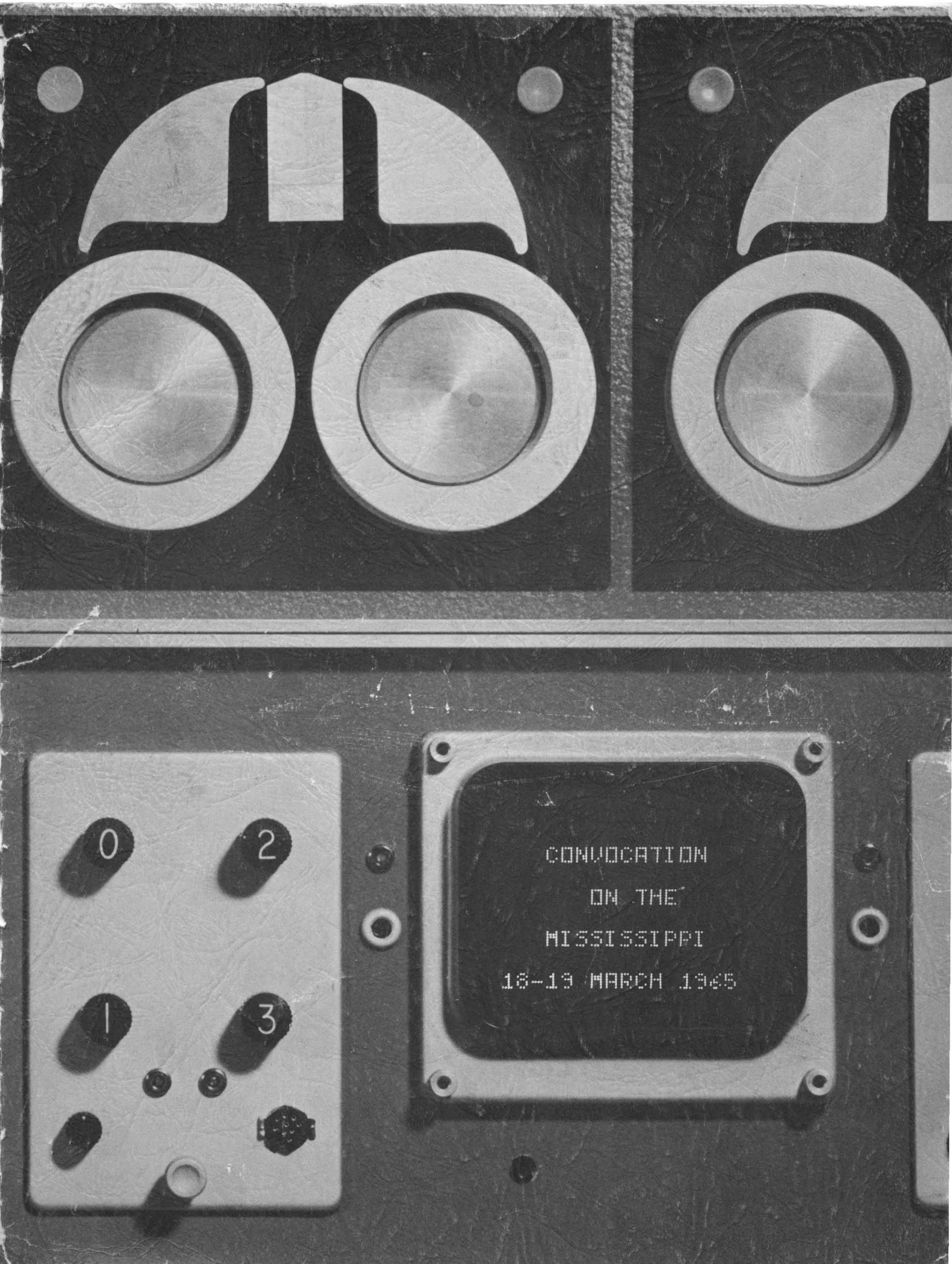


CONVOCATION
ON THE
MISSISSIPPI

18-19 MARCH 1965

A Summary Description

and

Proceedings

of the

Final LINC Evaluation Program Meeting

March 18-19, 1965

Washington University
St. Louis, Missouri

The conference reported in this volume was partially supported by the National Institutes of Health under Grant FR 00218. The work of the various participants reporting was supported by grants from the National Institutes of Mental Health and the Division of Research Facilities and Resources, National Institutes of Health. The LINCs used in these studies were funded by the National Institutes of Health with the cooperation and support of the National Aeronautics and Space Administration under NIH contract PH 45-63-540.

Computer Research Laboratory
Washington University

TABLE OF CONTENTS

Conference Participants

Section 1 - Commentary

Section 2 - Report from Bowman Gray School of Medicine, Winston-Salem, N.C., by G. S. Malindzak and F. L. Thurstone.

Section 3 - Report from Brown University, Providence, R.I., by D. S. Blough and L. Marlowe.

Section 4 - Report from Communication Research Institute, Miami, Florida, by J. C. Lilly.

Section 5 - Report from Duke University, Durham, N.C., by C. A. Boneau.

Section 6 - Report from Johns Hopkins University School of Medicine, Baltimore, Maryland, by G. F. Poggio and V. B. Mountcastle.

Section 7 - Report from Johns Hopkins University School of Medicine, Baltimore, Maryland, by B. Weiss.

Section 8 - Report from Northwestern University Medical School, Chicago, Illinois, by F. S. Grodins and J. E. Randall.

Section 9 - Report from Presbyterian Hospital, Philadelphia, Pennsylvania, by E. O. Attinger and A. Anne.

Section 10 - Report from University of Washington, Seattle, Washington, by J. W. Woodbury and A. M. Gordon.

Section 11 - Report from Washington University School of Medicine, St. Louis, Missouri, by J. L. O'Leary.

Section 12 - Report from University of Wisconsin Medical School, Madison, Wisconsin, by J. E. Hind and C. D. Geisler.

Section 13 - Report from Stanford University, Palo Alto, California, by J. Lederberg and L. Hundley.

CONFERENCE PARTICIPANTS

Representing Laboratories Participating in the LINC Evaluation Program:

ATTINGER, E. O. and A. ANNE, Research Institute, Presbyterian Hospital, Philadelphia.

BLOUGH, D. S. and L. MARLOWE, Department of Psychology, Brown University, Providence.

BONEAU, C. A., Department of Psychology, Duke University, Durham.

GRODINS, F. S. and J. E. RANDALL, Department of Physiology, Northwestern University Medical School, Chicago.

HIND, J. E., C. D. GEISLER and J. KEENAN, Laboratory of Neurophysiology, University of Wisconsin Medical School, Madison.

HUNDLEY, L. L., Department of Genetics, Stanford University, Palo Alto.

LILLY, J. C. and F. GRISSMAN, Communication Research Institute, Miami.

MALINDZAK, G. S. and F. L. THURSTONE, Department of Physiology, Bowman Gray School of Medicine, Winston-Salem.

POGGIO, G. F. and G. WERNER, Department of Physiology, Johns Hopkins University School of Medicine, Baltimore.

WEISS, B., Department of Pharmacology and Experimental Therapeutics, Johns Hopkins University School of Medicine, Baltimore.

WOODBURY, J. W. and A. M. GORDON, Department of Physiology, University of Washington, Seattle.

WURTZ, R. H. and L. SIMPSON, Department of Neurology, Washington University School of Medicine, St. Louis.

Representing the LINC Evaluation Board:

COX, J. R., Washington University
DEMPSTER, E. R., University of California
PIPBERGER, H. V., Veterans Administration
ROSENBERG, M. D., Rockefeller Institute
SHIPTON, H. W., State University of Iowa

Representing Other Laboratories Instrumented with LINCs:

BOURIS, William, Stanford University
BROWN, R., Massachusetts Eye and Ear Infirmary
BRYAN, J. S., National Institutes of Health
COULTER, N. A., Ohio State University
DAVIS, H., Central Institute for the Deaf
ELDREDGE, D. H., Central Institute for the Deaf
GERSTEIN, G. L., University of Pennsylvania
HANCE, A. J., Stanford University
HARTMAN, C., Air Force Cambridge Research Laboratories
LANGBEIN, D., Massachusetts Eye and Ear Infirmary
LATIMER, C. N., Lederle Laboratories
MOLNAR, C. E., Air Force Cambridge Research Laboratories
SHEFFIELD, H., Lederle Laboratories
SHERRIFF, W., National Institutes of Health
STEWART, D. L., Air Force Cambridge Research Laboratories

Representing the National Institutes of Health:

WAXMAN, B. D., Division of Research Facilities and Resources
WIENCKOWSKI, L., National Institute of Mental Health

Representing Manufacturers:

FELLOWS, G., Spear, Inc.
FOLEY, J. P., Spear, Inc.
HINDLE, W., Digital Equipment Corporation
KLAUSMEIER, H., S. Sterling Co.
POTTER, T., S. Sterling Co.
RUDERMAN, M. E., Digital Equipment Corporation

Representing the Washington University Computer Research Laboratory and Biomedical Computer Laboratory:

CLARK, W. A.
ELLIS, R. A.
ENGEBRETSON, A. M.
GERTH, V. W.
GLAESER, D. H.
LEWIS, H. C.
LEWIS, K. W.
McDONALD, M. D.
NOLLE, F.
ORNSTEIN, S. M.
PAPIAN, W. N.
PASSERO, S. L.
SANDEL, T. T.
STUCKI, M. J.
TOWLER, C. F.
TOWLER, J. C.
WILKES, M. A.

Representing Washington University

School of Medicine:

BECKER, B., Ophthalmology
BEHRER, M. R., Pediatrics
BROWN, D., Biological Chemistry
BUTCHER, H. R., Surgery
LAATSCH, R., Anatomy
SATTERFIELD, J. H., Psychiatry
SLEATOR, W., Physiology
TERNBERG, J. L., Surgery

Other Schools:

BUNCH, M. E., Psychology
DAMMKOEHLER, R. A., Computer Sciences and Applied Mathematics
DEPIREUX, J., Botany
HEISE, J., Botany
KAYUSHIN, Dr., Botany
LVOV, Dr., Botany
TOWNSEND, J., Physics
WHITLOW, G., Computer Sciences and Applied Mathematics
WOOLUM, J., Botany

Section 1

Commentary

The final meeting in the LINC Evaluation Program opened on the 18th of March in St. Louis, Missouri with clear skies and cold. The preceding day was one of almost unprecedented harsh weather. It speaks well of the motivation and perseverence of those participating and interested in the program that only 3 of a possible 65 failed to arrive before the meetings convened. Before the first day ended all were present.

All sessions were held in the Washington University School of Medicine Auditorium. Various aspects of interest in the LINC were represented by those attending the finale. Among them were representatives from all the laboratories participating in the evaluation program, LINC Evaluation Board members, LINC manufacturers' representatives, personnel from the granting institutions within NIH, representatives of laboratories possessing and using LINCs, local observers from Washington University, and staff members of the Computer Research and the Biomedical Computer Laboratories at Washington University. A roster of those attending is appended to this report.

The general format of these sessions was the same as those at the first inclusive gathering of participants in Portsmouth, New Hampshire, in June 1964. During the first day participants in the program made formal presentations describing for their colleagues and Board members the progress made in their research since their last report. A LINC was provided and used for demonstrations with these talks. The sessions were chaired by T. Sandel of the Washington University Computer Research Laboratory.

Reports were heard from all participating laboratories except the Johns Hopkins University School of Medicine's Department of Physiology (Drs. Poggio and Werner were delayed in arriving by the weather). The written reports submitted by the various laboratories appear in the following section.

The second day's activities were chaired by W. A. Clark of the Computer Research Laboratory. During these sessions reports were heard from various LINC users not formally a part of the Evaluation Program, but none-the-less interested in sharing information and exchanging experiences with other LINC users. It is heartening to note that the use of LINC seems to provide a solid bond between persons in very disparate disciplines; a common language with respect to computation bridges whatever interdisciplinary gaps may exist.

The activities of the two days were not totally devoid of lighter moments. We gathered from various informal remarks that merely being a resident at the Parkway House Motor Hotel must have constituted a considerable adventure. And, of course, true to our traditions as with the assembly sessions and our meeting in Portsmouth a social gathering was arranged. In this case, our host was Washington University who honored us with dinner following a pay-as-you-go cocktail hour. The meal was highlighted by a provocative talk by the Provost of Washington University, Dr. George E. Pake, who spoke on the role of the university developmental and scientific laboratory as a pathfinder for industry where ideas are so new and unproven as to make it impractical for industry to support new technical developments. The Evaluation Program would seem to provide substantial credence to the viewpoint expressed by Provost Pake. Dinner ended on a light note with a reading by the program's unofficial poet laureate, J. Walter Woodbury, of an original work (?) commissioned for the occasion.

In addition to the strictly formal aspects of the meeting, there were the usual cloakroom caucuses and other gatherings. Perhaps the most noteworthy of these was the spontaneous gathering of some of the participants at W. A. Clark's apartment following the banquet. At that time, a policy was evolved concerning the Computer Research Laboratory's responsibilities in future programming efforts with respect to LINC. It is described later.

The appended reports suffice to describe the first day's activities; the second day, however, did not consist of documented information and some comment is of interest.

J. R. Cox and M. D. McDonald of the Biomedical Computer Laboratory described and demonstrated their GUIDE utility program. Copies of the program on tape and descriptive materials were provided for those who wanted them. R. A. Ellis of the Computer Research Laboratory described and distributed tapes of various test programs he has written for LINC. Included were programs designed to exercise memory and to test instruction code operation.

Recommendations concerning maintenance and improvement of the performance of LINC tape units were presented by D. L. Stewart of the Air Force Cambridge Research Laboratories.

C. E. Molnar of the Air Force Cambridge Research Laboratories led a general discussion of engineering modifications. Among the proposals discussed was a suggestion that the machines be modified to operate programs out of the upper half of memory. Other proposals included a modification to allow automatic interruption of programs and a stated need for analog signal outputs. These and other proposals have been noted and are under study.

W. A. Clark commented on the CRL programming effort and announced the intention of the group to provide in the not-too-distant future a final double-memory assembly program, now being written by M. A. Wilkes. He summarized the discussion of the previous evening and stated that the CRL group would undertake to provide standard arithmetic subroutines for floating point and multiple precision operations. The participants felt that routines for addition, multiplication, division, and the generation of square roots, sines, cosines, logarithms, and exponential functions would be of greatest help to them. They noted that logical and experimental operations in each application were well enough understood so that each investigator could relatively easily write his own programs, and expressed the opinion that the lack of generality of such programs made it pointless for CRL to assume any responsibilities in that direction. All, however, expressed a strong desire to be informed of the existence, availability, and credibility of all programs written for the LINC. Suggestions were made that certain machine modifications might facilitate the proposed arithmetic subroutines. A final comment of interest was made. CRL has made the decision to divest itself of further technical and engineering responsibilities with respect to LINC. Such efforts should now be continued by interested manufacturers.

In addition to the various technical details discussed above, progress reports were heard from J. S. Bryan of the National Institutes of Health, D. H. Eldredge of the Central Institute for the Deaf, A. J. Hance of Stanford University, D. Langbein of the Massachusetts Eye and Ear Infirmary, and H. W. Shipton of the State University of Iowa. W. Sherriff of the National Institutes of Health showed a movie made with the use of LINC of a simulation of various kinds of activity in nerve.

It was the clear consensus of the participants that yearly meetings ought to be continued.

Concurrent with the technical sessions on the second day, the LINC Evaluation Board met with the NIH representatives to discuss progress in the program as it came to an end. Some of the topics they discussed are of interest.

The Board recognized that the concept of the laboratory computer as embodied in LINC and as demonstrated by the work of the participants was a significant conceptual and technical addition to research in the biomedical laboratory. They expressed regret that at the time of the initiation of the program that workers in areas such as biochemistry and molecular biology had not submitted proposals. They expressed the hope, however, that the pioneering efforts of the LINC Evaluation Program participants would play a catalytic role in future uses of computers in research in these other disciplines.

The Board noted with approval the salutory character of the do-it-yourself training program employed in the assembly phases. They recognized that within the temporal restraints of the program this was probably as efficient a use of training time as was possible.

The Board concurred that follow-up on use of the machines after three additional years in the laboratory would be desirable. They also suggested that a continuing bibliographic effort to encompass all work done with LINC be undertaken.

All the foregoing is provided to summarize the substance of our final meeting. In the closing moments of our brief congress, J. W. Woodbury graciously presented us of CRL with a properly executed and signed resolution thanking us for our efforts. In our closing now, we'd like to rejoin -- the pleasure was all ours!

RESOLUTION

Whereas the good people of CRL neé CDO have labored long fruitfully, cheerfully and unstintingly and even nights to provide functioning LINC's to sundry physiologists, psychologists and gadgeteers.

And whereas the aforesaid good people have carried on these labors with stout heart, a stiff upper lip and with a smile in the face of insuperable obstacles, obstinate LINC-lunks and even technical difficulties.

And whereas they have leaped these obstacles, soothed the savage breasts and solved technical difficulties with talent, tact and technique.

Be it, therefore, solemnly resolved that the undersigned LINC-lunks wish to hereby express their deep thanks and appreciation to these aforementioned good people of CRL not only for their perseverance, tact and talent, for their hard work in a cause in which they believed, for their unfailing good humor and good sense but, also for supplying the undersigned with one of the most challenging, interesting, and educational experiences of their lives by introducing them to an exciting and powerful new experimental tool and technique and opening up a new world of ideas, possibilities and expectations. With a rousing thanks and well done we attach our signatures and appreciation this 18th day of March 1965.

J. Walter Woodbury
Albert M. Gordon
C. Alan Boneau
Lloyd Marlowe
T. T. Handley
Donald S. Blough
George Gerstein

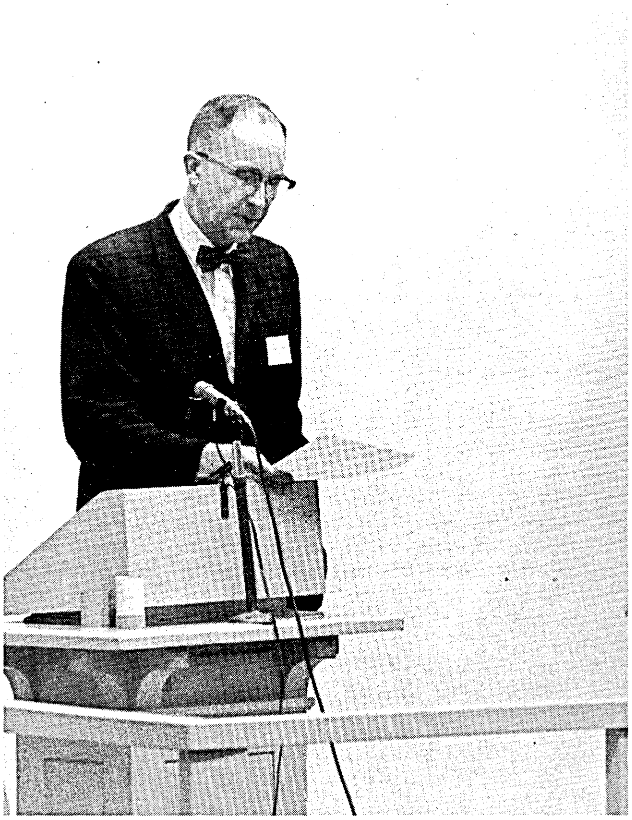
James E. Randall
Fred S. Grodins
Bill Bowring
N. A. Coulter Jr.
Sidney Goldring
Joe Hind

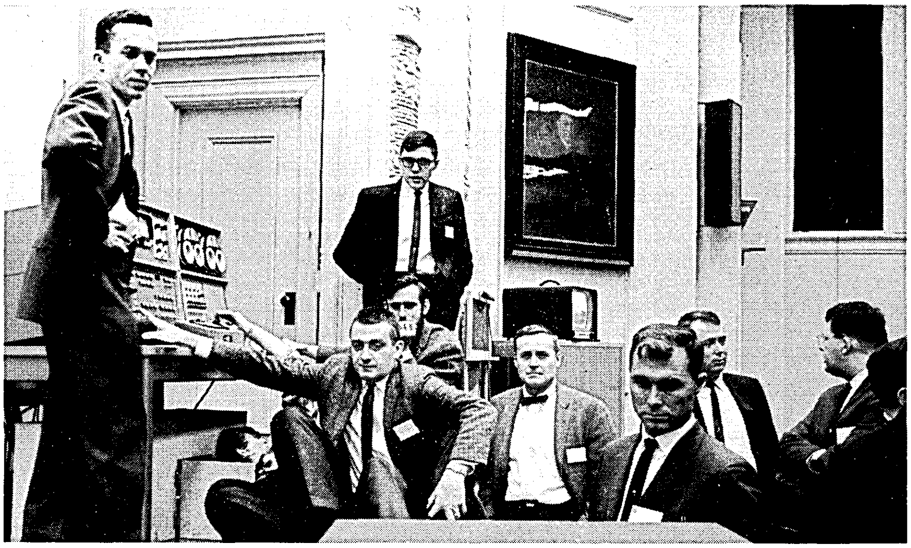
A. J. Hance
John C. Lilly
Donald H. Eldredge

CAVEAT

In some instances in the following reports there have been deletions of manuscripts for the sake of brevity and producing this document in a reasonable time scale. Where this has been done, a request for the particular manuscript will be promptly honored. In other cases, reproduction may be poor because of the state of the materials reproduced. In these cases, we can only ask your understanding. The only remedial actions we could have taken would have been inordinately time consuming.





LINC EVALUATION PROGRAM

Final Progress Report

George S. Malindzak, Jr., Ph.D.
Assistant Professor of Physiology
Bowman Gray School of Medicine
Winston-Salem, North Carolina 27103

Fredrick L. Thurstone, Ph.D.
Assistant Professor and Director of
Biomedical Engineering
Bowman Gray School of Medicine
Winston-Salem, North Carolina 27103

March 13, 1965

2-1

CONTENTS

Chapter	Title	Page
I	Research Proposal	1
II	Past and Present Research	8
III	Future Research	11
IV	Training Program	13
V	Computer Performance	15
VI	Log Book	18
VII	Bibliography	19

Appendices

I	LINC-026 Keypunch Interface	
II	Selected LINC Programs	

I. Research Proposal

Title: Studies Related to the Hydrodynamics and the Transmission Characteristics of the Arterial System.

A. Introduction

The specific aim of this project is to continue study of the hydrodynamics and transmission characteristics of the mammalian arterial system in terms of the derived model with particular reference to (a) the degree to which reflections of the arterial pressure pulse are present in the aorta and the extent to which the reflections distort and modify the pressure pulse contours, (b) the physical nature of the temporal and spatial variance of reflected energy in the aorta trunk and the establishment of a quantitative basis for its prediction, and (c) the effect that varying amounts of reflections have on over all cardiac performance in particular, the total work of the heart.

A method of numerical analysis based on Fourier Analysis of complex pressure pulses in the aorta of mammals has been derived that describes the transmission of these pressure pulses within the circulatory system. The theoretical analysis involves the calculation of incident and reflected wave parameters from time and space pressure pulse data measured along the axis of the vessel. Fourier Analysis makes possible the application of these procedures on individual harmonics.

B. Theoretical Analysis

The system considered in this analysis is simulated by the tube in which it is assumed that a pressure disturbance is propagated along this elastic tube from left to right (positive X direction). It is also assumed that there is negligible attentuation of this disturbance during propagation. Since the values of Δx used are 5 cm. or less, this is a reasonable assumption.

It is noted further that when the disturbance reaches the site of terminal impedance, a part of the energy of the disturbance is reflected back from the terminus, and appears as a retrograde traveling wave which is superimposed on the incident wave. For a single harmonic component of the pulse wave, the resultant pressure wave P(x,t) is given as:

$$P(x,t) = P_o + \sum_{1}^{n} I_n \sin \omega_n \left(t - \frac{x}{c_n} + \frac{\psi_n}{\omega_n}\right) + R_n \sin \omega_n \left(t + \frac{x}{c_n} + \frac{\phi_n}{\omega_n}\right) \qquad (1)$$

where $P(x,t)$ = pressure as a function of distance (x) and time (t), (mm Hg.)

P_o = mean pressure

I_n = modulus of incident harmonic component n (mm Hg.)

R_n = modulus of reflected harmonic n (mm Hg.)

