tested.

```
ASMB,R,B,L
      NAM Y0800
      ENT Y0800      MUST BE A 4 DIGIT NUMBER
*
      EXT INSMX
      EXT INS1
      EXT INS2
*
Y0800 NOP
      LDA INSMX
      SZA,RSS       DOES INSMX =0
      JMP NONE      YES, THEREFORE NO PARAMETERS WERE ENTERED
      CLB,INB
      CPA 1         DOES INSMX =1
      JMP ONE       YES, ONLY PARAMETER N1 WAS ENTERED
      JMP TWO       INSMX =2, BOTH N1 & N2 WERE ENTERED
      .
      .
```

c. If the Y command is to access or modify data, see previous Data Storage paragraphs for organization of data storage and associated parameters defined by program NAM DTADO.

d. Assemble the user program to produce a relocatable object tape.

Creating Y Command Overlays

To merge the user-written Y command with the 5451A System, an absolute overlay containing the Y command must be made. The following procedure describes how to generate and use the overlay.

1. Load the System Overlay Generator (SOG) tape using the Basic Binary Loader.

2. Turn Teleprinter switch to LINE.

3. Set starting address by pressing P button and entering octal number 002010 in the Computer DISPLAY REGISTER. Push INTERNAL PRESET, EXTERNAL PRESET, and RUN.

NOTE

In a system where the Teletype is used in place of a Photoreader or Punch, the following HALTS will occur during the overlay procedure. Described below is the action that should be taken on the Teletype when these HALTS occur.

M Register Contents	Action Required on Teletype
102055 = HALT 55	Punch Off Reader Off
102056 = HALT 56	Punch On Reader On (If *LOAD)

4. The Teleprinter will print the following message:

   TURN ON PUNCH

   SET SW15 TO SUPPRESS EXTRANEOUS PRINTOUT

   ENTER USER COMMAND NUMBERS, OR TYPE "?"

5. Turn on Punch and respond to message by typing 4-digit Y command numbers, one per line. Type /E as last line. If for example the user command were YØ8ØØ, the response would be:

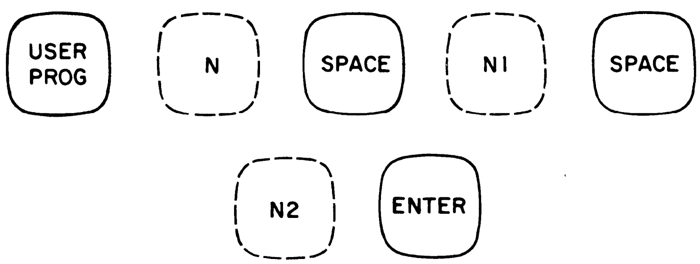
```
Ø8ØØ
/E
```

6. When Teleprinter prints message "*LOAD" then place relocatable Y-command object tape in Photoreader. If the Y-command object tapes are on a library tape, the software will selectively load only those listed in step 5.

7. Press L and RETURN keys on Teleprinter. The object tape will be read in. If all EXT symbols have been defined the system overlay will be punched and the overlay list printed out. If some remain undefined, the Teleprinter will respond with message "*LOAD"; then press U and RETURN keys to determine what EXT symbols remain undefined. The programs containing the undefined symbols must then be loaded.

8. When Teleprinter prints message "*END" then generation of system overlay is finished.

9. Load the standard 5451A program tape in the standard way (see Section II). Then load the system overlay just generated in the same way. The Fourier Analyzer will now operate and the user-written Y commands may be executed using the Y command format:



where:

N is a positive number expressed as four digits, leading zeroes included (e.g. 0001)
N1 and N2 are optional program parameters that can have values from -32,767 to +32,767.

Resident Y Commands

The following Y Commands are resident in all 5451A Fourier Analyzers.

USER PROG **3010**

Under normal conditions a data block is upscaled (maximized) before and after execution of a command to maintain full accuracy during calculations. This process entails doubling each data word in the data block until further doubling would cause overflow. Calibration of the data block is maintained by adjusting the block calibrator and scale factor each time the data word is doubled. Y Command 3010 allows the user to *disable or enable* the maximization process before execution of a command. The command format is:

USER PROG | 3010 | SPACE | N1 | ENTER

where:

N1 = 1 to *disable* maximization. Default of N1 *enables* maximization.