ω_n = angular frequency (2 π) (radians/second)

c_n = velocity of propagation of harmonic n (meters/second)

ψ_n = arbitrary phase of Incident Wave

ϕ_n = arbitrary phase of Reflected Wave

This expression is fundamental to the entire derivation to follow. The actual pressure wave is the sum of the several harmonics comprising the wave, each expressed as in equation (1). Equation (1) may be differentiated with respect to time and distance as follows:

$$\frac{\partial P}{\partial t} = I\omega \cos \omega \left(t - \frac{x}{c} + \frac{\psi}{\omega}\right) + R\omega \cos \omega \left(t + \frac{x}{c} + \frac{\phi}{\omega}\right) \qquad (2)$$

$$\frac{\partial P}{\partial x} = \frac{-I\omega}{c} \cos \omega \left(t - \frac{x}{c} + \frac{\psi}{\omega}\right) + \frac{R\omega}{c} \cos \omega \left(t + \frac{x}{c} + \frac{\phi}{\omega}\right) \qquad (3)$$

or $$c \frac{\partial P}{\partial x} = I\omega \cos \omega \left(t - \frac{x}{c} + \frac{\psi}{\omega}\right) + R\omega \cos \omega \left(t + \frac{x}{c} + \frac{\phi}{\omega}\right) \qquad (3a)$$

if one adds equation (3a) to equation (2),

$$\frac{\partial P}{\partial t} + c\frac{\partial P}{\partial x} = 2\ R\omega\cos\omega(t + \frac{x}{c} + \frac{\emptyset}{\omega}) \tag{4}$$

and subtracting yields:

$$\frac{\partial P}{\partial t} - c\frac{\partial P}{\partial x} = 2\ I\omega\cos\omega\,(t - \frac{x}{c} + \frac{\psi}{\omega}) \quad . \tag{5}$$

Thus, from actual measurement of time and distance derivatives of the pressure wave, one can calculate the reflected and incident components of the wave.

Equations (4) and (5) contain on the left side of the equation the term c in addition to the time and space derivatives, and herein lies the source of difficulty of application of the theory. It is easy to calculate a curve of (R/I) (reflection coefficient) as a function of c (or $(R/I)^2$) for any one harmonic. Such a curve is of purely theoretical significance, since system behavior can be elucidated only if the velocity term can be determined experimentally. This is the source of greatest difficulty in the study of arterial reflections to date. Electrical transmission lines are usually long enough, and the ratio of frequency to wave velocity is great enough, that nodes and antinodes occurring at 1/4 wave length intervals can be clearly distinguished and studied. In the arterial system, circumstances are by no means so favorable. Therefore, it is impossible to determine (R/I) by conventional methods, so that c cannot be derived from (R/I) data alone.

Careful examination of the vector geometry of incident and retrograde wave summation does reveal, however, that certain relations exist which may be used to obtain experimental values for c and therefore for (R/I). For example, it is possible to calculate the ratio of amplitudes of resultant waves at two points in the aorta, if one has usable values for (R/I) and c. The equation for this is:

$$\left[\frac{A_2}{A_1}\right]^2 = \frac{1 + \left[\frac{R}{I}\right]^2 + 2\left[\frac{R}{I}\right]\cos(\alpha - 2\beta)}{1 + \left[\frac{R}{I}\right]^2 + 2\left[\frac{R}{I}\right]\cos(\alpha)} \quad . \qquad (6)$$

where $\frac{A_2}{A_1}$ = the ratio of amplitudes of a harmonic measured at points x_2 and x_1 along the vessel.

α = angle between the incident wave and reflected wave (α is a function of c)

$\beta = \frac{\omega \Delta x}{c}$

The terms necessary for the calculation of (α) are given as:

$$\alpha_1 = \tan^{-1} \frac{\frac{c}{\Delta x}\left[\frac{A_2}{A_1} - 1\right]}{\omega - \frac{c}{\Delta x}(\theta_2 - \theta_1)} \qquad (7)$$

$$\alpha_2 = \tan^{-1} \frac{\frac{c}{\Delta x}\left[\frac{A_2}{A_1} - 1\right]}{\omega + \frac{c}{\Delta x}(\theta_2 - \theta_1)} \qquad (8)$$

where, $\omega = 2\pi f$; θ_1 and θ_2 = harmonic phase angles measured at points x_1 and x_2, respectively; and Δx = distance between the two points of measurement, x_1 and x_2.

Since ω and Δx are known and $(\theta_2 - \theta_1)$ and (A_2/A_1) can be measured experimentally, it is possible to calculate actual values for α as a function of c. Again, however, c is an unknown quantity.

Fortunately, there are several equations in unknowns which can either be measured c derived from each other. If equation (4) and (5) are combined to obtain an expression for (R/I), this expression has the form:

$$\left[\frac{R}{I}\right]^2 = \frac{c^2 - Ac + B}{c^2 + Ac + B} \tag{9}$$

where

$$A = \frac{2\omega\Delta x \ (\theta_2 - \theta_1)}{(\theta_2 - \theta_1)^2 + \left[\frac{A_2}{A_1} - 1\right]^2} \tag{9a}$$

$$B = \frac{(\Delta x)^2 \omega^2}{(\theta_2 - \theta_1)^2 + \left[\frac{A_2}{A_1} - 1\right]^2} \tag{9b}$$

Thus, if one assumes a value for c, equations (7) and (8) can be used to calculate α, after which equation (9) can be used to calculate (R/I) and $(R/I)^2$. These can then be used in equation (6) to calculate a theoretical value for (A_2/A_1). The theoretical value can be compared with the experimental value, and if there is a difference (error function) a new value for c may be assumed and the process repeated. This iterative process eventually results in zero error, and the value of c at which this occurs is taken as the "true" phase velocity for that particular harmonic.

The actual error function in this system of equations is not as simple as one might have hoped, for the error crosses the zero axis at several different values of c. In general, the smaller the value of c becomes, the greater is the frequency of zero crossing. Fortunately, in all the curves we have analyzed, all zero crossings, except one, have occurred at c values which are so low as to be out of consideration.

Therefore, the one zero crossing yielding a reasonable value for *c* is taken as the "true" phase velocity.

It may be seen that this scheme of numerical analysis permits the conversion of data from pressure pulses recorded at two points in the blood vessel to actual values for *c*, (R/I), and $(R/I)^2$, all of which are important physical parameters which have not been determined heretofore. The authors are well aware that the assumptions made in this analysis reduce the sophistication of the approach. Future development of the method undoubtedly will include the perturbations associated with attenuation, non-linearities of system properties, etc.

C. Experimental Analysis

Our experiments were performed on dogs, anesthetized with nembutal (30 mg/Kg.) For the pressure recordings, special cannulae were constructed of stainless steel spinal needles (20 gauge) for direct insertion into the aorta. These were connected to the ports of Statham P-23G strain gage transducers. The entire system was fluid filled and the linear ranges for these transducers were ± 0-20 mm. Hg. and + 0-200 mm. Hg., respectively. Phase shifts of the transducers were within 2° at all significant frequencies. The overall frequency response of the system was determined by the Grass Polygraph recorder used for monitor recorder and signal amplification for the FM Analog tape recorder. The frequency response curve was flat, and down 5% at 40 cps. It should be noted that our analyses were usually limited to the fifth harmonic, which never exceeded 20 cps. in most experiments.

The data obtained from these studies consist of a series of simultaneously measured, arterial pressure pulses recorded from the aortic trunk of anesthetized mongrel dogs under various conditions described below. The special intra-arterial cannulae were designed for insertion directly though the aortic wall so that the cannula tip came to lie securely against the periphery of the internal diameter of the vessel with the orifice at right angles to the flowing stream. In this manner, the cannulae records true lateral pressure and does not disturb the normal flow pattern. Four such cannulae were placed along the length of the vessel and recordings registered. Originally, two simultaneous measurements were made; however, results from these studies suggest three or more simultaneous measurements would simplify the method of solution for the model equations. As many as seven simultaneous measurement can be made with the cannulae tips spaced about 5 cm. apart for information regarding the spatial distribution of the pressure pulse and its reflections.

The pressure pulses were measured under various conditions corresponding to (1) transmission medium and terminal impedance alteration by use of drugs (pharmacological vasoconstrictors and vasodilators), and (2) mechanical alteration of the terminal impedance of the aorta (i.e., mechanical occlusion of aorta at known distances from the catheter tips).

The pressure pulses from these experiments are recorded on FM Analog Magnetic tape (Mnemotron 700/1400) for subsequent selection, digitalization and analysis by the LINC computer.

2-9

II. Past and Present Project Utilization of the LINC

Within the above framework, the LINC has been and is currently being used for off-line and on-line processing of experimental data. Prior to the acquisition of LINC, analog-to-digital conversion was done by hand and processing was accomplished on an IBM 1620 computer for which FORTRAN programs had been written. After acquiring LINC the decision was made to utilize the LINC as extensively as possible without jeopardy to the on-going research program. A conventional IBM 026 Keypunch was then modified to accept BCD information generated by the LINC and to punch this information onto standard punch cards for use with existing IBM 1620 programs. A description of the design and use of this modification is presented in Appendix I. This development permitted simultaneous use of the IBM 1620 and the programs written for it, and the LINC as an analog-to-digital converter until routines could be written for complete processing.

Then began the huge task of writing the necessary arithmetic routines, such as floating point, add, subtract, multiply and divide, square root, sine, cosine, arctangent, log, etc., required for project data analysis. Thanks to the efforts of several LINC users, the time required to finish this task was limited to less than a year. It has been only within the last two months that the LINC has performed satisfactorily from the standpoint of providing the required arithmetic output.

In terms of off-line processing the LINC is currently performing Fourier Analysis of the arterial pressure pulse data producing an amplitude and phase for each desired harmonic for each of four channels of information. This information is transferred to IBM punch cards and fed to an IBM 1620 for further processing in terms of incident and reflected vectors, characteristic and terminal impedance harmonic phase velocity, and related factors. Programs for accomplishing this were written prior to the commencement of the LINC evaluation program and have not yet been completely programmed for the LINC.

Current on-line processing involves computation of harmonic amplitudes and phase angles for each of four simultaneously measured pressure functions followed by scope display of the data. This need developed from bizarre results following LINC processing of data after an experiment. Specifically, the results showed a non-linear (almost random) distribution of mean arterial pressure and harmonic phase angle with distance along the aorta. While this could be real, it is difficult to justify on the basis of arterial dynamics. Such unlikely distributions could be brought on by clot formation within the cannulae, cannulae showing in situ, or other likely circumstances difficult to detect with conventional monitor equipment.

When such data are subjected to the numerical analysis outlined in this proposal, bizarre results are obtained simply because the model has not taken these factors into account. With the current LINC on-line processing facility, we are able to monitor spatial distribution of mean pressures, and harmonic amplitudes and phase angles for each pressure channel. In this manner we are able to detect prior to actual processing whether or not the data collected during the experimental phase meet the criteria for the numerical analytical scheme. If failure is indicated, corrective measures are taken and subsequent recorded data are earmarked as "good". Alternately, one could run through a complete experiment without knowing until final processing is done that the data do not meet the criteria for successful numerical analysis.

A few of the programs written for this project are presented in Appendix II. Included with each program are a routine description, a LAP

III listing of the program instructions with comments (in some cases), an example of the output, and a flow chart where possible. All such programs listed have been checked to the best of our ability and function to the extent shown on the output example. More programs might be listed but either they have not been thoroughly checked or completed to satisfaction at this time.

III. Future Research

It has been only since our arithmetic routines have been working properly that we have come to realize the full potential of the LINC in regard to our research program. What we have viewed as a sophisticated A-D converter prior to this time has now become a useful laboratory tool.

The research program on studies related to the hydrodynamics and the transmission characteristics of the arterial system is scheduled to continue for at least five more years. It is obvious from this evaluation program that the LINC or a machine with "LINC-type" characteristics is extremely valuable if not necessary to achieve the desired end product. With the present investment in programming and trained personnel duplication of this effort would be impractical if not fool-hardy.

With these thoughts in mind it is planned to carry this particular research program to its logical conclusion with the aid of the LINC. This will involve a more elaborate examination of the behavior of passive hydraulic transmission elements, such as rubber or teflon tubes as well as active elements such as aortas and other large arteries.

The passive element study will involve excising a segment of aorta and insertion of a teflon or rubber tube in situ. The pressure parts will be separated by a known distance in a tube whose physical properties are either known or easily measured. This maneuver would reduce by at least five the number of parameters we must now take into account. Once the relationship between the passive element and the cardiovascular system is worked out, this technique will be applied to an active aorta of varying length on different animals and on the same animal with varying degrees of arterial stiffness from whatever cause. The major objective will be to describe in detail the spatial and temporal variations of pressure and flow in the arterial system of mammals with special emphasis on the determination of complex vascular impedance patterns and in vivo arterial elasticity.

Our future plans include also the acquisition and utilization of a Datamec tape recorder. This machine has been ordered by another laboratory within the medical school for a project involving pattern recognition of the phonocardiogram and its correlation with other events of the cardiac cycle. The LINC is scheduled for use as a data reduction device prior to storage of information on digital magnetic tape for processing by an IBM 7094 computer.

IV. Training Program

During the past 16 months the LINC installation has been associated with several graduate and medical teaching functions which are likely to recur each year. These are summarized below:

Course Title	Student Classification	Contact Hours
Data Processing and Computer Techniques	Graduate Students	45
Medical Physiology	Medical & Graduate Students	48
Research Methods	Graduate Students	7
Medical Electronics	Graduate Students	45

In addition three graduate students have learned to program the LINC and will most probably utilize the machine in processing their research data. Last summer, three fellows worked on research projects in this laboratory in which the LINC was utilized.

The LINC has not received widespread acceptance among the graduate students for three very good reasons: (1) most of the students are not to the point in their program where processing experimental data is an important consideration, (2) those who are aware of this need have been discouraged by the amount of programming required to supply simple basic routines which are easily available on any machine using FORTRAN, and (3) there is no reference programming manual from which they could learn LINC symbolic programming techniques.

We have tried to train three programmers (other than graduate students) on the LINC thus far. The first had a nervous breakdown, the second simply didn't want to learn and the third quit to take a better job. In all three cases, it required no less than two months before these programmers could write a new program with confidence (graduate students seem to learn much faster).

Most of the graduate students currently using the machine feel the display feature on the scope, the magnetic tape units and the debugging facilities (I-STOP, XOE-STOP, EXAM, I-by-I, and C-by-C) are among the best features on the LINC. The teletype has added a great deal to the respect of many for the machine.

V. Computer Performance

A. Maintenance

Of the nearly 2000 hours LINC has been operating since fabrication, less than 4 hours has been spent in maintenance and repair. This effort involved replacing a transistor in one of the drivers, cleaning the air filter, resoldering two broken wires in the fantail cables and adjustment of the space bar return spring tension on the keyboard. A considerable amount of time was spent trouble shooting problems related to the tape units which culminated in a visit made by Mr. H. Lewis. During this visit tape tension was adjusted, shoes were aligned, and a bent drive shaft was replaced. Since that time no malfunctions have occurred that could be directly attributed to LINC hardware.

B. Suggested Changes

The analog-to-digital converter has been adequate for most applications encountered. However, it is felt that 6-11 bit A-D conversion, switch selectable would enhance the overall usefulness of the machine. For this particular project, at least 9 bits of conversion seem to be required.

It is our recollection that analog output was to be provided for on LINC III. In several instances we have wanted output on an x - y plotter, or in some other analog form, especially in situations where the analog computer was involved. A digital-to-analog feature for our laboratory would be very desirable.

The analog-to-digital preamplifiers are rated linear between ±1 volt, yet our system appears to be linear over the range ± 3/4 volt. In addition, there appears to be unexplainable DC level differences and phase shifts introduced through the preamplifiers. In our system, channel 10 is particularly bad.

The plug-in modular design of the data terminal box has facilitated most of our interfacing problems. At the same time they have caused considerable congestion at the console due to the many and varied cables that must make connection on the front face plate. While this may be partially obviated by physical separation the two units current space limitations dictate proximity of all four units. It would be advantageous if data terminal box connections with external equipment could be made in the rear of the unit.

A design feature to allow for optimal lengths of interconnecting cables between the main frame and component parts would be very helpful. In our particular setup, 30 foot cables are too short if the machine is to be remote from the experiment and much too long if the machine is to remain non-portable.

Provisions for more than two simultaneous display scopes in addition to the one at the console would be helpful in many instances. Also, an interconnecting cable greater than 30 feet in length between LINC and remote the oscilloscope appears to be particularly desirable in our laborabor setup.

C. Summary Remarks

With regard to the utilization and capabilities of any laboratory computer, it can probably be stated with a certainty that the ultimate productivity of such a device, or its contribution to research productivity, is at least as much determined by factors other than machine capability as it is by the specific machine design. That is to say, the degree to which a laboratory instrument computer can enhance the productivity of a research program is influenced to a greater degree by the available software and supporting information than it is by any specific hardware considerations.

There is little doubt at this time concerning the hardware aspects of the LINC computer. The machine has proved to be a reliable, fast, and efficient computing device. In practically no case has the actual hardware limitation of the LINC computer imposed a serious restriction on its use. To be sure, there have been specific machine malfunctions but these have not been frequent and have not been more than one should expect with a system of this degree of complexity. The familiarity with the nature of machine function which was obtained in the process of fabricating the instrument has proved to be at a sufficient level to allow machine maintenance and interface design to be accomplished by those who would actually use the instrument.

In the area of software and supporting information, however, a rather severe limitation exists on the degree to which the LINC can be utilized in research activities. This limitation is exhibited most clearly in the lack of available basic computer programs and in the insufficient communication along those who are using the LINC computer and attempting to increase its effectiveness in their research. It is perhaps unfortunate that the LINC was placed into the hands of a number of individuals before a basic set of arithmetic programs were available and upon which a more uniform programming system could be built. In addition, the lack of sufficient communication between individual users has resulted in a great deal of needless duplication of programming effort and also to very non-uniform programming procedures. It would seem advisable to establish at the earliest possible date a centralized programming staff and office which would be charged with the function of alleviating these difficulties. Until a regular and useful programming system is established, it is probable that many investigators can best apply their programming and research efforts to machines for which greater supporting information is available. This may be true even though the most significant features of on-line operation and rapid analysis which the LINC possessess may not be available.

VI. Log Book

Our log book consists of 295 pages of information ranging from registration cards for equipment ordered to all memos sent us from CDO. This makes it impractical to send a copy for distribution. Instead, copies of related correspondence is attached.

VII. Bibliography

A. Publications:

1. Malindzak, George S., Jr. and Ralph W. Stacy: "Reflections of Pressure Pulses in the Aorta", Proceedings of the 2nd Annual Symposium on Biomathematics and Computer Science in the Life Sciences, (In Press).

2. Malindzak, George S., Jr. and Ralph W. Stacy: "Dynamics of Pressure Pulse Transmission in the Aorta", to be published by the N. Y. Academy of Science.

3. Malindzak, George S., Jr.: "Transmission Line Characteristics of the Mammalian Arterial System", to be published by the 6th International Conference on Medical Electronics and Biological Engineering.

B. Talks:

1. "Use of a Computer in Physiological Research", 8th Annual IEEE Symposium, Greensboro, North Carolina.

2. "Reflections of Pressure Pulses in the Aorta", 2nd Annual Symposium on Biomathematics and Computer Science in the Life Sciences, Houston, Texas.

3. "The LINC Computer", Professional Group on Biomedical Electronics of the IEEE, Winston-Salem, North Carolina.

4. "Engineering in Medicine and Biology", Western Electric Field Engineering Force, Winston-Salem, North Carolina.

5. "Signal Enhancement of the Fetal Electrocardiogram by Statistical Methods", Department of Electrical Engineering, Rice University, Houston, Texas.

6. "Computers in Medical Science", Western Electric NIKE-X Engineers, Winston-Salem, North Carolina.

7. "Dynamics of Pressure Pulse Transmission in the Aorta", to be given at the New York Academy of Sciences Conference on Advances in Biomedical Computer Applications, June, 1965.

8. "Transmission Line Characteristics of the Mammalian Arterial System", to be given at the 6th International Conference on Medical Electronics and Biological Engineering, Tokyo, Japan, August, 1965.

APPENDIX I

LINC-IBM 026 Keypunch Interface

The purpose of this interface is to provide the LINC computer with a punch card output. The IBM 026 keypunch was chosen for this application because of its inexpensive rental and its relative ease of modification. A punch card output was deemed desirable because data can be permanently stored and transferred to a larger machine with greater capabilities than the LINC. The operation and function of the keypunch unit is not altered in any way by the addition of these modifications. With this system it is possible to punch the cards with any of the characters available in a standard IBM format. Under LINC program control, it is possible to format information in any form onto IBM punch cards.

In order to proved the capability of punching any IBM character format, it is necessary to be able to punch as many as three holes in a single column. Numbers and plus and minus signs require a single punch, letters require two punches including a 10, 11, or 12 punch together with a digit punch, and punctuation requires three punches including an 8 punch and an 11 or 12 punch. The GA lines available in the data terminal box have been designated to provide the necessary punching information as follows:

GA 0	BCD Decoded	Digit punches 0-9
GA 1		
GA 2		
GA 3		
GA 4	-	11 punch
GA 5	-	12 punch
GA 6	-	Space
GA 7	-	Extra 8 punch
GA 8	-	Extra 0 punch
GA 9	-	Release

Figure 1 shows a component arrangement for our interfacing for the LINC. The DEC cards and all the control functions are combined in the LINC data terminal box. To avoid noise problems, relay drivers were built into a separate unit and placed physically away from the LINC and supplied with their own voltage supply which was located in the keypunch. Figure 2 is the circuit diagram for the entire interface. It is noted that the GA lines are all buffered with inverters for purposes of isolation. The block labeled "one-shot" consists of two monostable multivibrators. These were of our own design; however, any unit is capable of a delay from 5 to 30 milliseconds would be suitable.

A timing diagram illustrating two punching cycles is illustrated in Figure 5. The GA lines carry zero levels corresponding to the presence of bits in the accumulator register. With an OPR instruction, the OPR line is enabled which in turn triggers the punch delay if a card is in position in the keypunch. This is verified by a -3 level being present on the -3A line. The end of the punch pulse triggers the second delay whose output enables the XL line which allows the LINC to proceed. The -3A line is broken whenever a card is between columns or during a release cycle. The keypunch will require approximately 25 milliseconds to complete its column transfer after the XL restart signal has been supplied to the LINC before it will again provide the -3A level. The punch delay is approximately 10 milliseconds and the delay before restart is approximately 27 milliseconds, thus a maximum rate of about 60 milliseconds per character is provided.

The relay drivers shown in Figure 4 obtain their +15 volts from the supply located in the keypunch. The voltage levels on the base of the transistors swings between a +2 and a -1 volt to provide positive on/off action of the transistor. The relays and the relay contacts are shuttered with diodes to provide transient protection.

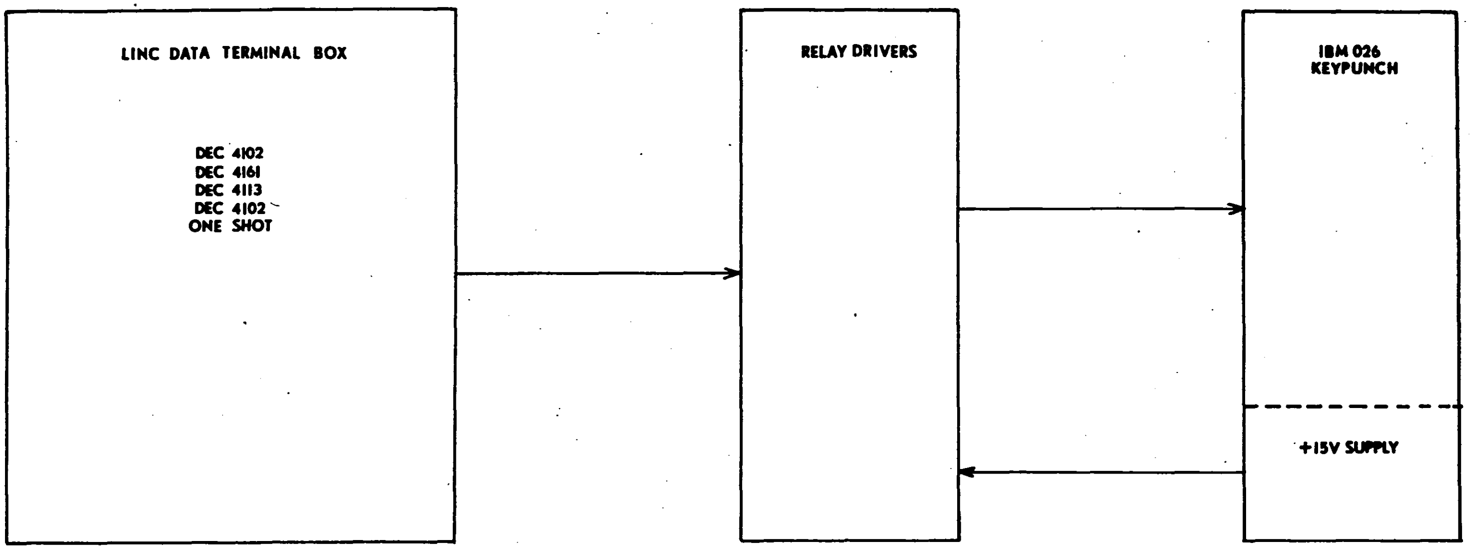
Figure 6 shows the rear of the IBM 026 keypunch and the location of the items to be modified. In the keypunch, relay 1 and 2 operate during the release cycle. For this reason we have paralleled these relays with relays 1A and 2A (Potter and Brumfield type KA140, 110VAC, DPDT) which open the -3A line during the release cycle. Closeups of the wiring on relay 45 and relays 1 and 2 are shown in Figures 7 and 9. One of the largest problems in using the IBM 026 keypunch is in the large amount of electrical noise generated by this machine. One of the major sources of noise was found to be the program cam contact. Placing a 0.1 mfd capacitor across these contacts eliminated this noise source. In addition, it was found to be necessary in the data terminal box to run a direct ground lead from each socket to a chassis tiepoint. These connections are evident in the illustration of the data terminal box shown in Figure 10. In addition, each of the voltage supplies was by-passed with a 0.025 mfd capacitor to the chassis ground. A 0.5 mfd capacitor on the -3A line was also necessary for transient suppression. To prevent false triggering of other enabling lines within the LINC, the following enabling levels were tied off to ground: VNEL, SNEL, UNEL, BEGT, MONT, CLEL, MINT, and TNEL.

The punch cam modification is shown in Figure 3. The metal jumpers connecting 1, 2, and 3R and the lead from 2L are removed, 1 and 3R are then jumpered and the -3 line connected to 2L and relay 1A line connected to 2R. These are the contacts which open the -3A line when the keypunch is between columns on the card.

In operation, the keypunch system is simple to sue and reliable in operation. It is only necessary to place the proper character code in the accumulator, execute an OPR 10 instruction, and then execute a small delay before the execution of the next OPR instruction. A test program which can be used to check the operation of the keypunch interface is included in this appendix. This test program also illustrates the punch code for the various characters which can be produced. This character code starts at manuscript line number 21 with the number 0 and proceeds through manuscript line number 124 and the code for release. In this program, the necessary small delay after an OPR instruction is produced by the XSK12 instruction and its associated loop.

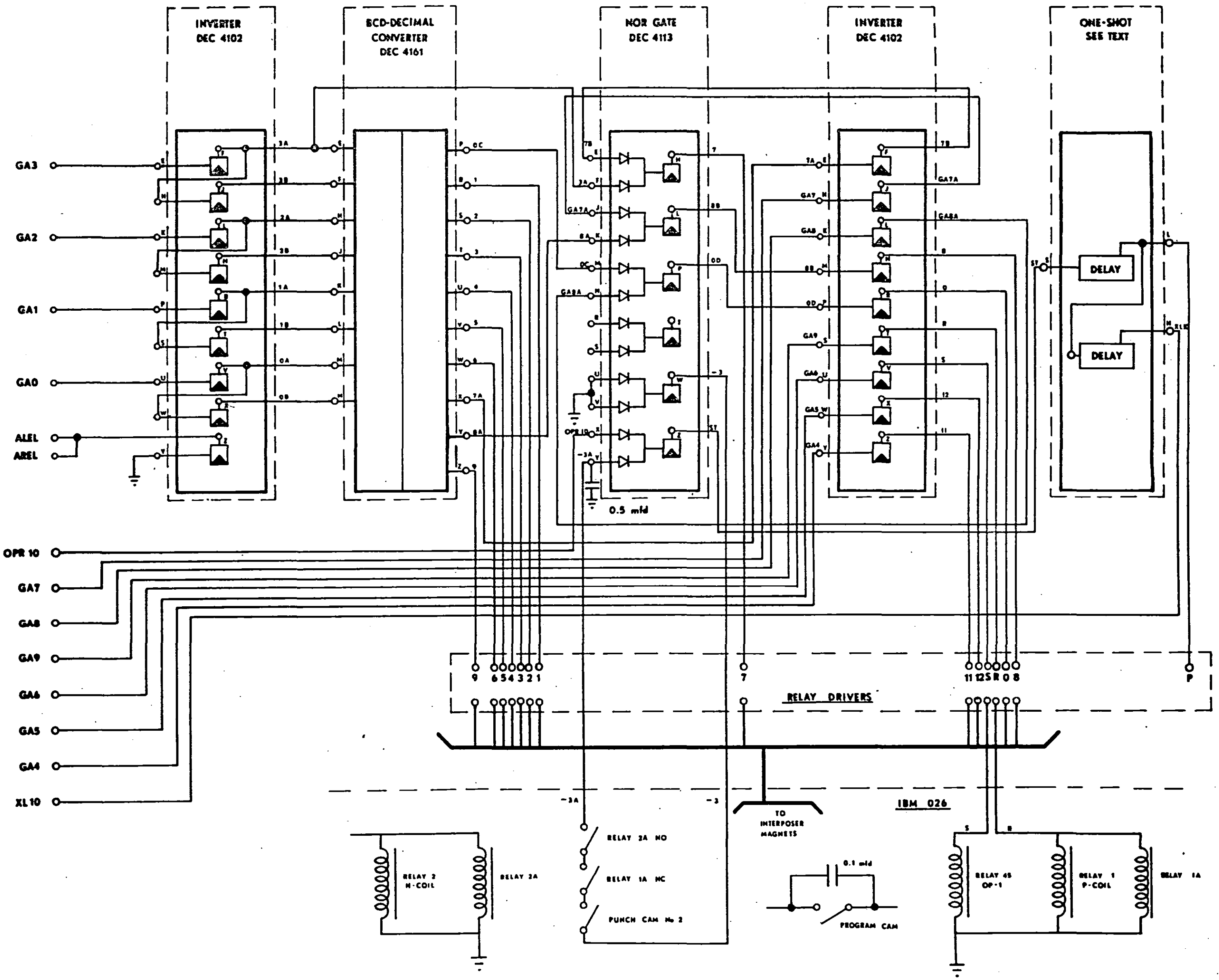
Wishing not to burden this report with further detail concerning this LINC to keypunch interface, the authors would prompt anyone interested in duplicating such a system to contact the authors for any further specific information they might desire.

A test program is attached.