USER PROG **3011**

Under normal conditions when a self-conjugate multiply is performed on a frequency domain data block, a double precision block is created. Y Command 3011 allows the user to *disable or enable* double precision block creation, before execution of a self-conjugate multiply command. Existing double precision data blocks are unaffected. The command format is:

USER PROG | 3011 | SPACE | N1 | ENTER

where:

N1 = 1 to *disable* double precision block creation.
Default of N1 *enables* double precision block creation.

USER PROG 3012

This command allows the user to specify the maximum number of scaledowns that could occur in Block Division. In the worst case, this means that the quotient could be scaled down (divided by 2) as many times as specified by the maximum number of scaledowns. To specify a maximum scaledown number, enter this command before performing Block Division.

The command format is:

USER PROG | 3012 | SPACE | N1 | ENTER

where:

N1 is the number of scaledowns ranging from 1 to 16.
Default of N1 not allowed.

A NOTE ON THE INTERRUPT SYSTEM

In all Fourier programs, the interrupt system is always on when the Fourier program is on. If, in the course of a user Assembler language program, the interrupt system is turned off, then before control is returned to the Fourier program, all I/O channels must be turned off (with the CLC, 00 command) and the interrupt turned on (with the STF, 00 command).

APPENDIX A: TELEPRINTER SYMBOLS

The symbols listed in this table represent codes for the Keyboard commands. These symbols are typed on the Teleprinter whenever a command is entered from the Keyboard, or when listings are requested. By the same token, commands can be entered from the Teleprinter by typing these symbols.

b = space after symbol

Teleprinter Symbol	Keyboard Command
+ A+	Block Add: Add block N1 to block 0. Leave the result in block 0. + N1 ENTER ①
− A-	Block Subtract: Subtract block N1 from block 0. Leave the result in 0. -N1 ENTER
BLOCK SIZE BS	Block Size: Set block size to N1. BLOCK SIZE N1 ENTER
TRANS FCN CH	Transfer Function: Transfer and coherence function. N1 is first data block, N2 specifies dual input. TRANS FCN N1 SPACE N2 ENTER
CLEAR CL	Clear Total Block: Clear block N1. CLEAR N1 ENTER Clear Partial Block: Clear block N1 from channel N2 to N3. CLEAR N1 SPACE N2 SPACE N3 ENTER
CORR CR	Correlation: Correlate block 0 with block N1 and leave the result in block 0. CORR N1 ENTER
CONV CV	Convolution: Convolve block 0 with block N1 and leave the result in block 0. CONV N1 ENTER
DELET /D	Deleting Program Steps: Delete steps N1 through N2. DELET N1 SPACE N2 ENTER
DSPLY Db	Display: Display block N1. DSPLAY N1 ENTER
F Fb	Fourier Transform: Fourier transform blocks N1 and N2. F N1 SPACE N2 ENTER

① Default values for Keyboard commands are not shown.

TELEPRINTER SYMBOLS (Cont'd)

Teleprinter Symbol	Keyboard Command
HANN H1	Interval-centered Hanning Window: Hann block N1 with the interval-centered Hanning window. HANN N1 ENTER
INSRT /I	Inserting Program Steps: Insert after step N1 any number of steps. INSRT N1 ENTER
SKIP IF	SKIP; Increment program pointer if any data point in block N1 from channel N2 to N3 is negative. SKIP N1 SPACE N2 SPACE N3 ENTER
JUMP Jb	Jump: Jump unconditionally to label N1. JUMP N1 ENTER
KEY BOARD Kb	Keyboard Input Command, Block Fill: Keyboard one channel value into block N1 from channels N2 to N3. KEYBOARD N1 SPACE N2 SPACE N3 ENTER Keyboard Input Command, Point-by-Point Fill: Keyboard successive values into block N1 starting with channel N2. KEYBOARD N1 SPACE N2 ENTER Keyboard Scale Factor Input: Input amplitude scale factor N1, coordinate code N2, frequency code N3. KEYBOARD N1 SPACE N2 SPACE N3 ENTER Keyboard Data Input, Time: Input real part, N1 N1 ENTER Keyboard Data Input, Frequency: Input real part/magnitude, N1; imaginary part/phase N2. N1 SPACE N2 ENTER
LIST /L	List an Entire Program from Line 0: LIST ENTER List a Single Line, N1: LIST N1 ENTER

TELEPRINTER SYMBOLS (Cont'd)