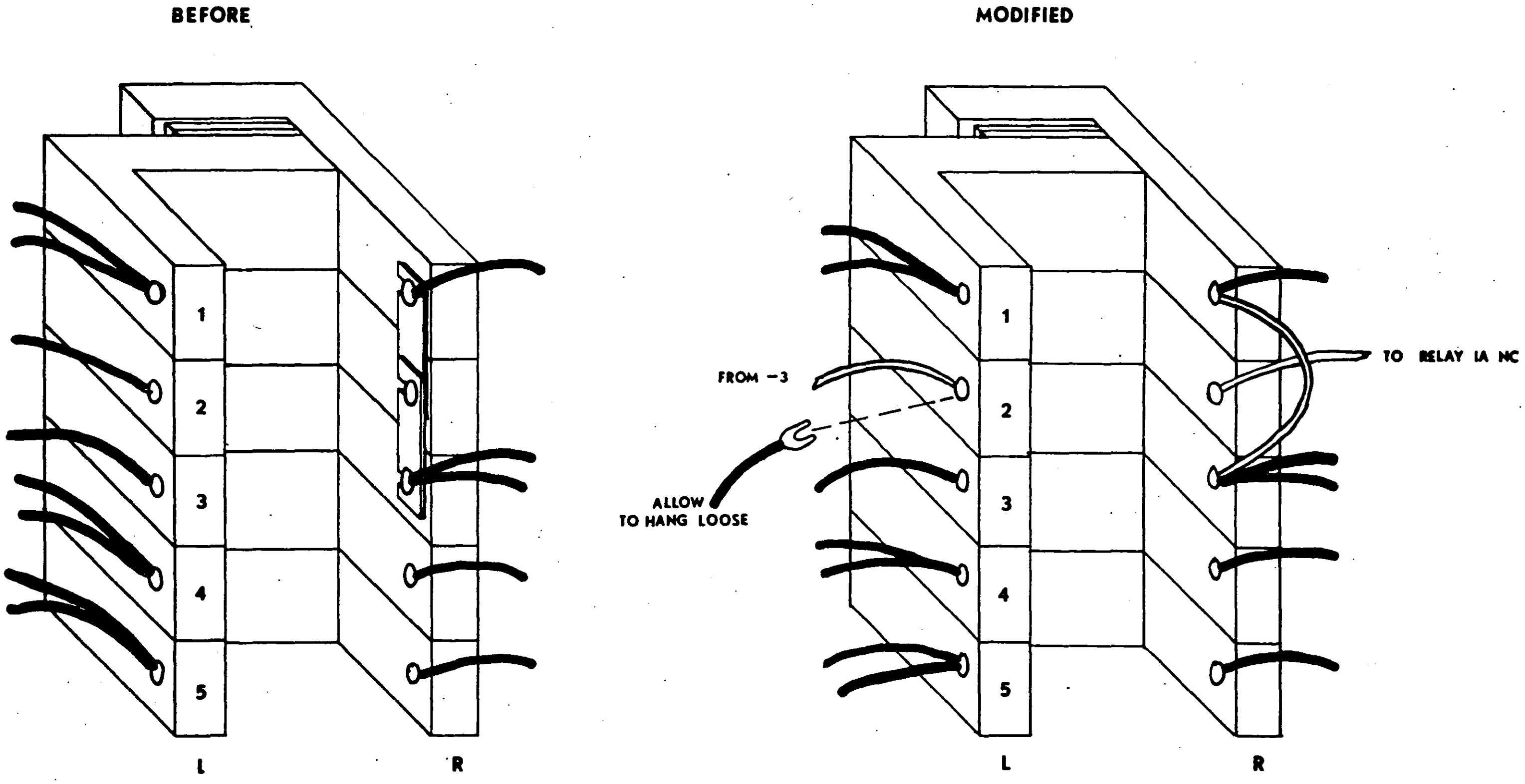
SYSTEM BLOCKS

FIGURE 1

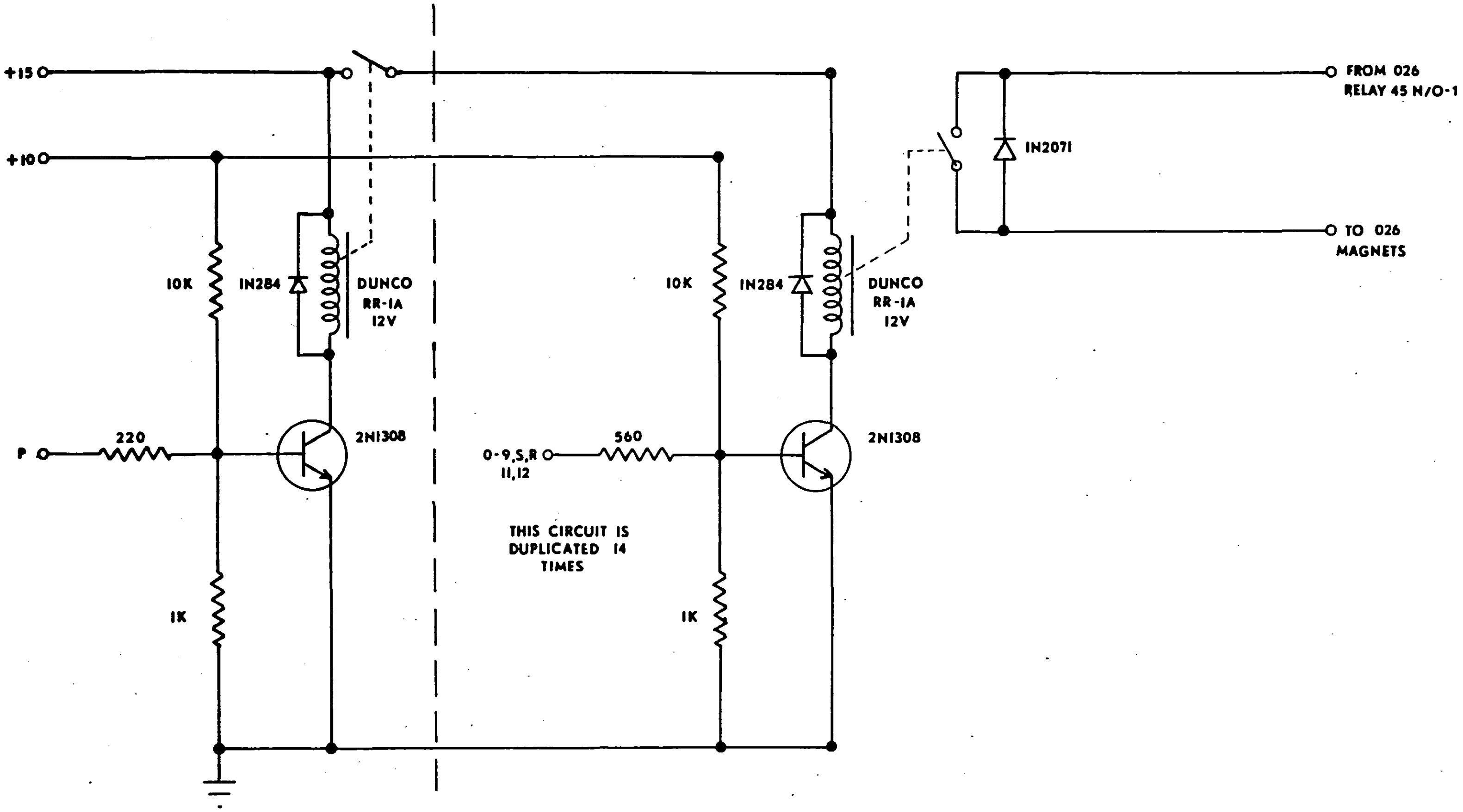


LINC-IBM 026 INTERFACE

FIGURE 2

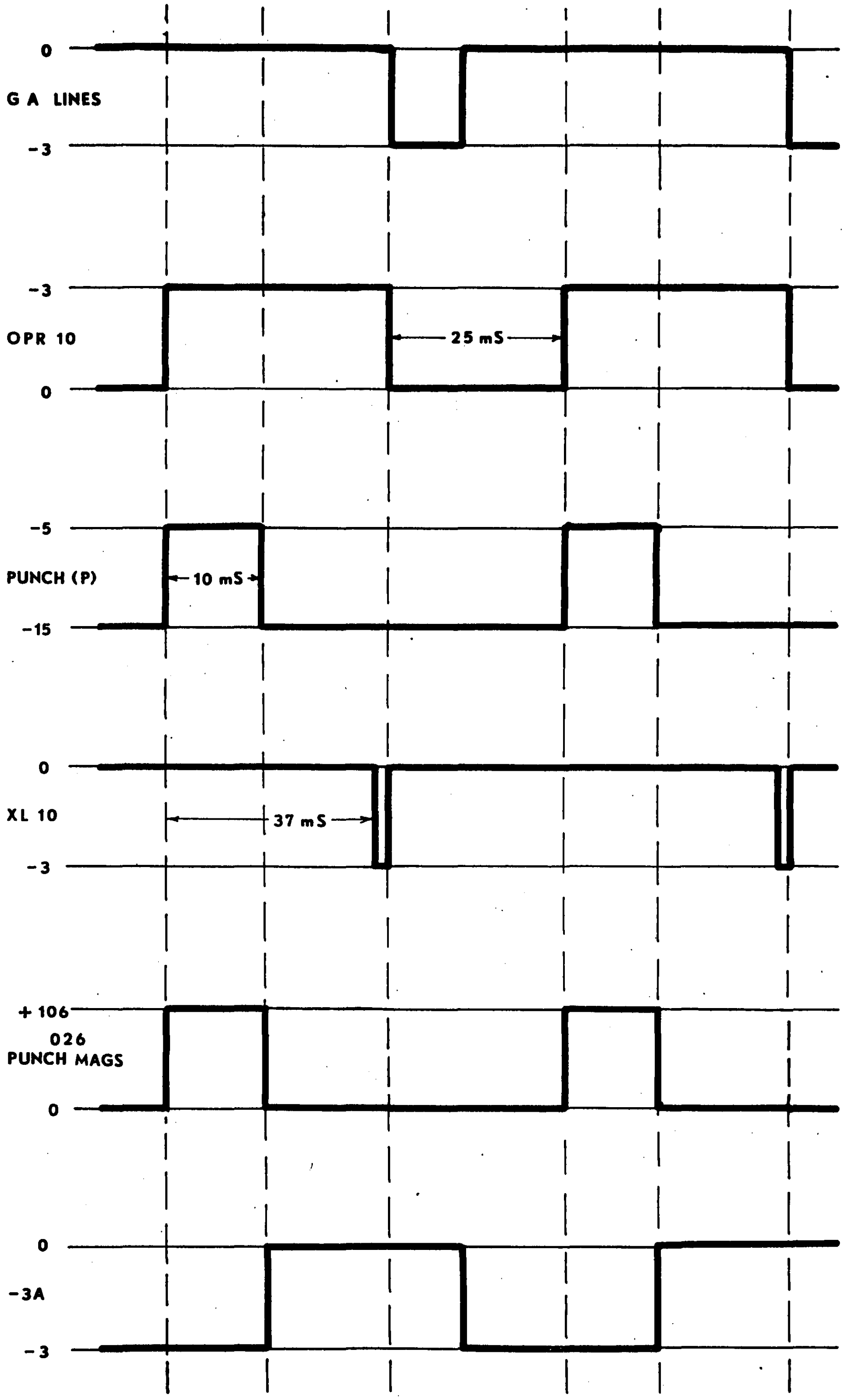


PUNCH CAM MODIFICATION
FIGURE 3



RELAY DRIVERS

FIGURF 4



TIMING

FIGURE 5

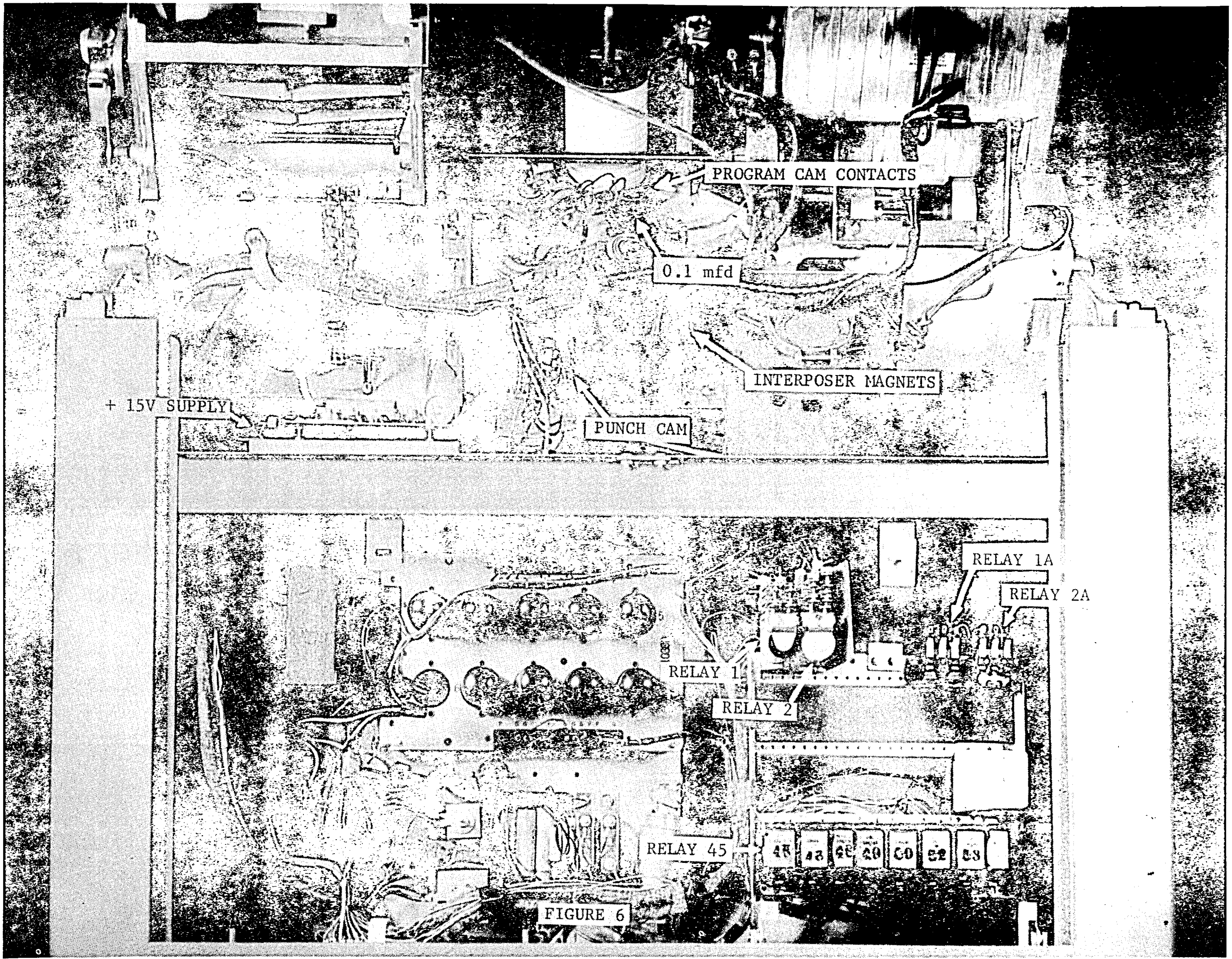

PROGRAM CAM CONTACTS
0.1 mfd
INTERPOSER MAGNETS
+ 15V SUPPLY
PUNCH CAM
RELAY 1A
RELAY 2A
RELAY 1
RELAY 2
RELAY 45

FIGURE 6

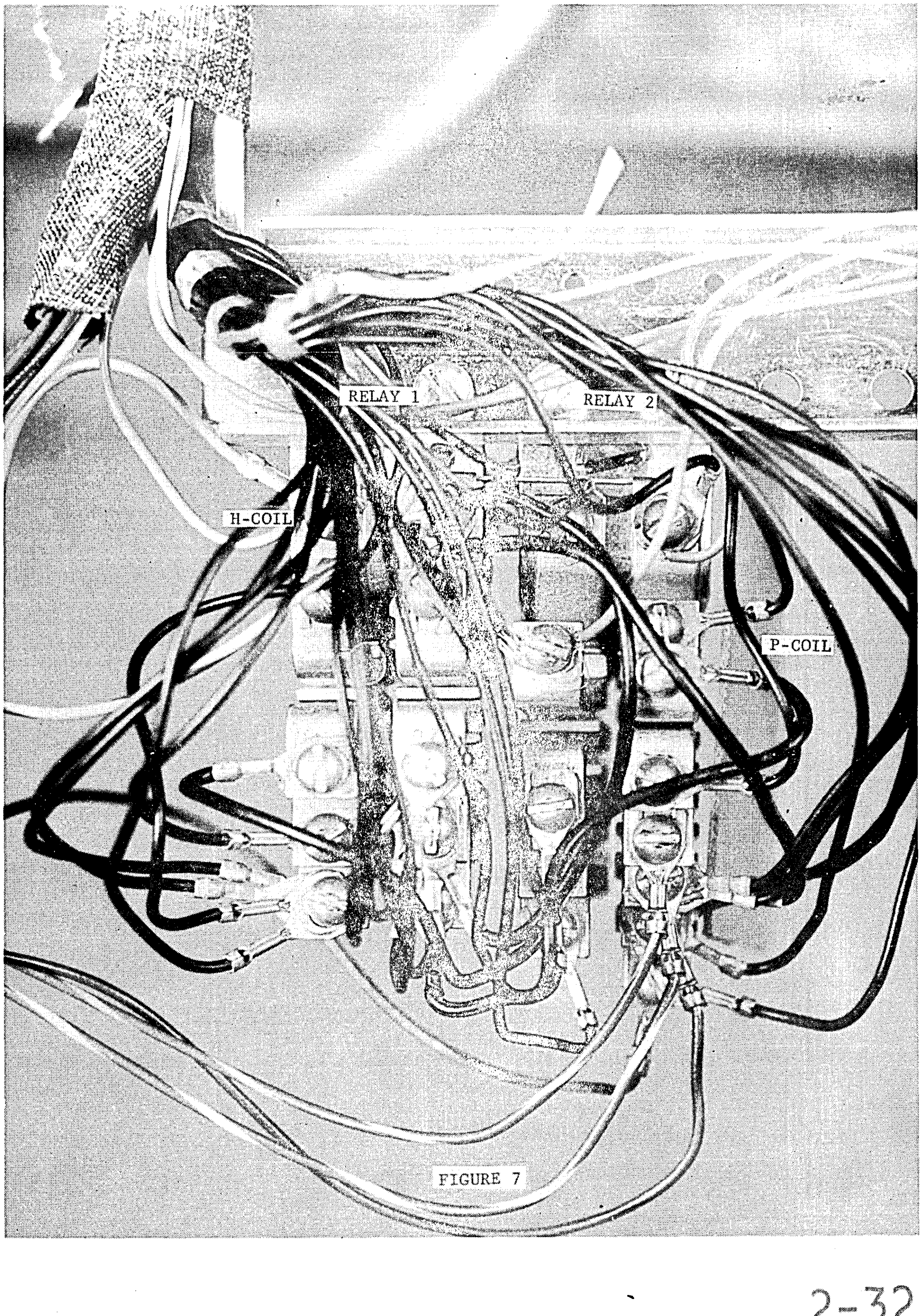


FIGURE 7

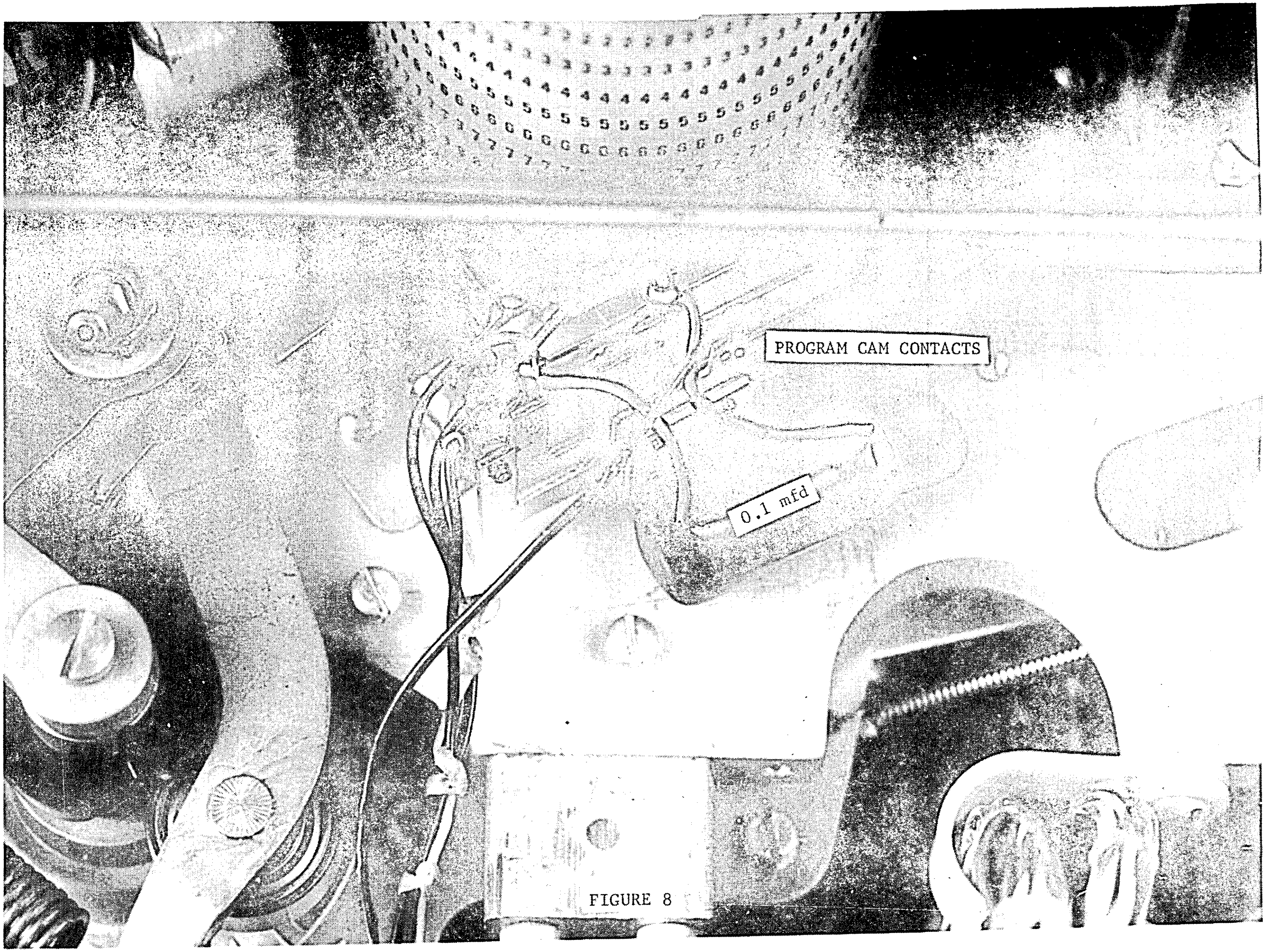


FIGURE 8

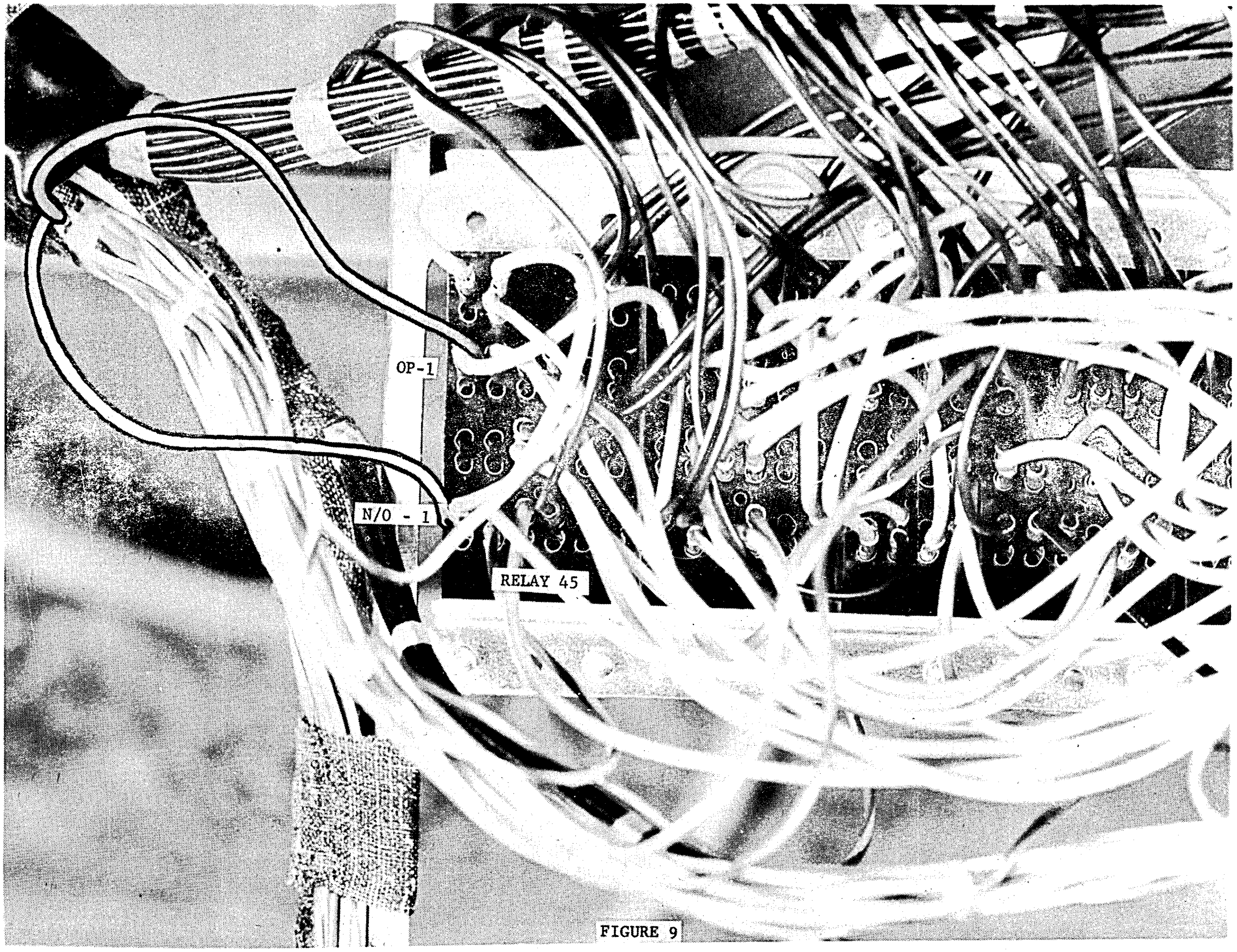
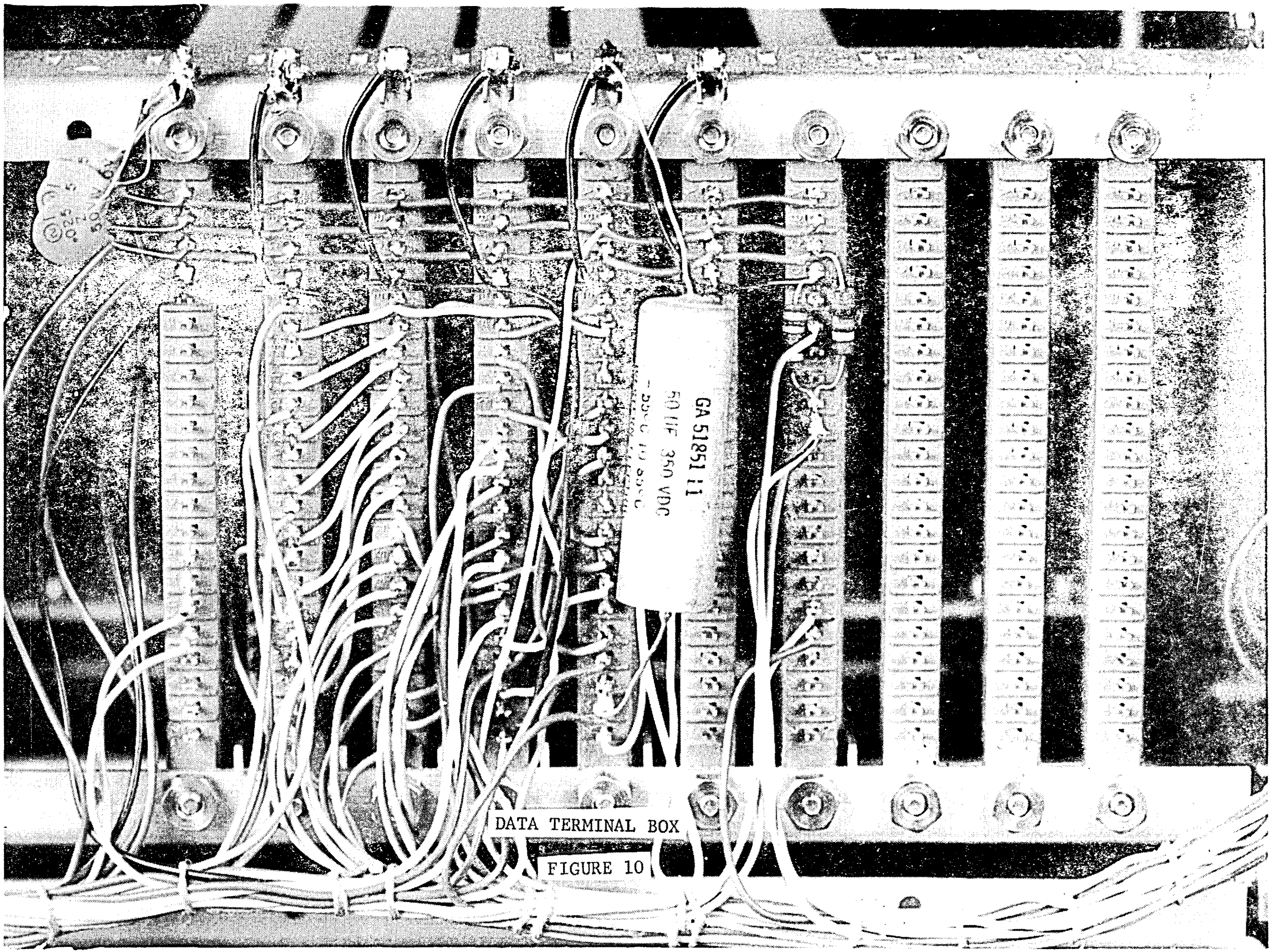


FIGURE 9



DATA TERMINAL BOX

FIGURE 10

APPENDIX II

Selected LINC Programs

1. Binary-to-decimal Conversion-V (with Fraction)
2. Variable Sample and Display-IV
3. Plot Routine-1
4. Plot Routine-3
5. Plot Routine-5
6. Fourier Analysis-11, Generation of SIN $N(K(2\pi/m))$ and COS $N(K(2\pi/m))$ Table
7. Master III Routine
    a. Executive Control, Calibration, Sample, Display and Keypunch
    b. Fourier Analysis -17 - Compute A_n & B_n
    c. Fourier Analysis - 24 - Compute C_n & θ_n
    d. Fourier Analysis - 25 - Display 1, Display 2, & Plot

MASTER III Routine

Program Description:

(1) Master III Control Routine, Start 20

(2) Fourier A_n and B_n Routine, Start 400

(3) Fourier C_n and Θ_n Routine, Start 400

(4) Display 1, 2, and plot Routine, Start 400

Procedure:

1. All SNSs off
2. RCG, 2/420, (location of Master III - binary)
   a) Start analog tape
   b) Read in calibration data
3. Start 20
4. 1st, pause, strike "P" on keyboard to obtain calibration factor
5. 2nd, pause; SNS 0 on
6. Advance analog tape to pressure pulse, press P, then start 20 to samples and display. Proceed to (7) only when they appear satisfactory.
7. SNS 0 off to Scaling Routine
8. 3rd, pause; SNS 2 on; SNS 1 on to skip keypunch routine, press P to enter FR. Analysis Routine and compute A_n, B_n, C_n, Θ_n, Display 1. Press P.
9. After display 1, use SNS 3 on to enter display 2. TURN ON TTY for plot routine.
10. After display 2, turn SNS 3 off to enter display 1 and repeat procedure (9).
11. After all the display, SNS 4 on to enter TTY, plot routine.
12. Master III routine is re-read after plotting.
13. All the SNSs off
14. Repeat (1). To skip calibration factor routine, SNS 0 on.

Master III Routine Consists of the Following Principle Subroutines:

(I) Calibration Factor Routine

(II) Sample and Display Routine

(III) Scaling Samples Routine

(IV) IBM-026 Keypunch Decimal Output Routine

(V) Fourier Analysis - 17 - Compute

$$\left.\begin{aligned} A_n &= \frac{2}{m} \sum_{K=1}^{m} Y_k \operatorname{Cos} (N(K(2\pi/M))) \\ B_n &= \frac{2}{m} \sum_{K=1}^{m} Y_k \operatorname{Sin} (N(K(2\pi/M))) \end{aligned}\right\} \quad \text{Fourier Coefficients.}$$

From 4 sets of Y_k Pressure Pulse Data Input

From Channel 10, 11, 12, & 13

(VI) Fourier Analysis - 24 - Compute

$$C_n = \sqrt{A_n^2 + B_n^2} \quad , \quad \text{Fourier Moduli}$$

$$\theta_n^o = \tan^{-1} \frac{A_n}{B_n} \quad \text{Phase Angle}$$

From 4 sets of A_n, B_n, Result of Routine (V)

(VII) Fourier Analysis - 25 - Display C_n & θ_n^o; For each channel, or display C_n & θ_n^o ; For each harmonic (θ_n^o in degree, C_n, θ_n in decimal), and Plot the four sets of Y_k pressure pulse data simultaneously. Against Time Axis, each time interval being 48 milliseconds.

(VIII) Recover original control routine

SNS 0, 1, 2, 3, 4, are used during the computation and output (from keypunch, oscilloscope, and teletype) in addition to the keyboard control.

With SNS 0 off, enter Calibration Factor Routine. Calibration factor is obtained according to the following procedures:

(1) Obtain data from analog channels, 10, 11, 12, 13, which are connected to a Model 700/1400 Mnemotron FM Analog Tape Recorder on which has been previously recorded pressure pulse calibration data.

(2) Average 200_8 input samples for each channel and obtain a single point on the calibration curve; do the same for the next calibration level. Compute the slope between points to yield calibration factors for each channel. With SNS 0 on; enter Sample and Display Routine. The procedure is following

  (a) By means of a Schmit-Trigger circuit connected to XL 1 sample 200 points along a pressure pulse wave from each channel. The Schmit-Trigger is used to fire a delay which in turn provides a start pulse on XL 1; a stop pulse is provided at the same point in the following cycle. The data are then displayed on the Oscilloscope, which now include 1 complete cardiac cycle.

  (b) A potentiometer is provided to adjust the fixed delay so as to obtain a complete cycle for each set of data, at the desired start and stop points.

With SNS 0 off, enter Scaling Sample Routine which is simply a routine to multiply each sample by its corresponding calibration factor. Use SNS 1 up to skip Keypunch Routine.

With "P" on the keyboard, enter IBM 026 Keypunch Decimal Output Routine. This routine includes last octal digit round off and octal to decimal conversion. The punch cards contain

(1) Total number of samples from each channel (on the first card)

(2) Ten data words with a decimal point before the last digit of each number and a space between consecutive numbers on each card. After each set of data are punched out, the program pauses until "P" is struck on the keyboard. Four sets of data are punched out in this fashion under the keyboard control.

After the last set of data cards have been punched out, the program pauses. With SNS 2 On, and striking "P" on the keyboard, Fourier Analysis -17 is read into QN1, QN2, and QN3.

Fourier Analysis -17 computes Fourier Coefficients for each of the four sets of scaled samples. Y_k's are stored in the upper half of memory. The computations are done as follows:

(1) Fourier Cosine Coefficient

$$A_o = \frac{1}{M} \sum_{K=1}^{M} Y_k$$

$$A_n = \frac{2}{M} \sum_{K=1}^{M} Y_k \ \text{Cos} \left[N(K(2\pi/M)) \right]$$

(2) Fourier Sine Coefficient

$$B_o = 0$$

$$B_n = \frac{2}{M} \sum_{K=1}^{M} Y_k \ \text{Sin} \left[N(K(2\pi/m)) \right]$$

Where N is the number of harmonics (for experimental purpose, N is set to equal to 5), M is the total number of samples from each channel, K is the increment such that $1 \leq K \leq M$, $2\pi = 6.28$ radian.

After the above computation, Fourier Analysis -24 is read into QN1, QN2, and QN3.

Fourier Analysis -24 computes Fourier Moduli and Phase Angles defined as following:

(1) Fourier Moduli

$$C_n = \sqrt{A_n^2 + B_n^2}$$

(2) Phase Angle

$$\theta_n = \tan^{-1} \frac{A_n}{B_n}$$

Four sets of C_n and θ_n are computed from four sets of A_n and B_n.

After the above computation, Fourier Analysis -25 is read into QN1, QN2, QN3.

Fourier Analysis -25 consists of three main subroutines:

(1) Display 1

(2) Display 2

(3) Plotting 4 simultaneous pressure pulse waves against time axis (in ms).

The subroutines are described as following:

(1) Display 1

Display 1 consists of three main subroutines

A. Octal to Decimal Conversion
B. D. P. Multiply
C. Display Loop

Display format is as follows (in decimal)

CH	C	θ^o
10	XX.XXX	XXX.XXX
11	X.XXX	XXX.XXX
12	X.XXX	XXX.XXX
13	X.XXX	XXX.XXX

N = N'

Where N' = 0, 1, 2, 3, 4, or 5, CH represents channel, C represents Fourier Moduli, θ° represents phase angles, N represents the number of harmonic.

Keyboard control is following:

A. EOL to advance to next harmonic data, N' + 1 → N'.

B. R , Repeat display

(2) Display 2

Display 2 shares the subroutines A, B, C, with Display 1.

Display format is as follows (in decimal)

N	C	θ°
0	XXX.XXX	XXX.XXX
1	XXX.XXX	XXX.XXX
2	XXX.XXX	XXX.XXX
3	XXX.XXX	XXX.XXX
4	XXX.XXX	XXX.XXX
5	XXX.XXX	XXX.XXX

CH = X

Where X = 10, 11, 12, 13, N represents the number of harmonic, C represents Fourier Moduli, θ° represents phase angles, CH represents channel from which data is obtained.

Keyboard control is the same as in Display 1.

(3) Plot 4 simultaneous pressure pulse waves against time axis (4_8 ms intervals).

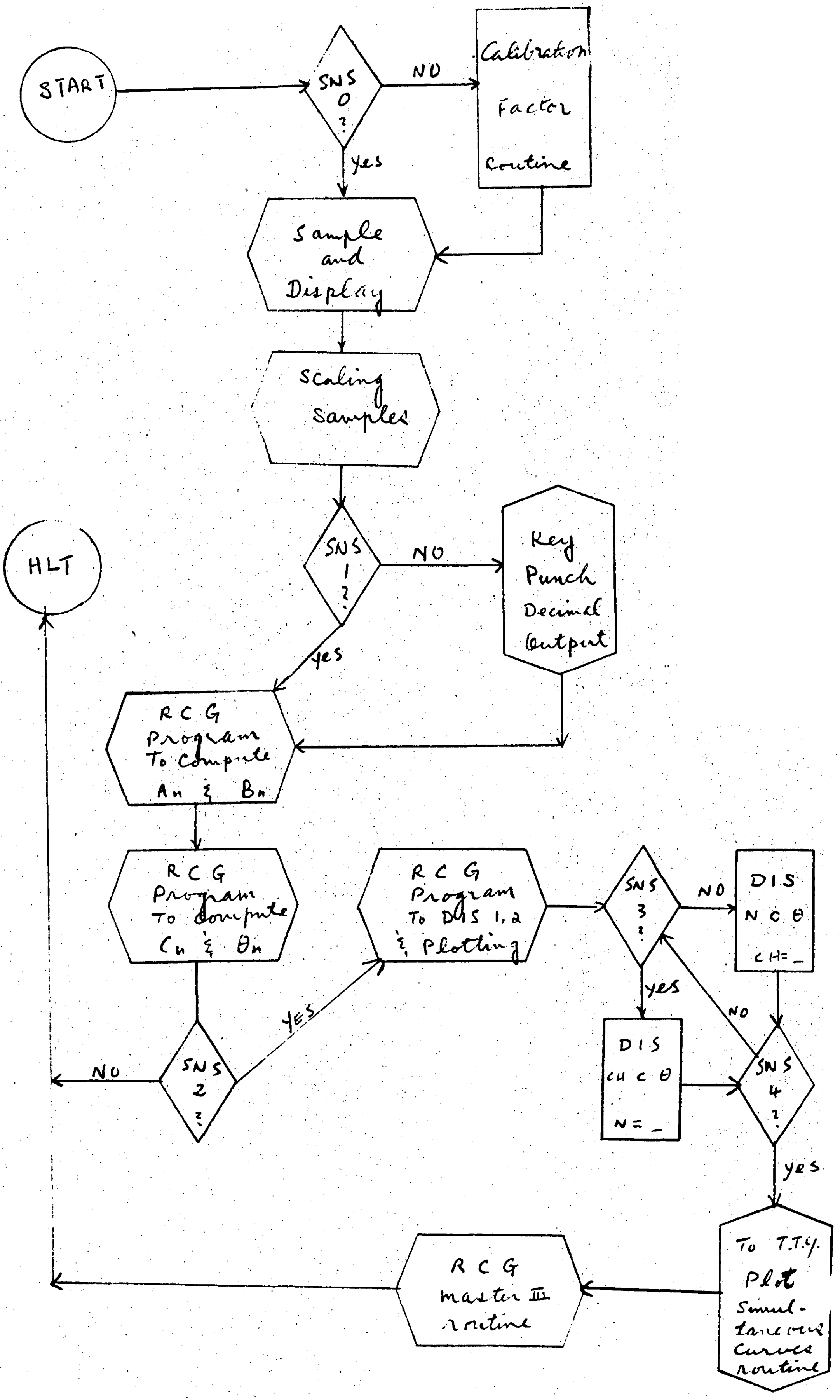
This routine first print out the time increment in ms (octal) and then plot four sets of pressure pulse data stored in the upper half of memory. Four curves are plotted simultaneously using the numbers 1, 2, 3 and 4 to distinguish the functions. The abscissa is represented as an incremented time base. Index register number 3 in # 75 contains P-1, where P is the total number of curves plotted; location 1744 contains the constant used to adjust base line; the round off routine is in 7U, this may be varied to suit the data to be plotted. The value C of the scaled samples varies as, $1400 \lesssim C \lesssim 700$; 200_8 locations are used to store a single set of data.

SNS 3 is used to select one of the display subroutine (either Display 1 or Display 2) and 4 ON is used to enter the plot routine; SNS 4 OFF provides for continuous display.

After plotting the curves, Master III routine is recovered by reading back its QN 1 and QN 2 portions.

The program halts and is ready for next sample and display by Start 20.

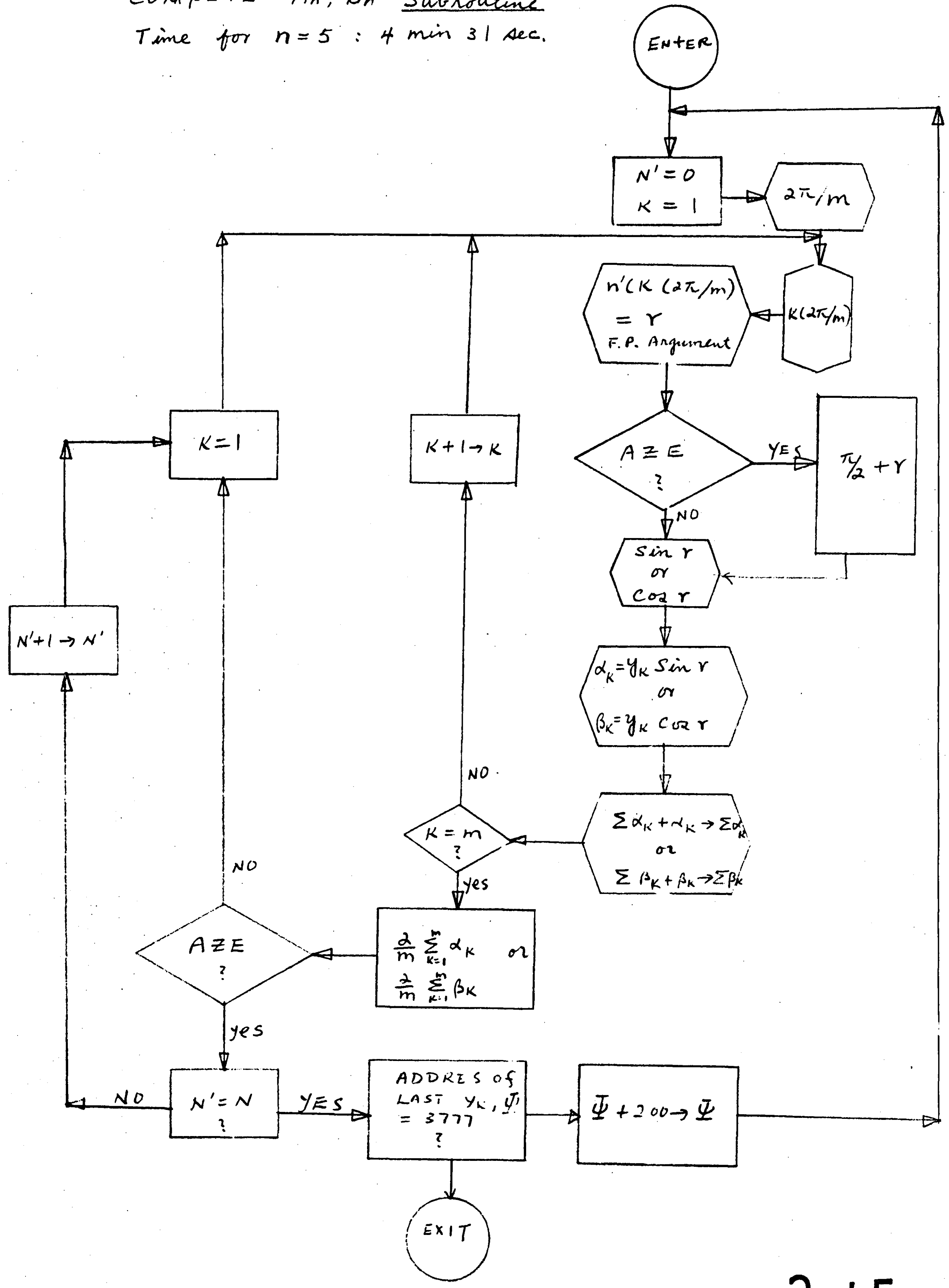
Master III Routine Flow Chart



FOURIER Analysis - 17'

COMPUTE A_n, B_n Subroutine

Time for $n = 5$: 4 min 31 sec.

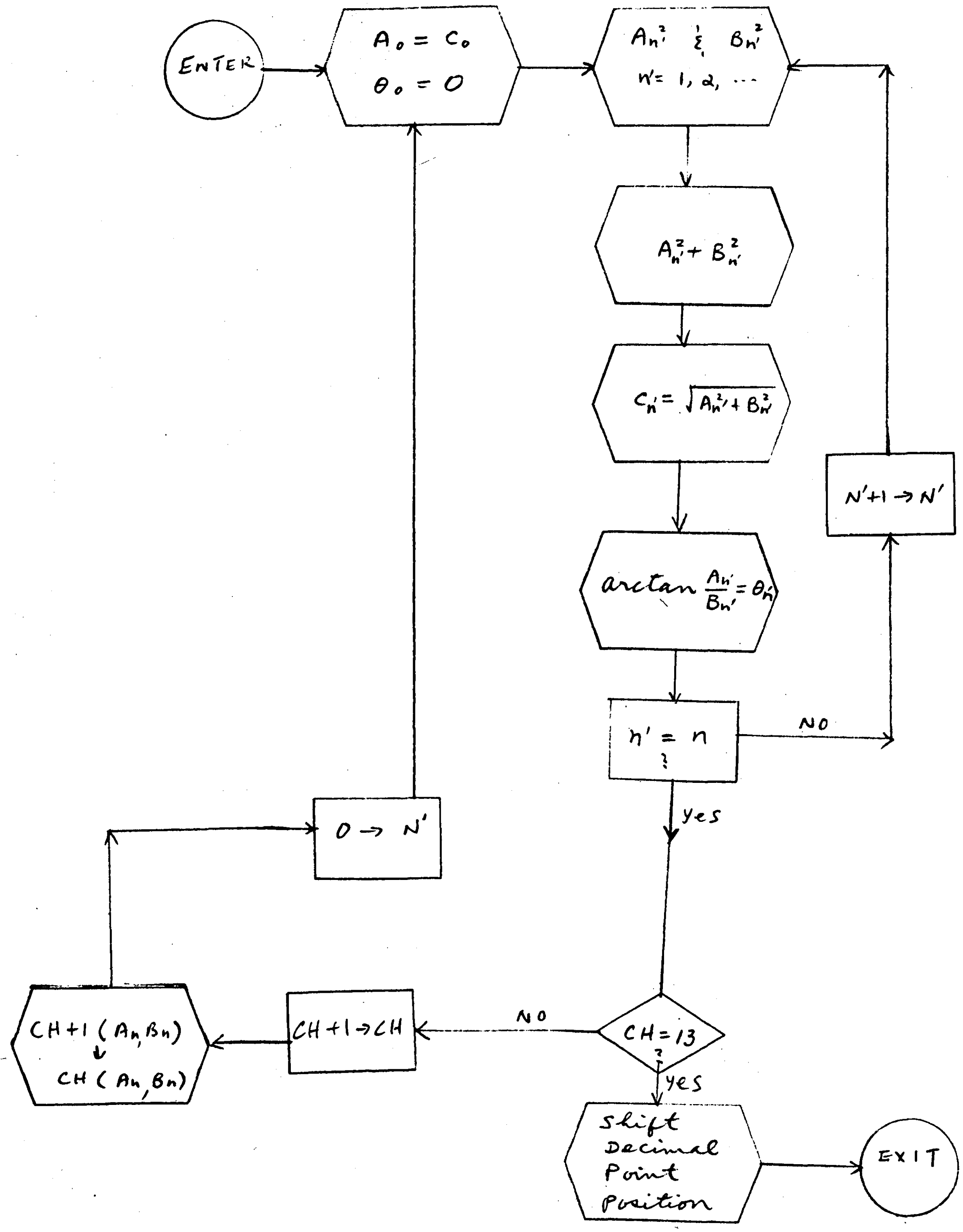


FOURIER Analysis -24

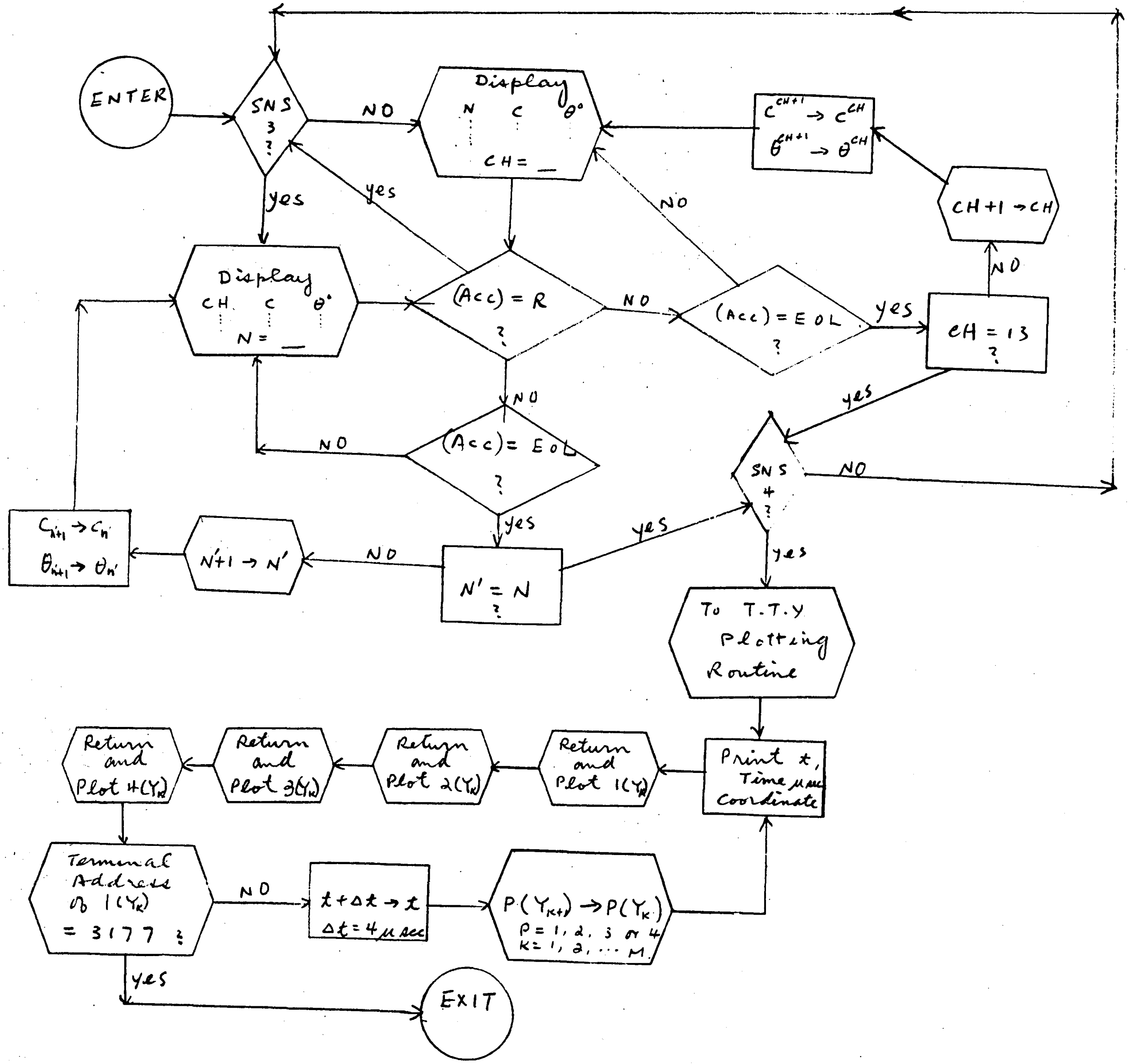
Compute C_n & θ_n Subroutine

From 4 sets of Data In-put From

Channel 10, 11, 12, & 13.



FOURIER Analysis -25
Display C_n and θ_n in
Two Formats and
Plot 4 Sets of Pressure
Pulse Data Simultaneously
Subroutine



Program Name: Binary to Decimal Conversion-V (with Fraction)

Author: GEORGE S. MALINDZAK, JR., PH.D.
DEPARTMENT OF PHYSIOLOGY
BOWMAN GRAY SCHOOL OF MEDICINE
WINSTON SA[illegible]

Address:

Phone: AC-919; 725-7251 x 7177

Subroutine: X Yes ______ No

If Manuscript:

No. Manuscript Lines: 424

Origin: 400

Entry Point: 1A

Tags: 1(A,B,C,D,E,F,G,S,T,U,X)
6(A,B,C,D,E,F,G,I,J,K,M,S,Y,Z)

Equalities:

If Octal:

Memory Locations:

Entry Point:

β Registers Used: 1,2,3,4,5,14,15,16,17

Analog Inputs Used (including knobs):

Operate Lines Used:

Tape Blocks and Units Used: LAP III Mss-5 Blks; Binary-1Qtr.

Description and Operating Procedures: Binary to Decimal Conversion-V: This routine will convert a 4 digit octal (12 bit binary) number to its decimal equivalent and display the value in the upper right quadrant on the oscilloscope. The program assumes the number to be converted is in the accumulator at the time of entry. The x and y coordinate is set at 415 and 465 respectively. The fraction conversion feature of the program starting at 1X assumes the number to be converted is a positive fraction. This routine was written for conversion of double precision numbers where the first datum word represented the integer value and the second datum work represented the fraction. In order to demonstrate these features the following utility program is suggested for use with this routine.

```
20 LSW
21 JMP 400
22 SET i1
23 465
24 LDA i
25 340
26 DSC i
27 0100
30 DSC i
31 000
32 LDA i
33 300
34 STC 415
35 RSW
36 JMP 530
37 JMP 20
```

Numbers enter through the LSW and RSW will be converted to their integer and fraction decimal equivalent and displayed on the oscilloscope. Without the utility routine only integers are converted and displayed.

Program Name: Master III Routine (Fourier Analysis 17)
Compute A_n and B_n

Author:

Address:
GEORGE S. MALINDZAK, JR., PH.D.
DEPARTMENT OF PHYSIOLOGY
BOWMAN GRAY SCHOOL OF MEDICINE
WINSTON-SALEM, NORTH CAROLINA

Phone:

Subroutine: X Yes ________ No

If Manuscript:

No. Manuscript Lines: 1374_8

Origin: ⊟ 400

Entry Point: $5Z-11_8$

Tags:
3(A,B,C,D,E,F,G,H,I,J,K)
3(L,M,N,O,U,X)
4(A,B)
5(A,B,C,D,G,J,N,M,O,P,Q)
5(U,V,W,X,Y,Z)
6(A,B,C,D,E,F,G,I,J,K,L)
6(M,S,Y,Z)
7(A,C,M,V)

Equalities:

If Octal:

Memory Locations:

Entry Point:

β Registers Used:

Analog Inputs Used (including knobs):

Operate Lines Used:

Tape Blocks and Units Used:

Description and Operating Procedures:

See attached description

Program Name: Master III Routine (Control Routine)
Calibration, Scaling, Keypunch

Author:

Address:
GEORGE S. MALINDZAK, JR., PH.D.
DEPARTMENT OF PHYSIOLOGY
BOWMAN GRAY SCHOOL OF MEDICINE
WINSTON-SALEM, NORTH CAROLINA

Phone:

Subroutine: ________ Yes ____X____ No

If Manuscript:

No. Manuscript Lines: 1367_8

Origin: # 1A

Entry Point: # 1A

Tags:
1(A,P,R,S,X)
2(A,B,C,K)
3(A,B,C,E,F,K,T,U)
4(A,B,D,E,G,K,L,M,N,R,Z)
5(A,B,C,D,E,F,G,J,K,L,M)
5(N,P,R,S,T,U,Y)
6(A,B,C,D,E,F,G,I,J,K)
6(M,S,Y,Z)
Channel 10,11,12,13

Equalities:

If Octal:

Memory Locations:

Entry Point:

β Registers Used: All

Analog Inputs Used (including knobs):

Operate Lines Used:

Tape Blocks and Units Used:

Description and Operating Procedures:

See attached description

Program Name: IBM 026 Keypunch Test Program-II

Author:

Address:

Phone: AC-919; 725-7251 x 7177

Subroutine: ________ Yes ___X___ No

If Manuscript:

No. Manuscript Lines: 124

Origin: 20

Entry Point: # 1A

Tags: 1(A, B, T)

Equalities:

If Octal:

Memory Locations:

Entry Point:

β Registers Used: 1, 2, 3

Analog Inputs Used (including knobs):

Operate Lines Used:

Tape Blocks and Units Used:

Description and Operating Procedures: Load binary version of program into QNO; ready keypunch by registering card from hopper, and start 20. Output is of the variety attached, and is recurrent until either the keypunch or the computer is halted.

Program Name: Fourier Analysis-11, Generation of Sin $N(K(2\pi/m))$ and Cos $n(K(2\pi/M))$ table

Author: GEORGE S. MALINDZAK, JR., PH.D.
DEPARTMENT OF PHYSIOLOGY
BOWMAN GRAY SCHOOL OF MEDICINE

Address: WINSTON SALEM [illegible]

Phone: AC-919; 725-7251; x 7177

Subroutine: ________ Yes ___X___ No

If Manuscript:

No. Manuscript Lines: 772_8

Origin: 20

Entry Point:

Tags: 3(G,H,I,J,K,L,M,N,O)
5(A,B,C,D,G,J,M,N,O,P,Q,U,V,W,X,Y,Z)
6(A,B,C,D,E,K)
7(A,C,M,V)

Equalities:

If Octal:

Memory Locations:

Entry Point:

β Registers Used: 1,2,3,4,5,7,10,11,12,13,14,15

Analog Inputs Used (including knobs):

Operate Lines Used:

Tape Blocks and Units Used: LAP III Mss-10 Bloks; Binary-2 qtrs

Description and Operating Procedures: This series of programs consist of three basic routines described blow:

A. Floating point $(N(K(2\pi/M))$, argument for the trigonometric functions.
   O N represents the number of harmonics
   K, $1 \leq K \leq M$, is the number of increments where M is the total numbers of samples in one period.

B. Sin $(N(K(2\pi/M))$ is computed if SNS 4 is off. Answer enters sine table at 2001 + (2M-1)

C. COS $(N(K(2\pi/M)))$ is computed if SNS 4 is off. Answer enters cosine table at 2401 to 2401 + (2M-1)

Each output value is in octal and is in double precision form occupying two consecutive locations with the octal point assumed between the two locations.

To Use: (1) After the binary program is read into quarter 0 and 1, put N into location 11 and M into location 12.

(2) To generate a cos $(N(K\ 2\pi/M))$ table at 2401-2401+(2M-1), set SNS 4 on and Start 20.

(3) To generate a sin $(N(K(2\pi/M)))$ table at 2001-2001+(2M-1), set SNS 4 off and Start 20.

Program Name: Plot Routine-5

Author: GEORGE S. MALINDZAK, JR., PH.D.
DEPARTMENT OF PHYSIOLOGY
BOWMAN GRAY SCHOOL OF MEDICINE
Address WINSTON-SALEM, NORTH CAROLINA

Phone: AC 919; 725-7251; x 7177

Subroutine: X Yes ________ No

If Manuscript:

No. Manuscript Lines: 274_8

Origin: 20

Entry Point: 1A-14_8

Tags: 1(A,B,C,D,G,J,I,Q,R,S,T,X)
2(A,B,C,F,G,K,L,M,N)

Equalities:

If Octal:

Memory Locations:

Entry Point:

β Registers Used: 1,2,3,4,7,10,12,13,14,14,16,17

Analog Inputs Used (including knobs):

Operate Lines Used:

Tape Blocks and Units Used: LAP III Mss-4 Blks.; Binary-1 Blk

Description and Operating Procedures: This routine plots four curves simultaneously using the numbers 1, 2, 3 and 4 to distinguish the function. The abscissa is represented as an incremental time base. Index register number 3 in 2R contains N-1, where N is the total number of curves plotted; location 273 contains the constant used to adjust base line; the round off routine is in 2G, this may be varied to suit the data to be plotted. The value of C in this program varies as, $1400 \leq C \leq 700$; 200_8 locations are used to store a single set of data. Initial address of the first set of data is assumed to be 3000. Terminal address is set to 3177

To Use: See Plot Routine-1

Program Name: Plot Routine-3

Author: GEORGE S. MALINDZAK, JR., PH.D.
DEPARTMENT OF PHYSIOLOGY
BOWMAN GRAY SCHOOL OF MEDICINE
Address: WINSTON-SALEM, NORTH CAROLINA

Phone: AC 919; 725-7251; x 7177

Subroutine: X Yes ________ No

If Manuscript:

No. Manuscript Lines: 245_8

Origin: 20

Entry Point: $1A-14_8$

Tags: 1(A,B,C,D,G,H,I,J,S,T,X)
2(A,B,C,F,G,K,L,M,N)

Equalities:
See Plot Routine-I for additional comments

If Octal:

Memory Locations:

Entry Point:

β Registers Used: 1,2,3,4,7,10,12,13,14,15,16,17

Analog Inputs Used (including knobs): None

Operate Lines Used:

Tape Blocks and Units Used: LAP III Mss-4 Blks; 1 Binary-Blk

Description and Operating Procedures: This routine plots a single function on the TTY according to the contents, C. With SNS 5 on, the routine plots a point representing the function and its octal value along the ordinate, with SNS 5 off the routine plots a point representing the function only. Memory location is plotted as the abscissa. $1400_8 \geq C \geq 700$ for this particular program but this may be easily modified to plot any function. Loaction 214 contains the constant to adjust the base line. The round off of the last digit routine in 2G may be varied to make any data fit for plotting.

To Use: Same instructions as Plot Routine-1

Program Name: Plot Routine-1

Author: GEORGE B. MALINDZAK, JR., PH.D.
DEPARTMENT OF PHYSIOLOGY
BOWMAN GRAY SCHOOL OF MEDICINE
Address: WINSTON SALEM, NORTH CAROLINA

Phone: AC 919; 725-7251; x 7177

Subroutine: X Yes ________ No

If Manuscript:

No. Manuscript Lines: 232_8

Origin: 20

Entry Point: # $1A-14_8$

Tags: 1(A,B,C,D,G,J,S,T,X)
2(A,B,C,F,G,K,L,M,N)

Equalities: None

If Octal:

Memory Locations:

Entry Point:

β Registers Used: 1,2,3,4,7,10,12,13,14,15,16,17

Analog Inputs Used (including knobs): None

Operate Lines Used: None

Tape Blocks and Units Used: LAP III Mss-4 Blks; Binary - 1 Blk

Description and Operating Procedures: This routine uses the TTY to print out a function stored in memory with its location along the abscissa and its contents as a series of dots (.) and octal value along the ordinate. The maximum number of dots printed out is equal to C, where $65_8 \geq C \geq 0$.

To Use:

(1) Read function data into any quarter but QNO
(2) Read binary program into QNO
(3) Turn on teletype and start 20
(4) Type on TTY (P) @ , where P is the starting address of the function data, and (Q) E, where Q is the terminal address of the function data
(5) Set paper hit "return", and away you go

Conventional delete feature from keyboard through the RUB OUT key is included.

Program Name: Variable Sample and Display-IV

Author: GEORGE S. MALINDZAK, JR., PH.D.
DEPARTMENT OF PHYSIOLOGY
BOWMAN GRAY SCHOOL OF MEDICINE
Address: WINSTON-SALEM, NORTH CAROLINA

Phone: AC 919; 725-7251; x 7177

Subroutine: ________ Yes ____X____ No

If Manuscript:

No. Manuscript Lines:

Origin: 20

Entry Point: # 1Z

Tags: 1(A,B,C,D,E,F,G,H,I,J,S,T,U,Z) 5 (D,N)
2(B,C,E,F,G,S) 6 (D,N)
3(S) 7 (A,D,M,N,S)
4(D)

Equalities:

If Octal:

Memory Locations:

Entry Point:

β Registers Used: (1,2,e,11,12,13,14,15,16,17)

Analog Inputs Used (including knobs):

Operate Lines Used:

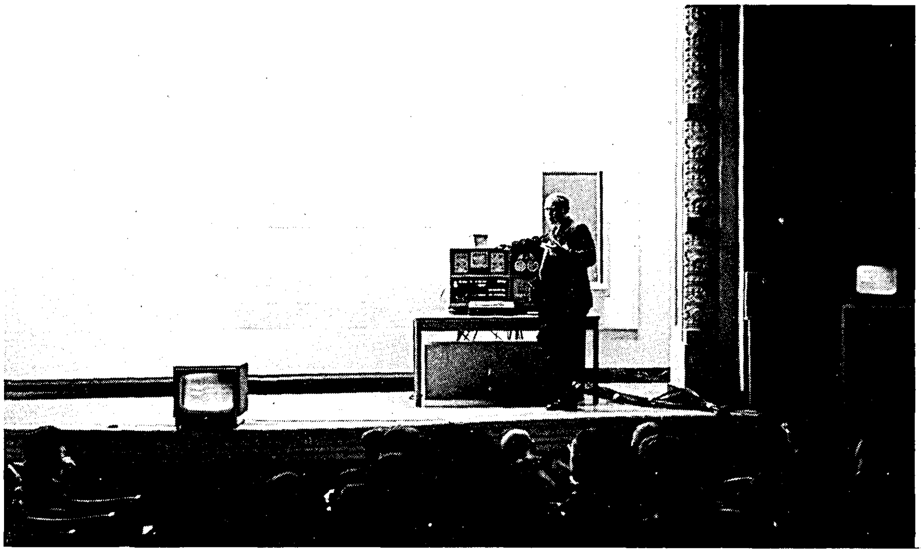
Tape Blocks and Units Used:

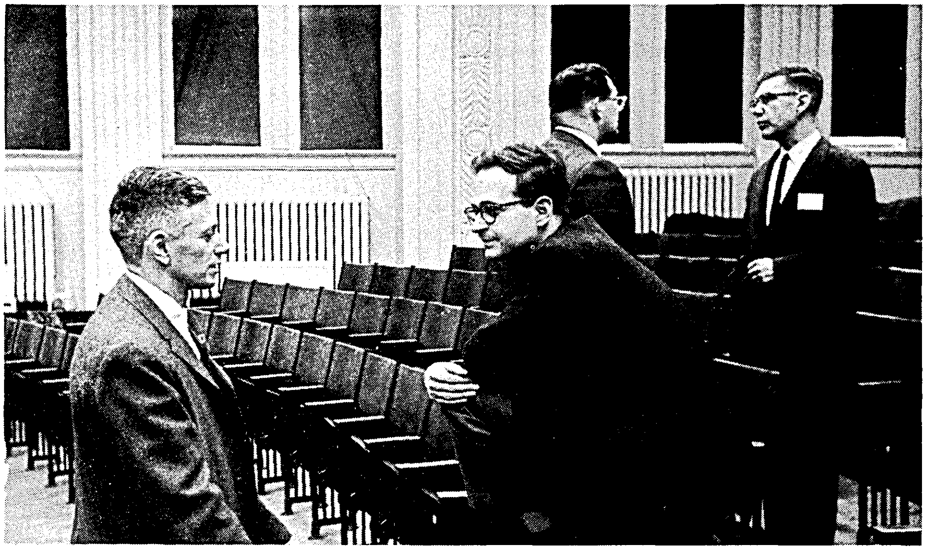
Description and Operating Procedures: This program is designed to sample and display analog information presented to LINC over ANCH10 at a sample rate variable from 15.6 to 1000/sec. There are several options within the program that permit one a good deal of flexibility in data reduction and preprocessing of experimental information. The delay between samples is determined prior to program execution and set in the sense switches (0-5). With all sense switch off the basic sampling rate is I KC; with SNS 0 on, the rate is 500/sec or 1 every 2 ms; with SNS 0, and SNS 1 on, the delay between samples is 6 ms;etc., with SNS 0, SNS 1, SNS 2, SNS 3, SNS 4, SNS 5 on, the delay between samples is 64 ms, and the sampling rate is about 15.6/sec.