Teleprinter Symbol	Keyboard Command
LIST /L (Cont'd)	List a Range of Lines from Line N1 to Line N2: LIST N1 SPACE N2 ENTER List from a Certain Line N1 to the End of the Program: LIST N1 SPACE (99 for 8K machines, 199 for 16K machines) ENTER
LABEL Lb	Label: Label N1, where N1 is an integer from -32,767 to +32,767. LABEL N1 ENTER
MASS STORE MS (optional)	Mass Store: Execute data transfer between optional storage devices.
COUNT #b	Loop: Loop back to label N, N1 times. COUNT N SPACE N1 SPACE N2 ENTER
PUNCH Pb	Punch Output: Punch out block N1 from channel N2 to N3. PUNCH N1 SPACE N2 SPACE N3 ENTER
ANALG IN RA	Analog Input: Analog input into block N1. Display block N2. ANALG IN N1 SPACE N2 ENTER
BUFFD ANALG RB	Buffered Analog Input: Analog input into block N1. Display block N2. BUFFD ANALG N1 SPACE N2 ENTER
RPLAC /R	Replacing Program Steps: Replace step N1 through N2 with any number of steps. RPLAC N1 SPACE N2 ENTER
PHOTO READR Rb	Photoreader Input: Input data via Photoreader into block N1 from channels N2 to N3. PHOTOREADR N1 SPACE N2 SPACE N3 ENTER
HISTO GRAM RH	Histogram: Histogram block 0 from channels N2 to N3 into block N1. HISTOGRAM N1 SPACE N2 SPACE N3 ENTER
POWER SPECT SP	Power Spectrum: Compute input, output, and cross power spectra. N1 is first data block, N2 specifies single/dual. POWER SPECT N1 SPACE N2 ENTER

TELEPRINTER SYMBOLS (Cont'd)

Teleprinter Symbol		Keyboard Command
LOG MAG	TL	Log Magnitude: Convert the magnitude (or real part) of block N1 to log. LOG MAG N1 ENTER
POLAR	TP	Rectangular to Polar Coordinate Change: Convert block N1 from rectangular to linear polar coordinates. POLAR N1 ENTER
RECT	TR	Polar to Rectangular Coordinate Change: Convert block N1 to linear rectangular coordinates. RECT N1 ENTER
PRINT	Wb	Print Output: Print out block N1 from channels N2 to N3. PRINT N1 SPACE N2 SPACE N3 ENTER
INTER CHNG	Xb	Interchange: Interchange block 0 with block N1. INTERCHNG N1 ENTER
STORE	X>	Store: Store data block 0 in block N1. STORE N1 ENTER
LOAD	X<	Load: Load data block N1 into block 0. LOAD N1 ENTER
USER PROG	Yb	User Program: Execute user Y command N1 with parameters N2 and N3. USER PROG N1 SPACE N2 SPACE N3 ENTER
÷	:b	Block Divide: Divide block 0 by block N1. Leave the result in 0. ÷ N1 ENTER Integer Divide: Divide block N1 by the integer N2 (N2 positive only). ÷ N1 SPACE N2 ENTER
MULT	*b	Block Multiply: Multiply block 0 by block 1. Leave the result in 0. MULT N1 ENTER Integer Multiply: Multiply block N1 by the integer N2. MULT N1 SPACE N2 ENTER

TELEPRINTER SYMBOLS (Cont'd)

Teleprinter Symbol	Keyboard Command
*b (Cont'd) MULT	Loop Counter Multiply: Multiply block N1 by the loop counter integer. MULT N1 SPACE 0 ENTER Complex Multiply: Multiply block N1 by a complex number (N2+j N3). MULT N1 SPACE N2 SPACE N3 ENTER
*— * MULT	Block Conjugate Multiply: Multiply block 0 by the complex conjugate of block N1. Leave the result in 0. *MULT N1 ENTER Conjugate: change the sign of the imaginary part of each channel. *MULT N1 SPACE 0 ENTER
←b ←	Rotate or Shift: Rotate or shift block N1, N2 channels to the left. ← N1 SPACE N2 ENTER
$b ∫	Integrate: Integrate block N1 from channel N2 to N3. Result in block N1. ∫ N1 SPACE N2 SPACE N3 ENTER
%b d/dx	Differentiate: Differentiate block N1 from channel N2 to channel N3. $\frac{d}{dx}$ N1 SPACE N2 SPACE N3 ENTER
<b SUB RTRN	Sub-Routine Return: Return to next program step after sub-routine jump step. SUB RTRN ENTER
.b END	Program End: End of program. END ENTER
?b ?	Interrogate Pointer: List pointer location. ? ENTER
/b TERM	Terminate an Edit or Keyboard Entry: Terminate and close edit file. TERM ENTER
/. POINT	Pointer: Set program pointer to line N1. POINT N1 ENTER