Once the sampling rate has been determining start 20 will initiate the program; once sampling is complete, memory is displayed 1000_8 locations at a time. The location in memory from which the display starts may be set by pot # 6 (course) and pot # 7 (fine). The upper half of memory may be displayed. The x-axis may be expanded by adjusting pot # 4. Sixteen possible expansion ranges are available. Pot # 2 adjusts the length of the cursor which may be moved along the data through manipulation of pot # 0 (course) and pot # 1 (fine). Once the cursor is in the display one has the option of an octal or decimal display if the point beneath the cursor (upper right quadrant of the scope). SNS 1 on, the number displayed will be decimal; SNS 1 off the number displayed will be octal. In this octal display, the number may be

Description and Operating Procedures continued:

either signed or unsigned depending on the positions of SNS 0; in the decimal display there is no sign option; all numbers are signed. During all numerical displays, the location from which a point is obtained for the display beneath the cursor is shown in the lower right quadrant of the scope.





Report on Use of LINC Computer Through February, 1965, and Proposal for Its Continued Use

Donald S. Blough and Lloyd Marlowe
Department of Psychology
Brown University

3-1

Report on Use of LINC Computer Through February, 1965, and Proposal for Its Continued Use

Donald S. Blough and Lloyd Marlowe

Department of Psychology

Brown University

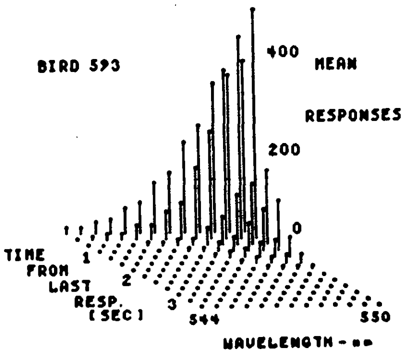
I. Past and present research

A. Nature of the research. The LINC has been used primarily for on-line control and data collection of operant conditioning experiments. Six pigeon boxes are connected to the machine (see below). To date some half-dozen experiments have been run. These experiments were concerned with stimulus control (discrimination and generalization) processes, and with reinforcement scheduling procedures. In most of the work, distributions of inter-response times were of central concern. Three of the most extensive experiments will be described briefly.

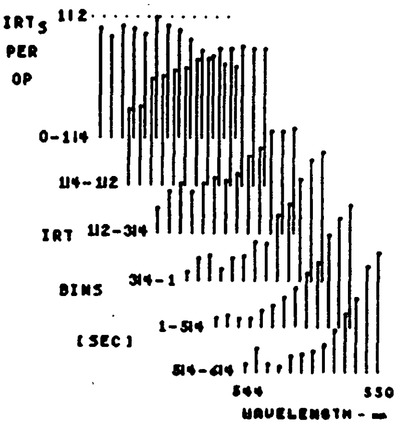
1) Inter-response times during generalization testing. It is known that rate of response declines when stimuli present in the experimental situation depart from those values that are associated with reinforcement. We have been interested in the detailed nature of this change in rate. Our point of view has been that responses are, to a degree, "chained" together, such that the probability of a response being emitted at a particular moment is a joint function of current stimuli and of preceding behavior. To deal with this question, pigeons were trained to peck at keys illuminated from behind by lights of fixed wavelength and intensity. Following a number of sessions of such training, food reinforcement ceased and the light on the key varied randomly in either wavelength or intensity. Responses emitted during these changes were categorized by both stimulus and inter-response time. The LINC computed conditional probabilities that a response would occur, given that a particular stimulus was present and that a particular interval had elapsed since the previous response. It was found that for IRTs under about 3/4 sec response probability varied little with stimulus variation. Such responses, constituting well over half of the total emitted, were evidently chained to previous behavior and not under stimulus control.

2) Inter-response times during maintained discrimination testing. The generalization procedure is unsatisfactory for quantitative analysis of response probabilities, because the subjects are tested without reinforcement. Relatively few responses occur before extinction, and those that do occur take place during the transitory extinction process, when IRTs are changing as a function of the time since the last reinforcement. For this and other reasons, a steady-state discrimination experiment was run. Three pigeons were run simultaneously, each pecking a key illuminated by one of 13 wavelengths spaced at 1 mu intervals from 538 mu to 550 mu. All stimuli were presented for an equal number of brief trials during a daily session, and pecking went unreinforced to all. Interspersed among these unreinforced presentations were occasional extra presentations of 550 mu; on these trials, pecking was reinforced on a variable interval schedule. A relatively stable discrimination function

developed in this situation, with rapid responding at 550 mu and gradually less responding at wavelengths progressively farther away. The number of responses given to each stimulus showed a highly regular relation to stimulus wavelength. These data were analysed as a joint function of inter-response time and stimulus. A three-dimensional plot of such a function appears in Fig. 1.



However, a plot such as that in Fig. 1 does not clearly separate the action of stimulus and IRT variables. For this, the probability of response conditional on the occurrence of any particular IRT must be computed. This, the so-called "IRTs per opportunity" transformation, reveals the same finding that was suggested by the generalization studies. As shown in Fig. 2 the stimulus has relatively little influence on responses terminating short IRTs, while a regular declining function governs more widely spaced responses.



3) <u>The reinforcement of least-frequent interresponse times</u>. As the

above discrimination experiment continued, response rates increased but stimulus control became progressively more variable. We felt that this might be due to the phenomenon we had been investigating. That is, progressively stronger interresponse "chaining" might gradually override the control exerted by exteroceptive stimuli. For this and other reasons, we devised a schedule of reinforcement specifically designed to minimize chaining and to favor the emission of responses independent of one another. The appended report (Appendix A), presently under review by Psychonomic Science, gives a brief account of this schedule and the early results obtained with it. Machine simulation of "statistical pigeons" was a helpful adjunct to this work.

The above three examples suggest the core of Blough's research with the LINC. In addition, three graduate students have made extensive use of the machine. Mr. Jeffrey Sheff has conducted an investigation of wavelength difference limens, using the machine to control stimuli and to collect, analyze and display data. Mr. Charles Shimp has analyzed extensive data for his doctoral dissertation on discrete-trial discriminations. The experiments were run off-line, and the data fed to the computer through a paper-tape reader. The LINC provided sequential statistics that could be compared with existing mathematical formulations of discrimination learning. Mr. Shimp also programmed the machine to simulate pigeon performances, given certain assumptions about the variables controlling choice behavior, with impressive results. Mr. Lloyd Marlowe has done extensive programming on the machine (see below and Appendix B). He is also running an experiment on line, aimed eventually at studying time discrimination.

B. Inputs, outputs and auxilliary equipment.

Additions to the data terminal box. A 32 bit external clock, driven by machine time pulses, was installed last fall. This is used for the many timing jobs involved in on-line operant work. Three different 12 bit time ranges may be selected via OPR 1, 2, or 3. Twelve auxilliary relays were also installed, driven by a 12 bit register installed in the DTB. These are used to control the multiple stimuli in the pigeon boxes. Output to these is parallel, with the left or right 6 bits independently selected via OPR 5 or 6.

Standard inputs. At present, 6 experimental chambers are connected to the LINC. The 12 XL lines are used to sense responses and, in certain cases, other experimental events. The analog lines have been used in a servo-type control loop, by which the machine adjusts stimulus wavelength and intensity to pre-programmed values. White noise on one analog line has been used as a source of small random numbers. The OPR input lines, in addition to serving the external clock, have been used to enter stimulus codes.

Standard outputs. The relay register and the 12 new auxilliary relays are in constant use controlling stimulus and reinforcement events. They switch both 28 volt DC and 110 volt AC circuits. In the servo-type stimulus system mentioned above, the relays run wavelength and intensity drive motors. The Teletype, installed last fall, has been used very heavily to type out programs, data tables, and data histograms. The scope serves for on-line data monitoring, program displays, and for graphing data, in addition to heavy use with the LAP compiler. Experimental data are stored permanently on magnetic tape.

C. Programs.

Perhaps fifty programs have been written, for a variety of purposes, including: on-line experimental control and data collection; simple data analysis; display via scope or teletype of instructions for running programs, data tables, graphs, and so on; simulation of animal behavior under a variety of conditions; tests of various input and output functions. Soon after we started using the machine it became evident that convenient systems for running these programs would be needed. First a "meta" system was written, which permitted programs to be called systematically, and parameters to be typed into program displays. Recently, Mr. Marlowe wrote the "Monitor" system (described in appendix B) which enables us to assemble and run at will long sequences of individual programs.

In addition to the meta-monitor scheme we have a few programs that other LINC users might find helpful. These include several display programs, including tabular and graphical display, and data packing and rearranging programs. These programs are written to be run by Monitor. Copies of these programs, as well as the Monitor, will be available on tape at the March, 1965 LINC conference or upon request. It would be difficult to describe our on-line experimental programs in detail, largely because they take care of so many experimental events and contingencies that make sense only on exhaustive description of the experimental rationale and procedures. Appendix A includes a copy of the program used to run the experiment described therein, as well as a sample of the daily data printout obtained from the experiment.

II. Future research

Our research now depends entirely on the LINC. It not only enables us to do more easily what we formerly did with relay and solid-state programming equipment, but it has opened up entirely new possibilities. Most of the research we are now doing and plan to do would be impossible without the machine.

A. Nature of future work.

This will grow out of our past research, some of which is briefly described above. We plan to continue to develop and study the random-response schedules made possible by having LINC on line. We intend to apply such schedules to our stimulus control experiments, in the hope that they will drastically reduce individual differences and make possible a rational quantitative treatment of the rate measure. Quantitative functions will be sought relating rate to reinforcement probability and several stimulus parameters. We hope to continue discrete trial work and to begin work with response latency, relating both of these behavioral measures to the rate measure. The display capacities of the LINC encourage us to add some human perceptual studies to our program, if time permits.

Though the machine is now used in our own program for from 8 to 16 hours a day, there are tentative plans to use the machine for the analysis of taped electrophysiological data from Dr. Carl Pfaffmann's laboratory here at Brown. Since Pfaffmann's recent acceptance of an appointment at the Rockefeller Inst., this use is uncertain. One or more of his students may do the work, or Dr. Corbit, our new physiological man, may be interested in this application.

B. Inputs, outputs and auxilliary equipment.

Orders were recently placed for relays and logic elements to expand our input and output facilities. The auxilliary relay bank will be increased to 48 relays, enough to control 8 independent events in each of our 6 experimental boxes. We have no plans to expand the number of simultaneously run animals beyond 6, as this already strains the machine's memory capacity. At the same time, we will shift from XL to parallel reading of response events, for reasons of speed, flexibility, and reliability. External registers will hold "responses" until these are read in, at which time the response registers will be reset.

There is some chance, perhaps slight at present, that our LINC may communicate via a teletype line with the regional computer complex at MIT. The Brown Computation Laboratory at present rents such a line at low cost. At the moment we have no pressing problems that call for this tie-in, and, of course, the rate of data transmission would be low.

C. Programs.

Marlowe hopes to begin work this summer on a special purpose compiler for use in operant research. The hope would be to enable the experimenter to write out experimental contingencies in an approximation to English, LINC would then compile a machine-language program that would run the experiment. Such a compiler would speed up our programming a great deal, and would enable students to run experiments with minimal knowledge of the machine. Tentative plans have been made for Marlowe to work out the compiler using SNOBOL and the existing teletype link to the large computer facilities at M.I.T.

III. Training program

I think the evaluation program has generally worked well as a training device. Though I started with no special knowledge of any computer, I have been able to make effective use of the LINC, and to keep it going with relatively little outside help. However, for those of us with no programming background, more specific instruction would have been helpful. It is my own belief that making available a graded series of pre-written, annotated programs for newcomers to study would have been the easiest, most helpful device. Detailed handbooks on the nature of the machine and its programming would be especially valuable to the many individuals now involved with the machines who were not in the original group in the summer of 1963. The arduous and no doubt uninteresting work of preparing such material might in the end be of more service than equal effort on technical matters.

IV. Computer performance

A. Maintenance.

We have had relatively little trouble with the machine. As of March 1, 1965 it had run for 2700 hours. Two failures in the logic were detected in the first 6 months, both apparently due to overloading caused by faulty wiring (one, a wire missing from the frame, the other an open clamped load on one module). A transistor in one of the relay cases also failed. Sticking keys on the Soroban keyboard were fixed by careful oiling. The major problem from the beginning was electrical noise; grounding, shielding, spark suppression, and termination of all open lines to the DTB seem to have eliminated the difficulty. The machine is now grounded at one point only, through a heavy strap attached to the fantail.

Various bothersome tape difficulties have arisen from time to time. Last summer, tapes tended to bind on the shoes. This may have been due to the particular atmospheric conditions in our air-conditioned and humidity-controlled quarters. Mark clock and ACIP adjustments fixed some marking and writing problems. A recent intermittent failure to RDC, due to a disagreement between the data sum and the check sum, has not yet been traced. The tape unit fuse has also blown twice in recent weeks. This may be due to a recurrence of previous trouble caused by a short-circuit between NO and NC contacts on a tape control relay.

Burning on the scope screen was a serious problem with the original CRT. Display channel separation was also failing on the old tube before it was recently replaced with the new aluminized CRT. Within a week the screen location occupied by the LAP line number was noticeably burned on the new CRT. The problem arises from the nature of the LAP displays. If the brightness for a single line is turned down to avoid burning, the 20 line display is not bright enough to read. Using the button to brighten the screen in the latter case is quite awkward when one is shuffling manuscript pages. A possible solution is the adjustment of display loops to compensate for the size of the display. The analog ladders controlling the display required adjustment after about 1000 hours.

B. The LINC in operant research.

In many respects, we have found that the LINC is a very good laboratory computer. Rather than listing in detail its many good points and its few bad points, we shall discuss the LINC's capabilities in relation to a couple of problems peculiar to behavioral research.

1) Special time factors in operant conditioning experiments. In our free operant experiments, the subject (a pigeon) can make a response at any time with a minimum interresponse time of about 50 milliseconds. Also there are stimulus changes contingent on these responses which should appear to be an immediate consequence of a response. In most situations, a delay of about 60 milliseconds from response onset to stimulus change is the maximum delay that

should be allowed. If delays become very much greater than this, the delay before the stimulus change is detectable.

2) Adequacy of core program storage capacity. As a result of these special time factors it is frequently necessary to carry out all of the computation associated with a response in less than 50 milliseconds. Consequently, the program storage capacity has usually been adequate for running our experiments. A good example of this is the program included in Appendix A. Although this program takes almost 50 milliseconds to perform the necessary calculations, if all 3 subjects respond simultaneously, it requires only 3 blocks of program storage. Nevertheless, we can conceive of situations in which we would be pressed for program storage.

3) Inability to use tapes during experiments. A second result of the special time factors mentioned above is the inability to use the tape units during an experiment. Any tape operation will result in the possible delay of a stimulus change which can not be tolerated in many experiments. Of course there are some experiments in which there are times when these time factors are not in effect. However, in most of our experiments, it is virtually impossible to have these times coincide for each of 3 subjects being run simultaneously since the subjects must necessarily be run independently of one another. The problem becomes worse if we consider running more than one experiment simultaneously.

The major problem produced by this restriction is the inability to store all of the raw data (over 10,000 responses may be collected from 3 subjects in one session). Instead, the data must be at least partially reduced as it is collected. This restriction forces the experimenter to make a priori guesses about what data he can ignore. In addition to losing data, valuable programming time is lost in reducing the data. For example, about half of the calculation time in the program in Appendix A is spent in reducing and recording the data. This time could be spent running a second experiment.

Another problem produced by this restriction is the inability to read in other programs while an experiment is being run. For example, in the experiment in Appendix A, the average total response rate of all 3 subjects is less than 3 responses per second. In other words, on the average, only 50 milliseconds out of each second are used to run the experiment. This leaves plenty of time for doing other things such as typing in programs and analyzing data. However, due to the time factors in our experiments and the slow tape operations, it is not possible to take advantage of this dead time.

4) Possible solution - input/output processor. A possible solution is the addition of an input/output processor and a 256 word core buffer storage to handle the reading and writing of tapes. A LINC with this addition would look very much like two computers connected by a 12 bit parallel transfer cable and a few synchronizing lines. The first computer would look like the present LINC without any tape handling capabilities. The second computer would look like the present LINC with only the tape handling and operate instructions. When computer #1 needs information stored on tape, it executes a read tape

instruction which tells computer #2 to read the specified block into the buffer. In the meantime, computer #1 continues executing instructions and periodically checks an external level which will be turned on by computer #2 after the block has been read into the buffer. When the external level is detected by computer #1, it is reset. Then computer #1 executes a four register instruction which, in the last 3 registers, states 1) the number of registers to be transferred from the buffer to the main core memory, 2) the first register in the buffer to be transferred, and 3) the first storage register in the main core memory. A similar instruction would be used to transfer information from the main core memory to the buffer. A write tape instruction would write the buffer into the correct block and set an external level for computer #1. By making the transfer instructions completely general and separate from the read and write instructions, it is possible to read any set of registers in any block into any set of registers in the main core memory and vice versa for the write instruction. Also, it is possible to make any number of transfers going either direction between successive reading and writing instructions. Since the LINC can transfer information at the rate of 125 words per millisecond, computer #1 can transfer information to and from the buffer at its convenience and without producing intolerable delays.

Quite clearly the addition of this sort of input/output processor would greatly increase the flexibility of the LINC. Data could be periodically written onto tape without interrupting experiments. Experiments could be run simultaneously and could be started and stopped independently. Data analysis and programming could be performed in the dead time between the disposing of responses during experiments.

5) Changes in online input/output facilities. Generally speaking, operant conditioning experiments treat a response as a digit signal - i.e. it is either on or off. Consequently, we do not have much need for the analog-to-digital conversion units. However, in the future, more interests may develop in recording responses on a multivalued scale. Therefore, for our purposes, it would be best to make the analog-to-digital converters an optional feature. This is true for the knobs as well as for the online converters since the keyboard can do almost everything that the knobs can do, and do it more precisely.

In place of the digital-to-analog converters, we would prefer to have a more elaborate digital input system such as the one we are currently planning. In this system there are two banks of 12 digital response detectors. When the onset of a response occurs the response detector is turned on. It remains on until the computer reads the 12 bit bank, containing that response detector, into the accumulator over the TN line. After the response detector is reset by the computer, it stays off until another response is detected. After the content of the response detector bank is stored, it is checked bit by bit to determine which, if any, responses have occurred. If a bit is on, a table look up procedure is used to determine the location of the subroutine contingent on that bit. Although this approach may be slower than reading consecutive SXL lines, it makes it possible to read all of the inputs from one organism simultaneously. Also, the table look up system can be part of an "interpreter" which interprets call sequences, stored in the second half of memory, for subroutines stored in the first half of memory. Also, the response detector holds the response until the computer has time to check it. This can not be done directly with the current SXL lines.

It would be nice to expand the output facilities, also, since many events must be under the control of the computer. For example, in each of 3 of our experimental chambers there are 7 lights and a feeding mechanism which must all be under independent control. In some experiments, the number of relays required can be reduced by using combinations of relays to switch a certain event. However, this makes it difficult to run more than one experiment in a chamber since different experiments will frequently require different relay combinations. To cope with this problem, we have already added 12 relays and are preparing to install another 36 relays.

Another input/output facility that would be very useful would be an external shift register to handle the timing and shifting required by the teletype. This would make it easier to run the teletype during experiments.

6) Automatic interrupt system. If data processing and online control are being performed simultaneously, it is necessary to go periodically (at least once per 5 milliseconds) into the online mode to check all the inputs. If nothing is found, valuable time is lost. Also, there is a variable delay (up to 5 milliseconds) added to the usual delay from response onset until stimulus change. Just as important, the data processing program must be written with special exits scattered through it, and considerable care must be taken to make sure that an endless loop or some other programming error does not occur.

These problems could be solved by adding a very simple interrupt system which would jump the computer to the input scanning routine whenever the computer received an input from a response detector, from the keyboard, or from the tape buffer. The interrupt system would store a jump instruction in a register, other than the index registers, which would be used to resume data processing when the input was disposed of. The scanning routine would save the accumulator contents before doing anything else, and put this value back in the accumulator before executing the "resume" jump. A more elaborate interrupt system is not necessary since, once in the online mode, everything has top priority, i.e. nothing can be interrupted.

7) More soft ware. Most of the other problems connected with the LINC are programming problems. Although LAP has been an immense help, it does not do the work of a compiler. At present, considerable time must be spent to perform what appear to be relatively simple problems. This situation could be alleviated by writing arithmetic and experiment oriented compilers. A more feasible and flexible approach may be to write a macro assembly program and a large library of arithmetic subroutines. For example it would be very nice to have 1) subroutines for adding, subtracting, multiplying, dividing, and taking square roots for both fixed and floating point, double-precision numbers and 2) a quick and simple method for assembling them into a program.

Bibliography: Talks and papers on work done with the LINC

"IRT's, Generalization, and the LINC" - Colloquium, Harvard Univ. Dept. of Psychology, March 18, 1964

"Inter-response times during stimulus generalization tests" - Paper read at Eastern Psychological Assn. meetings, Phila., April 17, 1964

"Stimulus control of response probability" - Talk, Lab. of Communication Biophysics, M. I. T., January 14, 1965

"Recent research in operant conditioning" - Lexture series, The Rockefeller Institute, Feb. 15 - 26, 1965

"Reinforcement schedules and stimulus control" - Talk, operant conditioning group, Columbia University Dept. of Psychology, Feb. 17, 1965

"Reinforcement of Least-frequent Interresponse Times" - paper submitted to Psychonomic Science, March , 1965 (copy attached) (Appendix A)

"Momentary Response Probability during Generalization and Discrimination" - paper in preparation

APPENDIX A

A sample operant experiment run by the LINC

- Part 1 - Description of the experiment
- Part 2 - Sample of daily data printout
- Part 3 - Copy of on-line program

Appendix A - Part 1

Reinforcement of Least-frequent Interresponse Times

Donald S. Blough

Brown University

Abstract

Pigeon subjects pecked a key for intermittent food reinforcement. An on-line digital computer (the LINC) controlled the experiment. It reinforced only those responses that terminated inter-response times which had occurred least frequently in the immediate past. This schedule discouraged response-to-response dependency, while generating stable response rates and uniform behavior among subjects.

The rate with which an animal emits a simple response can be an effective measure of the influence of stimulus and other variables upon behavior. However, in most if not all operant conditioning situations, response rate is also governed by inter-response dependencies. When behavior is maintained for many hours by intermittent reinforcement, behavior patterns emerge and responses become "chained" together. This influence is hard to evaluate and to control, yet inter-response dependencies sometimes become so strong that rate is quite insensitive to other variables.

Inter-response dependencies can often be traced to stereotyped behaviors that fill the interval between recorded responses. Most schedules of reinforcement favor the growth of such behavior patterns by failing to control the behavior that comes just prior to the reinforced response. When such control is lacking, subjects tend to settle into "superstitious" patterns of behavior [1] that generate varying rates and patterns of response. Some schedules specify that the subject must space its responses by a certain interval in order to receive reinforcement. Here, interresponse dependencies are not left to chance, but are specifically generated by the schedule of reinforcement. Whether arising by chance or by design, interresponse dependencies ordinarily reveal themselves in non-random patterns of inter-response times (IRTs). Certain IRTs occur more often than one would expect by chance, others less.

The present schedule of reinforcement [2] favors random responding by forcing the subject to emit distributions of IRTs that approximate those of an ideal random emitter. Such an ideal subject, emitting responses randomly in time, generates an interresponse-time distribution described by

$$f(t) = \lambda e^{-\lambda t}$$

where λ represents the mean rate per unit time, and t represents the time between responses [3]. Physical processes that involve independent events (e.g.-radioactive disintegrations) produce IRT data of this form. The shape of the function is suggested by the data in Fig. 1.

To make this ideal distribution correspond even roughly to real behavior, we must first recognize that subjects have a certain minimum IRT, and that very short IRTs have special characteristics (see below). These facts raise several difficult problems, but as an approximation we assume that responses can be repeated after a brief fixed interval of time. The IRT distribution effectively begins at this time, set at 0.8 sec. in the experiment described below. Next, starting at this arbitrary origin, we mark off 16 intervals, or "bins", along the time axis. These bins are chosen such that, for a given rate λ, they divide the area under the IRT curve into 16 equal parts, (See "bin spacing", Fig. 1). This means that the ideal random emitter, responding at rate λ, would on the average drop the same number of IRTs into each bin.

This ideal model controls the performance of the real subject. A computer, the LINC [4] senses responses as they occur, computes their IRTs, and accumulates these IRTs in the 16 bins just described. Only the most recent 64 (or, recently, 152) IRTs are saved, however. This distribution of recent IRTs is used to determine the next response to be reinforced. When a reinforcement occurs, the LINC scans the 16 IRT bins and notes which bin contains the fewest IRTs. This bin becomes "hot": a response is now reinforced only if it follows the preceding response by an IRT that falls into this bin. That is to say, the subject is reinforced for making his "least probable" IRT, as estimated from his recent performance.

This schedule discourages interresponse dependence, expressed as particularly frequent IRTs. If, for example, a pigeon subject starts out with a stereotyped "bow" after each response, it will produce many IRTs that equal the time it takes to bow and peck. The IRT bin corresponding to this time will thus be relatively full, so IRTs of this length will not be reinforced. As a result, bowing should decrease in frequency until it becomes no more likely than other movements.

Such reinforcement of least frequent IRT is the defining characteristic of the present schedule. Our application of the schedule has other details that arise from the realities of reinforcement and animal behavior, all of which cannot be described in this brief report. Among them are the following: (1) The first response after reinforcement is never reinforced, nor is its IRT recorded. (2) Responses terminating IRTs of less than 0.8 sec. are never reinforced. A reinforcement in progress terminates if such an IRT occurs. Our data suggest [5] that such short IRTs are relentlessly non-random, and that they are not controlled by the same variables that affect longer IRTs. (3) Frequency of reinforcement may be varied by omitting reinforcement of responses otherwise scheduled to receive it. So far we have used probabilities of 1.0, 1/2, and 1/4 that a response will be reinforced, given that it terminates a "least likely" IRT.

The following describes a portion of our work with the schedule just described. Three pigeons were used. Two of the birds had been on a variable-interval schedule of reinforcement for many hours previously, and they had developed quite different rates and patterns of response. A third bird was put on the present schedule after only about 2 hours of variable interval reinforcement. All birds were maintained at about 75% of their ad lib weights. They worked daily for 80 min. in standard experimental chambers, pecking back-lighted keys that required 10 grams of pressure for electrical operation. Reinforcement consisted of 4 sec. (later, 3.3 sec.) access to mixed grain. The LINC computer ran the three birds simultaneously, programming reinforcements from the special IRT bins described above, with $\lambda = 1/2$. It also recorded responses into standard 0.1 sec. and 0.5 sec. IRT bins, and compiled a running cumulative-response record for each bird. Between sessions, the LINC analyzed the data and produced displays, two of which serve as figures for this report.

Figure 1 shows mean frequencies of interresponse times in successive 0.5 sec. bins. The means cover six daily 80 min. sessions for each bird; IRTs under 1 sec. are omitted in the Figure. The distribution of IRTs within the special reinforcement bins is not shown here, although the limits of these bins are indicated ("bin spacing"). Figure 2 shows the data of Figure 1 transformed by dividing each IRT value by the number of "opportunities" that IRT had to occur. This transformation estimates the conditional probability that a response will occur, given that a particular time has elapsed since the previous response [6]. For the ideal random emitter this transformation yields a horizontal line. It will be seen that the data here approach this ideal, though points under one second fail to conform, and the curves fall somewhat with time. IRT patterns within birds are much less marked than they were before the birds were placed on this schedule. The overall rates of the birds were very similar, with $\lambda = .53$, $.51$, and $.59$, slightly higher than the programmed .50. Cumulative records indicated that the birds responded with uniform rates during the daily sessions.

Several general characteristics of this schedule are worth noting. (1) It controls emissive behavior by basing reinforcement on interresponse time, while avoiding the reinforcement of specific IRT ranges either by chance or by design. (2) It uses a feedback principle to concentrate reinforcements where they are needed to produce a preselected behavioral equilibrium. (3) It has thus far produced IRT distributions that correspond rather well to those expected from a response-independent system, if very short IRTs are excepted. However, we cannot conclude that responses are therefore independent. We can be fairly sure only that they are not tied in any simple repetitive pattern. (4) While behavior is variable from moment to moment, the schedule produces stable average rates. Despite their earlier differences on other schedules, the subjects all emitted roughly the same number of responses on a given day and from day to day.

Because λ is fixed, the schedule precludes one of the objectives stated earlier. It does not let rate change freely with any variable in when an experimenter may be interested. However, rate is not entirely controlled by reinforced IRT. Deviations of obtained rate from programmed rate may show the effects of other variables. The next step may be to put the parameter λ itself into the feedback loop that determines reinforcement. If the animal's past average rate were used to set the programmed λ, it's rate might "home" on a value dependent upon other variables. It remains to see whether this will happen.

Figure Captions

Fig. 1 - Inter-response time frequencies of three birds by 0.5 sec., bins, omitting IRTs less than 1.0 sec. long. The preceding response occurred at time "0". The limits of the 16 reinforcement bins are indicated by "bin spacing".

Fig. 2 - The interresponse-times per opportunity (conditional probability of response) for each 0.5 sec. period following a response, calculated from the data in Fig. 1. Values are included for all IRT bins except the rightmost, for which the value is always 1.0.

Footnotes

1. B. F. Skinner, *J. exp. Psychol.*, *38*, 168-172 (1948).

2. This research was supported in part by USPHS grants MH-02456 and MH-08355.

3. W. J. McGill, in *Handbook of Mathematical Psychology*, R. D. Luce, R. R. Bush and W. J. McGill, Eds. (Wiley, New York, 1963), p. 316.

4. The LINC (Laboratory Instrument Computer) is a digital machine with core storage of two thousand 12 bit words, and flexible input, output and magnetic tape capabilities. It was available through an NIH sponsored evaluation program that placed LINCs in a number of bio-medical and psychological applications. The machine is now available commercially from the Digital Equipment Corp., Maynard, Mass.

5. D. S. Blough, in preparation.

6. D. Anger, *J. exp. Psychol.*, *52*, 145-161 (1956).

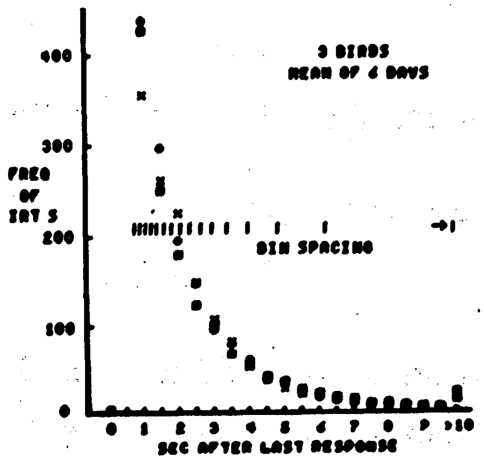

3 BIRDS
MEAN OF 6 DAYS
FREQ OF IRT S
400
300
200
100
0
BIN SPACING
0
1
2
3
4
5
6
7
8
9
>10
SEC AFTER LAST RESPONSE

FIG. 1

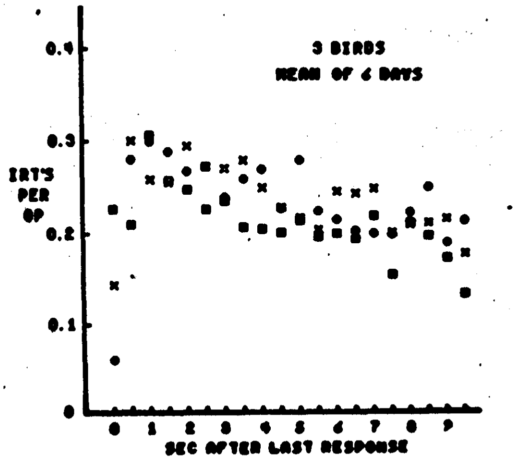

3 BIRDS
MEAN OF 6 DAYS
IRT'S PER OP
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
9
SEC AFTER LAST RESPONSE

FIG. 2

Appendix A - Part 2

Sample of daily data printout for "Least Frequent Interresponse Time" study.

```
date    0201

birds   0007  0593  0812

        0051  0003  0010  0157  0197  0047  0014  0085  0085  0031
IRT     0024  0042  0035  0027  0017  0032  0034  0029  0024  0028
        0649
        0418  0262  0145  0147  0096  0077  0056  0065  0057  0051
raw     0037  0043  0022  0011  0016  0020  0012  0009  0014  0008
        0055
freq.   0000  0000  0000  0000  0000  0000  0000  0000  0003  0000
        0000  0000  0001  0000  0000  0000  0000  0000  0000  0000
in      0040
        0000  0003  0001  0000  0001  0002  0001  0003  0004  0003
0.1     0004  0001  0001  0004  0003  0001  0000  0000  0000  0001
        0011
and     0071  0033  0017  0125  0078  0016  0032  0053  0024  0033
        0063  0041  0038  0047  0042  0060  0028  0031  0029  0023
0.5     0697
        0324  0158  0231  0171  0085  0095  0089  0091  0070  0061
sec     0034  0019  0020  0016  0020  0014  0013  0010  0015  0010
        0035
bins    0000  0000  0000  0000  0000  0000  0000  0000  0000  0000
        0000  0000  0000  0000  0000  0000  0000  0000  0000  0000
        0036
        0000  0000  0000  0000  0000  0000  0000  0000  0002  0001
        0001  0003  0003  0001  0001  0001  0000  0003  0003  0000
        0017
        0000  0000  0001  0018  0014  0014  0034  0038  0091  0078
        0033  0042  0046  0033  0028  0034  0028  0029  0026  0014
        0724
        0033  0255  0182  0131  0077  0069  0089  0081  0064  0050
        0057  0041  0032  0032  0028  0018  0016  0016  0010  0005
        0039
        0000  0000  0000  0000  0000  0000  0000  0000  0000  0000
        0000  0001  0002  0000  0000  0000  0000  0000  0000  0000
        0024
        0000  0000  0003  0000  0003  0000  0000  0000  0000  0000
        0000  0001  0000  0000  0000  0001  0000  0001  0004  0000
        0014

IRT     0129  0102  0075  0087  0080  0062  0062  0061  0066  0070
freq.   0071  0062  0043  0044  0047  0040
by      0084  0126  0145  0092  0070  0055  0080  0086  0104  0069
special 0075  0043  0042  0047  0045  0029
bins    0187  0113  0091  0082  0069  0054  0057  0085  0075  0075
        0082  0065  0074  0057  0050  0017
```

page 2

IRTs/OP

histograms,

3 birds by 0.1

sec. bins

```
         0    .    1    .    2    .    3    .    4    .    5

  1      XXX
  2      0
  3      0
  4      XXXXXXXXXX
  5      XXXXXXXXXXXXXX
  6      XXX
  7      X
  8      XXXXXXX
  9      XXXXXXXX
 10      XXX
 11      XX
 12      XXXX
 13      XXX
 14      XXX
 15      XX
 16      XXXX
 17      XXXX
 18      XXX
 19      XXX
 20      XXXX
 21      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
         0    .    1    .    2    .    3    .    4    .    5

  1      XXXX
  2      XX
  3      X
  4      XXXXXXXX
  5      XXXXX
  6      X
  7      XX
  8      XXXX
  9      XX
 10      XX
 11      XXXXX
 12      XXX
 13      XXX
 14      XXXX
 15      XXXX
 16      XXXXXX
 17      XXX
 18      XXX
 19      XXX
 20      XXX
 21      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
         0    .    1    .    2    .    3    .    4    .    5

  1      0
  2      0
  3      0
  4      X
  5      X
  6      X
  7      XX
  8      XXX
  9      XXXXXXX
 10      XXXXXX
 11      XXX
 12      XXXX
 13      XXXX
 14      XXX
 15      XXX
 16      XXX
 17      XXX
 18      XXX
 19      XXX
 20      X
 21      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

page 3

IRTs/OP

histograms,

3 birds by 0.5

sec. bins

```
        0    .    1    .    2    .    3    .    4    .    5

  5      XXXXXXXXXXXXXXXXXXXXXXXXX
 10      XXXXXXXXXXXXXXXXXXXXX
 15      XXXXXXXXXXXXXXX
 20      XXXXXXXXXXXXXXXXXX
 25      XXXXXXXXXXXXXX
 30      XXXXXXXXXXXXX
 35      XXXXXXXXXXX
 40      XXXXXXXXXXXXXXX
 45      XXXXXXXXXXXXXXXX
 50      XXXXXXXXXXXXXXXXX
 55      XXXXXXXXXXXXXX
 60      XXXXXXXXXXXXXXXXXXXX
 65      XXXXXXXXXXXXX
 70      XXXXXXX
 75      XXXXXXXXXXX
 80      XXXXXXXXXXXXXXXX
 85      XXXXXXXXXXXX
 90      XXXXXXXXXX
 95      XXXXXXXXXXXXXXXXXX
100      XXXXXXXXXXXX
105      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


        0    .    1    .    2    .    3    .    4    .    5

  5      XXXXXXXXXXXXXXXXXXXX
 10      XXXXXXXXXXXX
 15      XXXXXXXXXXXXXXXXXXXXX
 20      XXXXXXXXXXXXXXXXXXX
 25      XXXXXXXXXXXX
 30      XXXXXXXXXXXXXXX
 35      XXXXXXXXXXXXXXXXX
 40      XXXXXXXXXXXXXXXXXXXXX
 45      XXXXXXXXXXXXXXXXXXXX
 50      XXXXXXXXXXXXXXXXXXXXXX
 55      XXXXXXXXXXXXXXXX
 60      XXXXXXXXXXX
 65      XXXXXXXXXXXXX
 70      XXXXXXXXXXXX
 75      XXXXXXXXXXXXXXXXX
 80      XXXXXXXXXXXXXX
 85      XXXXXXXXXXXXXXX
 90      XXXXXXXXXXXXXX
 95      XXXXXXXXXXXXXXXXXXXXXXXXX
100      XXXXXXXXXXXXXXXXXXXXXX
105      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


        0    .    1    .    2    .    3    .    4    .    5

  5      XX
 10      XXXXXXXXXXXXXXXXXXX
 15      XXXXXXXXXXXXXXXXX
 20      XXXXXXXXXXXXXXX
 25      XXXXXXXXXX
 30      XXXXXXXXXX
 35      XXXXXXXXXXXXXXX
 40      XXXXXXXXXXXXXXXX
 45      XXXXXXXXXXXXXXX
 50      XXXXXXXXXXXXXX
 55      XXXXXXXXXXXXXXXXXXX
 60      XXXXXXXXXXXXXXXXX
 65      XXXXXXXXXXXXXXXX
 70      XXXXXXXXXXXXXXXXXXX
 75      XXXXXXXXXXXXXXXXXXXXX
 80      XXXXXXXXXXXXXXXXX
 85      XXXXXXXXXXXXXXXXXX
 90      XXXXXXXXXXXXXXXXXXXXXX
 95      XXXXXXXXXXXXXXXXXX
100      XXXXXXXXXXX
105      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

1545 0502 0438 (coded total pause time, 3 birds)

MONITOR PROGRAM FOR THE LINC

Lloyd Marlowe and Donald S. Blough
Psychology Department
Brown University

The MONITOR program is an elaboration of a program called META, which was used with the 1024 word memory to read in and start other programs. With the addition of the second 1024 words of core memory, the MONITOR program was written so that up to 210 programs could be read in and started without any operator intervention, by using a predetermined MONITOR control sequence. However, the program layout and the manner in which the programs are read in and started was not changed to take advantage of the full core memory. Consequently, the reader will probably notice numerous cases of inefficient program layout and storage. In addition to reading in and starting programs the MONITOR program presents a display that is associated with each program so that values for variables in the program can be conveniently typed in before the program is executed.

FORMAT OF MONITOR CONTROLLED PROGRAMS

Any program which is going to be used under MONITOR control must be written using the format described below. Each program has three basic parts: a display, a control block, and a program proper.

Display

A sample display is shown below.

```
         EXPERIMENT #1A
    STORE DATA BEGINNING AT
          REGISTER _ _ _
             BLOCK _ _ _
              UNIT _
```

The program display is made using the DISPLAY MAKER program (see Appendix A). The display produced by the DISPLAY MAKER will occupy 2N, (N = any integer greater than 0), consecutive blocks on either unit 0 or 1.

After the display and the remainder of the program are loaded into the MONITOR system using the PROGRAM RELOCATION program (see Appendix B), the 0 register of block 2N-1 of display will contain the block number of the control block in bits 0-8; it will contain the unit number in bit 11.

The numbers that are typed in are treated as decimal numbers - i.e. digits 8 and 9 may be used. This may cause some initial confusion in referring to block and register numbers, which must be converted to decimal numbers before they can be typed in.

Control Block

The control block (1 block on tape), and the program proper (any number of blocks), are stored as a unit in consecutive blocks on either tape unit without regard to the storage location of the display. MONITOR assumes that the first block of this unit is the control block. Thus the control block and program proper always have the same relative positions, no matter where they are stored.

When it comes time to execute the program proper, MONITOR does the following:

1) The contents of the 0 register of block 2N-1 of display are transferred to register 3400 in core memory.

2) The list of numbers typed into the program display is stored sequentially beginning at register 3401.

3) The control block is read into Q_3 (registers 1400-1777).

4) MONITOR executes JMP 1400 to begin execution of the control block.

Next the control block using the number in register 3400 calculates the locations of the blocks containing the program proper. The program proper is then read into any quarters other than Q_3 and Q_7. If the program proper requires all of the first half of memory, Q_3 will have to be read in later by the program proper itself. Next the type-ins stored at 3401 are stored in the appropriate registers of the program. Of course, any desired manipulations may be made on the type-ins in this process. Finally the control block executes a JMP X instruction to begin execution of the program proper at register X.

Program Proper

The program proper consists of what is usually called the program. However, instead of halting when the last instruction is executed, the program must execute this sequence of instructions:

```
RDC i
0/400
JMP 20
```

These instructions read the MONITOR program back in and restart it. This sequence of instructions must not be in Q_0.

A sample program using this format is given in Appendix C.

FACILITIES USED BY MONITOR

The basic MONITOR program occupies blocks 252-271. Storage for

MONITOR control sequences is laid out in multiples of 8 blocks per group beginning with block 272. Up to 30 groups can be stored. Thus everything after block 252 should be left open for MONITOR use. In each group of 8 blocks the first block is left blank. Each of the remaining 7 blocks contain one list of type-ins. Therefore a list for a program that is stored in a monitor control sequence can be referred to by block number (1-7) and group number (1-30). (All numbers in this paragraph are decimal.)

While being run, the MONITOR program uses all 8 blocks of core memory.

MONITOR uses sense switches 4 and 5.

The Soroban keyboard must be used in filling in displays.

GENERAL OPERATING INSTRUCTIONS

To start MONITOR, put the MONITOR tape on unit 0; turn on sense switch 5; execute RDC, 0/400; start 20; and turn off sense switch 5.

Displays presented by MONITOR are completed as follows:

1) To replace a blank with a character, simply strike the appropriate key on the key board. Upper case characters can not be used.

2) After a group of blanks have been filled, MONITOR ignores the keyboard until EOL or DEL is struck.

3) EOL advances MONITOR to the next set of blanks.

4) DEL replaces the last group of blanks, that was replaced with characters, with blanks.

5) If EOL is struck three times, the next page of display is presented. After the last page is presented, the MONITOR program continues.

6) If CASE is struck, all the blanks are reset and the COMMAND display (to be explained later) is presented.

7) Although MONITOR does some very simple error checking, one should be very careful in completing the displays since some errors can damage MONITOR. If an error is detected the COMMAND display is presented.

Before removing the MONITOR tape from the machine or before using a program that is not under MONITOR control, one should be sure that the COMMAND display is being presented.

MONITOR COMMANDS

After MONITOR is started the COMMAND display shown below will be presented.

EXECUTE COMMAND NUMBER _

1 EXECUTE 1 PROGRAM

2 CLEAR MONITOR
3 GET, FILL, STORE
4 DELETE
5 REPLACE
6 STORE MONITOR
7 ADD MONITOR

8 EXECUTE MONITOR
9 CONTINUE MONITOR

The nine commands in the display can be grouped into three functional groups: execution of one program (1), assembly of MONITOR control sequence (2-7), and execution of MONITOR control sequence (8,9).

Execution of One Program

1 EXECUTE 1 PROGRAM

When this command is selected, the following display is presented:

DISPLAY STORED AT
BLOCK _ _ _ UNIT _

The location of the first block of display for the program to be executed is typed into the display. The program display is then presented and filled in. The program is immediately executed and another command may then be selected.

Assembly of MONITOR Control Sequence

2 CLEAR MONITOR

This command effectively erases the MONITOR control sequence currently in storage.

3 GET, FILL, STORE

This command is similar to the EXECUTE 1 PROGRAM command. However, instead of executing the program, after the program display is completed, the values typed into the display and the location of the program control block are stored in the next available MONITOR storage block.

If DIS,0 is typed into the display, which asks for the display location, the following display is presented:

```
MONITOR CONTROLLED
DISPLAY STORED AT
BLOCK _ _ _ UNIT _
```

In this case the program display location is stored in the next MONITOR storage block, instead of a list of values.

4 DELETE

When this command is selected, the following display is presented:

```
DELETE LAST _ _ _ BLOCKS
OF MONITOR VALUES
```

If the number N is typed into the display, the N blocks last stored in MONITOR storage are effectively erased.

This command also copies the contents of blocks 268-271 into blocks 252-255.

5 REPLACE

This command is the same as the GET, FILL, STORE command except that the following display is used:

```
DISPLAY STORED AT
BLOCK _ _ _ UNIT _
STORE VALUES AT
BLOCK _ GROUP _ _
```

This makes it possible to store the values typed in (or the display location) in any of the MONITOR storage blocks, instead of the next location that is available. This, of course, erases whatever was in that storage location.

6 STORE MONITOR

When this command is selected, the following display is presented:

```
STORE XX GROUPS
OF MONITOR VALUES
AT BLOCK _ _ _ UNIT _
```

The number of groups to be stored is shown instead of XX. The block number typed in must be a multiple of 8. Each group occupies 8 blocks.

7 ADD MONITOR

When this command is selected, the following display is presented:

ROOM FOR YY GROUPS
OF MONITOR VALUES
ADD _ _ GROUPS OF
MONITOR VALUES FROM
BLOCK _ _ _ UNIT _

The sum of XX and YY is always 30. The number of groups asked for must not exceed the number YY. <u>The block number must be a multiple of 8</u>.

When a MONITOR control sequence is added to another control sequence, any empty blocks in the last group of the first sequence are left empty. These blocks are marked as being empty with a -1 in the 0 register. However, if GET, FILL, STORE is executed after adding a control sequence that has empty blocks at the end of the last group, the first empty block will be used to store the list of type-ins.

<u>Execution of MONITOR Control Sequence</u>

8 EXECUTE MONITOR

When this command is selected, MONITOR begins executing programs beginning with the list of values or display location that is in the first MONITOR storage block.

If the contents of the storage block is a list of values, the program is executed using these values.

If the contents of the storage block is a display location, the display is presented. After the display is completed, its program is executed using the values just typed in. If case is struck, instead, this program is skipped, and an interruption is signaled.

An interruption can also be made by turning on Sense Switch 5. When an interruption is made, the program that is currently being worked on is completed, the COMMAND display is presented, and commands may be executed. During the first type of interruption, Sense Switch 5 should be left on until a #1 command is completed or until a #8 or 9 command is started.

The MONITOR program continues executing programs until the program called for in the last storage block used is executed, or until an interruption is made.