APPENDIX B: WHAT? MESSAGES ON TELEPRINTER

When the WHAT? light on the Keyboard comes on, this is an indication that an illegal command has been given. In addition, if the Teleprinter switch is set to LINE, an error message will be typed. The most common causes of error are as follows:

1. N1, N2, or N3 in the command is an illegal block or channel for the block size set.
2. The symbol used for element 0 is incorrect. This would apply primarily to the keying-in of commands from the Teleprinter rather than from the Fourier Analyzer Keyboard.

Some examples of WHAT? messages are given below in alphabetical order by the identifying symbol.

Printout on Teleprinter

BS WHAT? (BLOCK SIZE)

N1 is an illegal block size.

CH WHAT? (TRANS FCN)

Insufficient blocks available for execution of command.

Block N1 given as 0 for dual channel operation.

/D WHAT? (DELET)

Line N1 non-existent.

H1 WHAT? (HANN)

Block N1 not in time domain.

IF WHAT? (SKIP)

Not in programming mode.

/I WHAT? (INSRT)

Line N1 non-existent.

Line N1 is last line in program stack.

J WHAT? (JUMP)

Label N1 does not exist.

Printout on Teleprinter (Continued)

/L WHAT? (LIST)

Line N1 does not exist.

RA WHAT? (ANALG IN)

Display block N2 does not exist.

RB WHAT? (BUFFD ANALG)

Display block N2 does not exist.

Buffer blocks overlap or are non-existent.

Loop counter missing in program stack.

RH WHAT? (HISTOGRAM)

Block 0 given as block N1.

Block 0 not in time domain.

/R WHAT? (RPLAC)

Line N1 does not exist.

SP WHAT? (POWER SPECT)

Insufficient blocks available for execution of command.

Block N1 given as 0 for dual channel operation.

:WHAT? (÷)

Divisor block N1 equals zero.

#WHAT? (COUNT)

Label N1 does not exist.

<WHAT? (SUB RTRN)

Not in programming mode.

←WHAT? (←)

N2 larger or equal to block size.

•WHAT?

End statement missing in program stack.

Pointer is set outside of program stack.

APPENDIX C: INTERFACE CABLE CONNECTIONS

Plug-in or Peripheral	Connector	Cable No.	I/O Card Description*	Remarks
5460A Display Unit w/ 180AR Oscilloscope H51	J113A/B	05450-60002	12566A/B .... Display Microcircuit Interface	Dual cable also connected to Control Unit J18.
5465A Analog-to-Digital Converter, Option 003	J12	05451-60002	12566A/B .... ADC Microcircuit Interface	Connector on Control Unit rear panel.
5475A Control Unit, Option 003	J11A/B	05451-60001	12566A/B .... Keyboard Microcircuit Interface	Connector on Control Unit rear panel, J11A unused.
2752A/2754B Teleprinter, Option 003	J14	05450-60004	12531C .... Teleprinter Input/Output	Control Unit J13 connects to Teleprinter
2748A/2758A High Speed Photoreader	—	12597-6004	12597A (Opt. 002) ...... Paper Tape Input	Optional. No connection to Control Unit.
2753A/2895A/8100A High Speed Punch	—	12597-6005	12597A (Opt. 003) ...... Paper Tape Output	Optional. No connection to Control Unit.

*Computer I/O slot numbers are identified on inside top cover.

INDEX

-A-

-B-

-C-

INDEX (Cont'd)

-C- (Cont'd)

INDEX (Cont'd)

-D- (Cont'd)

INDEX (Cont'd)

-G-

-H-

-I-

-J-

-K-

-L-

INDEX (Cont'd)

-L- (Cont'd)

INDEX (Cont'd)

-P- (Cont'd)

-R-

INDEX (Cont'd)

-R- (Cont'd)

INDEX (Cont'd)

-T- (Cont'd)

INDEX (Cont'd)

SYMBOLS

HEWLETT hp PACKARD

05451-90005

PRINTED IN U.S.A.