If Sense Switch 4 is turned on, the contents of Q_{4-7} are stored in blocks 252-255 after each execution of a program. Thus, the contents of Q_{4-7} after each execution of a program can be viewed by using Sense Switch 5 to interrupt MONITOR and then using COMMAND #1 to use a Decimal or Octal display program to display blocks 252-255. Before the next program is executed, blocks 252-255 will be copied into blocks 268-271.

Note that if a deletion is made at this point, blocks 268-271, the contents of Q_{4-7} after the execution of the next to last program, are copied into blocks 252-255 (see 4 DELETE).

9 CONTINUE MONITOR

If an interruption is made using Sense Switch 5, execution of the MONITOR control sequence may be resumed at the point where it was interrupted if, and only if, the EXECUTE MONITOR or CLEAR MONITOR commands are not used. This means that during an interruption the MONITOR control sequence may be altered using commands GET, FILL, STORE, REPLACE, DELETE, and ADD MONITOR. Also the tapes may be removed and other tapes may be worked on. If the MONITOR program has been left, read MONITOR in using the usual read in sequence. After the CONTINUE MONITOR command has been selected, the following display will be presented:

```
CONTINUE MONITOR

FINAL PROGRAM AT
BLOCK W GROUP XX

TO START EXECUTION
AT BLOCK Y GROUP ZZ
STRIKE C AND 3 EOLS

        OR

START EXECUTION AT
BLOCK _ GROUP _ _
```

This display tells the location of the last block of values stored and the location of the block of values that would have been executed next if an interruption had not occurred. To resume execution with that block of values, strike C and 3 EOLS. If it is necessary to restart the MONITOR execution at a different storage location, fill in the display with the correct block and group number.

APPENDIX A - DISPLAY MAKER

The DISPLAY MAKER program occupies blocks 247-251 on a MONITOR tape. The first block of its program display is stored at 247.

This program is used to make displays for programs under MONITOR control. After calling this program in with MONITOR command 1, the following display is presented.

DISPLAY MAKER PROGRAM

START DISPLAY AT
BLOCK _ _ _ UNIT _

The number of blocks stored will be presented when execution of this program is completed. After this display is completed, tape unit 0 will churn for a bit. When tape unit 0 stops, a display can be created.

A display has 2 basic components. First, there are statements, (e.g., "EXPERIMENT #A"). Second, there are groups of blanks, (e.g., "_ _ _"). When the display is presented under MONITOR control and characters are typed into the display, the statements will not be altered; the blanks will be altered.

A display is laid out in a matrix 25 letters wide and 16 lines high. The letters and lines are referred to by using the appropriate keyboard characters (see Table 1). By usual convention, X will refer to the horizontal axis (letters) and Y will refer to the vertical axis (lines).

A display is typed in using the commands described below. To begin a command, strike CASE twice and then strike the letter for the command.

C STATEMENT

After typing CASE,CASE,C (C stands for character), type the Y (line #) and X (letter #) values for the first character in the statement followed by the desired sequence of characters. Upper case characters can be used.

To delete a character, strike DEL once. A C,Y,X or B,X, (see below), are stored as 3 and 2 characters, respectively. If any part of a C,Y,X, or B,X has to be deleted, all of it must be deleted. CAUTION: Do not delete more characters than have been typed on the current page of display since this will delete characters on the previous page. The accumulator lights are steady when everything on a page has been deleted.

B GROUPS OF BLANKS

After typing CASE,CASE,B, type the X value for the first blank. The Y value is the same as the Y value typed in on the last C command. Blanks are typed in using upper case 5. Any other symbol may also be used, in which case it is treated as if it were a blank when characters are typed into the display when the display is presented under MONITOR control. (see command C for deletions.)

L LETTERS REMAINING

If this command is executed the number of characters remaining, before the display must be stored (S) or ended (E), is presented. To resume, strike EOL once.

P NEW PAGE

A new page of display can be started by executing this command. The first thing typed on a page must be a C,Y,X. CAUTION: Each page after the first page must have at least one B,X on it.

S STORE DISPLAY

When this command is executed 2 blocks of display codes are stored and storage for another 1011 letters is made available.

E END DISPLAY

When this command is executed, the final 2 blocks of display codes are stored and the total number of blocks stored is presented. The number presented is the number that must be used in the PROGRAM RELOCATION program display. To return to MONITOR, strike EOL 3 times.

TABLE 1 - CHARACTERS CORRESPONDING TO LETTER AND LINE NUMBERS

0	0	5	5	10	EOL	15	-	20	A
1	1	6	6	11	DEL	16	+	21	B
2	2	7	7	12	SPACE	17	/	22	C
3	3	8	8	13	i	18	#	23	D
4	4	9	9	14	p	19	CASE	24	E

APPENDIX B - PROGRAM RELOCATION

The PROGRAM RELOCATION program occupies blocks 245 and 246 on a MONITOR tape. The first block of display is stored at block 245. Note: The word program, when used in reference to the program being relocated, does not include the program display.

This program must be used when ever a MONITOR controlled program is copied from one location on tape to another location on tape, with one exception. After a program is converted to binary form from manuscript form, it is necessary to put any programming in Q0 through Q2 along with Q3 in consecutive blocks with Q3 (the control block) in the first block. This rearranging can be done with either the LAP octal copying program or the MONITOR decimal copying program. After the rearranging is completed the RELOCATION PROGRAM must be used.

As is pointed out in the RELOCATION PROGRAM display, a program must have a display before it can be relocated. This is necessary since the RELOCATION PROGRAM stores the location of the first block of the program (assumed to be the control block) in the 0 register of the next to last block of the display.

There are 2 displays in this program. The first page of the first display is a reminder. The second page is used to copy the program from its old location to its new location. The second display asks for the number of blocks in the program's display and for the location of the first block of the display. The number of blocks of display must be the same as the number presented at the end of the DISPLAY MAKER program when the program display was made. The tape on unit 1 can be changed before the second display is completed if the program display and the program are on different unit 1 tapes.

COPY PROGRAM

The COPY PROGRAM occupies blocks 243 and 244 on a MONITOR tape. The first block of display is stored at block 243.

This program is used to copy blocks of information from one location to another. For example, it can be used for putting the control block and program proper in the correct sequence before using the PROGRAM RELOCATION program and for relocating program displays.

CAUTION: Do not try to copy 0 number of blocks. Remember that block numbers must be typed-in in decimal form.

APPENDIX C - ILLUSTRATION OF MONITOR PROGRAM FORMAT

Some of the basic steps and details involved in writing a MONITOR controlled program are illustrated below in a simple program for making a frequency distribution of a list of fixed point numbers.

Since this program will be used in a number of different situations it is necessary to be able to change 1) the number of intervals on the abscissa, 2) the value of the left boundary of the left most interval, 3) the interval width, 4) the number of numbers in the list, 5) the number of consecutive lists, 6) the input block and the list storage location within it, and 7) the output block and the distribution storage location within it. These parameters can be changed by using the following program display.

```
FREQUENCY DISTRIBUTION

# OF INTERVALS _ _ _
LEFT BOUNDARY _ _ _ _
INTERVAL WIDTH _ _ _ _

ITEMS PER LIST _ _ _
# OF LISTS _ _ _

INPUT - REGISTER _ _ _
BLOCK _ _ _ UNIT _

OUTPUT - REGISTER _ _ _
BLOCK _ _ _ UNIT _
```

To make this display, first, read in the MONITOR COMMAND display, and then, execute the DISPLAY MAKER program (see Appendix A) using the following sequences of key strikes.

1,EOL,EOL,EOL

2,4,7,EOL,0,EOL,EOL,EOL

2,0,0,EOL,0,EOL,EOL,EOL(this will start the display storage at block 200 decimal on unit 0)

CASE,CASE,C,2(line number),2(letter number),F,R,E,Q,U,E,N,C,Y,SPACE, D,I,S,T,R,I,B,U,T,I,O,N,

CASE,CASE,C,4,3,#,SPACE,O,F,SPACE,I,N,T,E,R,V,A,L,S,CASE,CASE,B,#(letter number),CASE,5(blank),CASE,5,CASE,5

CASE,CASE,C,5,3,L,E,F,T,SPACE,B,O,U,N,D,A,R,Y,CASE,CASE,B,/,CASE,5,CASE,5, CASE,5,CASE,5

CASE,CASE,C,6,3,I,N,T,E,R,V,A,L,SPACE,W,I,D,T,H,CASE,CASE,B,#,CASE,5,CASE, 5,CASE,5,CASE,5

CASE,CASE,C,8,3,I,T,E,M,S,SPACE,P,E,R,SPACE,L,I,S,T,CASE,CASE,B,#,CASE,5, CASE,5,CASE,5,

CASE,CASE,C,9,3,#,SPACE,O,F,SPACE,L,I,S,T,S,CASE,CASE,B,p,CASE,5,CASE,5, CASE,5

CASE,CASE,C,DEL,3,I,N,P,U,T,SPACE,-,SPACE,R,E,G,I,S,T,E,R,CASE,CASE,B,A, CASE,5,CASE,5,CASE,5

CASE,CASE,C,SPACE,3,B,L,O,C,K,CASE,CASE,B,9,CASE,5,CASE,5,CASE,5,CASE, CASE,C,SPACE,i,U,N,I,T,CASE,CASE,B,#,CASE,5

CASE,CASE,C,p,3,O,U,T,P,U,T,SPACE,-,SPACE,R,E,G,I,S,T,E,R,CASE,CASE,B,B, CASE,5,CASE,5,CASE,5,

CASE,CASE,C,-,3,B,L,O,C,K,CASE,CASE,B,9,CASE,5,CASE,5,CASE,5,CASE,CASE,C, -,i,U,N,I,T,CASE,CASE,B,#,CASE,5

CASE,CASE,E,(this ends display and stores it and presents the number of blocks stored)

EOL,EOL,EOL(this returns you to the COMMAND display)

To view the display, execute either command #1 or command #3. After viewing the display, strike CASE, not 3 EOLs. This will return MONITOR to the COMMAND display.

The next step is to type in the program proper and the control block (see end of this appendix) using LAP. The programming of the actual problem will not be discussed since it is assumed that the reader knows how to program and since the program is quite simple. Instead, the features introduced by the MONITOR format will be pointed out.

1) The program proper and the control block do two very different things. The program proper deals with the making of the frequency distribution. The control block fills in the specific values of the different parameters in the program proper.

2) After the control block is read in and started at register 1400 it does the following:

   a) Register 3400 is used to locate and read in the program proper which is stored in the block following the control block.

   b) The values stored at 3401 and following and are loaded one by one into the program proper. As the values are loaded, they are altered so as to allow for more efficient running in the program proper.

c) Finally the control block executes JMP 1A to start execution of program proper.

3) When the program proper is completed

```
RDC i
0/400
JMP 20
```

is executed to read MONITOR back in.

After this program is converted to binary form, it will be in the LAP conversion blocks 216 (program proper) and 219 (control block). At this point put the LAP tape on unit 1 and put the MONITOR tape on unit 0. Suppose one wanted to store the control block and program proper starting at block 225. In this case, the MONITOR COPY PROGRAM (see Appendix B) would be used to copy block 216,1 to block 226,0 and the PROGRAM RELOCATION program (see Appendix B), would be used to copy block 219,1 to 225,0. After copying the control block, the PROGRAM RELOCATION program will ask for the number of blocks of program display and its location. In this illustration, the display required 2 blocks and it was stored at 200,0. At the completion of this rearrangement, the control block will be at 225,0; the program proper will be at 226,0; the display will be in 200,0 and 201,0; and the location of the control block will be in register 0 of block 200,0. At this point, the FREQUENCY DISTRIBUTION program can be run under MONITOR control. (All block numbers in this paragraph are decimal.)

This program has been laid out in a general format. However, since it is so short and since the program display only fills one of the two blocks that was stored, some minor alterations can be made to eliminate two blocks of tape storage. First, the program proper can be put in Q3 at the end of the control block. This eliminates one block. Second, this combined block can be stored on top of the second block of display. If this is done the display must be made and stored before the control block and program proper are stored, since the DISPLAY MAKER stores two blocks of codes no matter what. Then the PROGRAM RELOCATION program can be used to store the control block and program proper in the second block of display. When the PROGRAM RELOCATION program asks for the number of blocks of display, the number 2 must be typed in even though the second block of display is filled with programming. Undoubtedly, the user will frequently find it useful to partially or completely intermesh the control block, the program proper, and the display in order to save tape storage.




An Experimental Investigation of the Dolphin's Communication Abilities by Means of a Dolphin Machine Code

John C. Lilly, M.D., Director, Communication Research Institute,
Miami, Florida

It has been found desirable to establish a program for a computer or several programs for a computer which are capable of teaching a dolphin (or a human) a simple five element code. This code is first taught with a basic teaching program; it is then used in a second program which connects the code with objects, actions and finally with other symbols; a third program tests the dolphin's ability to solve problems using the newly developed "primitive language" from the code. This report is an account of the development of the elements for a first code, the necessary input and output devices to allow synthesis and analysis of code elements, the setting up a LINC computer as a prototype of a larger machine to handle these problems, and a special facility that will handle either a dolphin or a human subject in a series of tests.

The first code which we are proposing using is based on the separate elements of the dolphin's natural sonic and ultrasonic outputs. The species which we use (*Tursiops truncatus*) has three separate emitters for sonic and ultrasonic productions. Two of these are in and below the blowhole in the nasal passageways, they are separately controllable on the right and the left sides. The third source of sounds and ultrasounds is the larynx system in the midplane. We have extensive evidence that these three are totally independently controllable by the dolphin; he can emit sounds from each of these out of synchrony or in synchrony with the other two. (*Vocal behavior of the bottlenose dolphin* by J. C. Lilly, Proc. Am. Philos. Soc. 106: 520-529, 1962.)

The symbols chosen to represent simulations of portions of the sounds that the dolphin can produce are as follows:

"W" is used to symbolize a dolphin's whistle. "C" is used to symbolize a click train. "H" is used to symbolize an airborne emission. "U" is used to symbolize an ultrasonic emission by the dolphin. The simulations themselves of these four classes will be dealt with in greater detail below.

The experimental configuration is as follows:

(1) The dolphin is in a small tank with three transmitters and two receivers. One receiver is an underwater hydrophone; the other is an air-mounted microphone above the blowhole. The three underwater transmitters have the following functions:

Transmitter #1 sends code elements and combinations of code elements to the dolphin from the computer. The <u>first</u> teaching program says that if the computer emits, for example, a "WCU" combination and if the dolphin mimics the "WCU" combination, then the computer goes on and transmitter #1 gives him a new element say "W" or "CW" or some other combination.

Transmitter #2 is programmed in such a way that if the dolphin does not mimic transmitter #1, transmitter #2 mimics the dolphin with the code elements which approximate most closely that which the dolphin has emitted.

If the dolphin mimics five simulations in a row, five emissions by transmitter #1, he is given a reward.

Transmitter #3 is used to create the setting under which the rest of the program is given to the dolphin. For example, this transmitter emits a sequence of sounds (not necessarily in the code elements) which give the message in effect "program ready to be run - prepare yourself to work with program"; the message could simply be a spoken word like "ready." Other kinds of messages such as "wrong", "right", "start over", and so forth are emitted by transmitter #3. This transmitter is controlled by the computer and is fed, for example, from a magnetic drum with several heads mounted on the drum. Messages are recorded underneath the separate heads. The computer can choose the particular head to be hooked in to give that particular message in English to the dolphin or to a human subject. Alternatively other forms of storage could be used. Of necessity this transmitter uses a very limited vocabulary. The dolphin can be trained to this vocabulary (as we have already shown) by a human operator in other experiments. We have found that a dolphin can distinguish among a very large number of human emissions. Of course it is possible to substitute, for example, a row of colored lights or distinctive sounds instead of these human emissions.

An initial simple setup will probably be used; we will use a series of signal lights instead of the human voice.

The position in the sequencing of the reward may require a demonstration of a reward or the giving of a reward immediately after announcing the program, to show the subject that this aspect of machine operation also exists. In other words the existence theorem must be satisfied not only for every element of the code, every combination of elements of the code, but for all of the elements of the teaching program which are at the machine-subject interface.

The detailed treatment of the code itself in the learning procedure can be summarized as follows. The subject is presented with the elements of the code separately and distinctly. He is then presented with pairs of elements of the code sequentially giving all possible permutations and combinations of pairs. He is then presented with pairs simultaneously, i.e., two elements of the code are given concurrently and thus one works up through dyadic, triadic, and so forth groups elements.

Depending on the power of the subject to abstract and project ahead and extrapolate from the data up to this point, one can now start pairing combinations of the elements of the code with other sonic events, or objects, or actions, or printed symbols, and so forth. Such pairings now use the elements of the code and combinations of the elements of the code as symbols for the other symbols, actions, etc. in other modes of stimulation or in the sonic mode itself. Combinations of these newly assigned "meanings" for code elements and combinations of code elements can be presented in combinations with the combinations of those things etc., which they symbolize.

As a beginning set of code elements to be tested, we have chosen the following elements and the following symbols for each of these elements, the written symbols for each of the elements as follows:

"W" is the symbol for an element taken from the dolphin's own sonic exchanges; in this case it is a sine-wave of 3 tenths of a second duration either of constant frequency at 10 kc or frequency-modulated from 8 kc to 20 kc with a linear rise in frequency with time. This is an example of a dolphin "whistle."

The "C" is used to symbolize a train of dolphin clicks. The synthetic element is a train lasting ½ second at a click rate of 40 clicks per second. It is created by ringing a series of filters whose center frequencies are in the same band as that swept in the whistle, i.e. from about 8 kc to about 20 kc. The "Q" of these filters is chosen so that the ringing lasts no longer than approximately 4 milliseconds.

The "U" symbolizes the ultrasonic signals of a dolphin and consists of a series of clicks lasting .3 seconds emitted at 100 per second, and ringing for a filter centered at 40 kc.

The "H" symbolizes the airborne sonic emissions of a dolphin who has been in chronic contact with speaking humans. Such a dolphin emits a sound which bears some resemblance to the high pitched portions of the sounds emitted by humans. He does it in air. The synthetic element "H" is a series of filters which are rung in the frequency region between 1,000 cps and 4,000 cps with a click train at 125 per second lasting 0.5 seconds.

There is a special symbol called "N" which symbolizes another set of noises that dolphins tend to emit which are undesirable for use in this program. "N" symbolizes water noise produced by movements of the animal's body. Water noise can exist underwater and interfere with the pickup to the computer underwater or can be picked up in air and interfere with the microphone pickup. It has been found emperically that if we look at a band of sound below 100 cycles that one can detect such interferring signals rather easily. Therefore an acoustic analyzer channel hooked to the hydrophone and to the microphone channels has the property of telling the computer to send out a signal over transmitter #3 to let the dolphin know that this is not an acceptable tactic. In other words, transmitter #3 can put out a very loud unpleasant noise or some symbol meaning "stop moving." We have found that it is very easy to shape up this kind of behavior in a cooperating dolphin; he will attempt to minimize water noise when we can transmit this information to him successfully.

For a signal sent back by the dolphin accompanied by a water noise, the program causes this negation signal to appear on transmitter #3.

Permutations and combinations of "W C H and U" are given in Figure 2. Since a dolphin can give any three of these elements simultaneously we have given both the sequential and the simultaneous permutations and combinations. In other words, a dolphin can give two separate whistles plus an ultrasonic signal simultaneously, can give a whistle and a clicking simultaneously with an ultrasonic signal. Or he can give any one of these or in pairs, in air or underwater.

For example, if the dolphin wishes, he can whistle with his left nasal phonation apparatus and can click with his right phonation apparatus at the same time that he is emitting either an ultrasonic whistle or a series of clicks with his laryngeal apparatus. Thus, the elements of the code can be either simultaneous or sequential. For example, the combination taken from the table "W C U" can be a simultaneous whistling, clicking and ultrasonic signal or it can be a whistle immediately followed by clicking, immediately followed by an ultrasonic signal. This means that there are several hundreds of combinations which can be worked out with this simple set of code elements. The "H" sound is a special case of "W" or "C". A dolphin can emit an "H" plus a "W" or a "C" or he can emit two "H's". In other words, the "H" airborne sounds are the equivalent of the underwater "W" and "C" sounds. However, a dolphin can also do a "W" in air or a "C" in air in the other frequency region. "H" being the much lower frequency region which he uses only under special conditions. Therefore the interface between the dolphin and the computer must contain analyzers which not only analyze underwater signals but analyze the airborne signals. Thus, we will have a very large "vocabulary" of combinations which can be used in setting up symbols for actions, objects, printed symbols, etc.

The human subject is quite capable of doing several things simultaneously in the same way that the dolphin is. Several things can be used as the analogs of these above elements for the dolphin and the correct regions in the frequency and repetition rate domains chosen for the human. The final catalog of the elements which can be made into combinations then are as follows:

Underwater or airborne whistles - two simultaneously or one paired with a "C" or an "H"; airborne clicks "C", two or one paired with a "W" or an "H"; airborne or underwater "H" two or one paired with a "C" or a "W". The dolphin's emissions are separated out by analog analyzers into six inputs to the computer. "H" is selected by a high pass filter or by means of a radio receiver tuneable over the band from 30 kc to 300 kc. We have found that "H" can be separated from "W" and "C" by a simple low pass filter. "W" and "C" are separated by use of well known carrier detection methods. In other words, "W" can be suppressed by a properly designed rectification and filtering system. "C" can be suppressed by the usual methods for suppression of high speed transients. Since "C" does not last more than 4 milliseconds, this circuitry selects that which lasts longer than 4 milliseconds, i.e., rejecting the short term signals. With the proper design using counters, rectifiers and filters, we have constructed satisfactory discriminating analyzers for these four elements.

The first of the computer programs is a teaching program, i.e., the elements themselves and combinations of these elements are fed to the animal and he is asked to mimic them. In the past we have established mimicking routines with a human operator and dolphin with a great deal of success using up to ten elements given at a rate of one every .7 seconds.

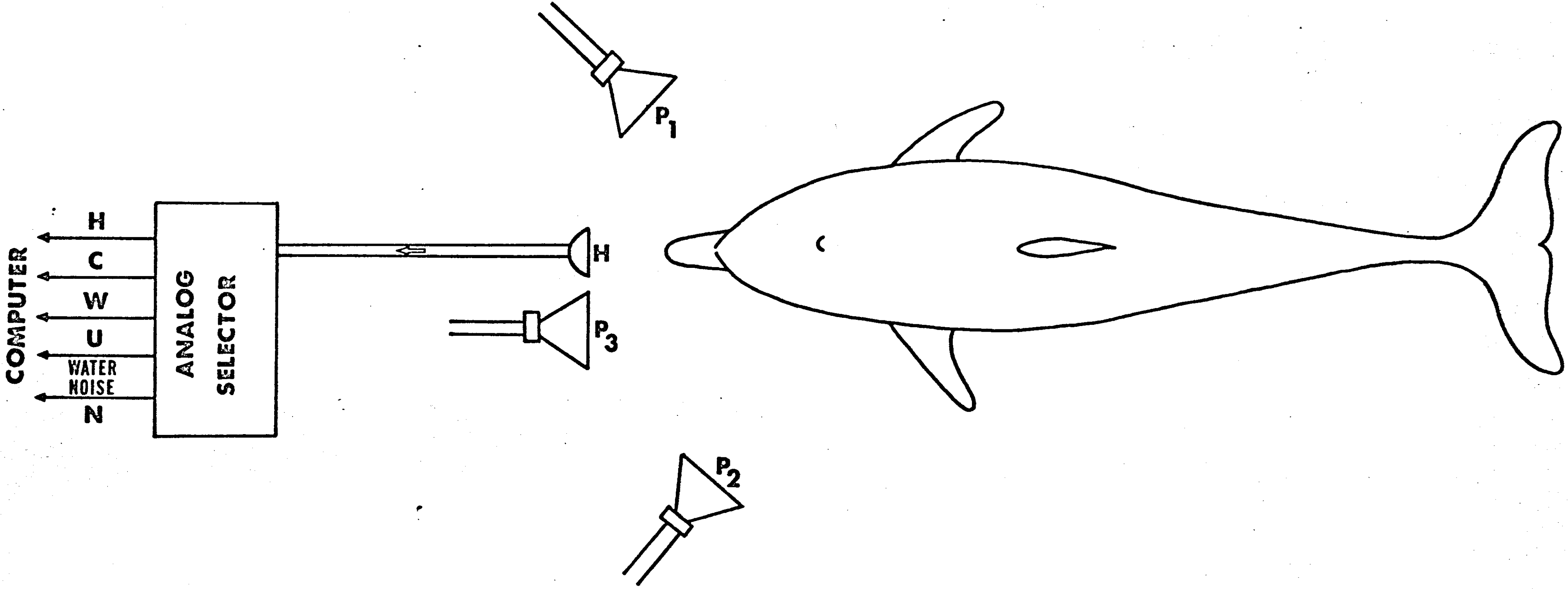
The mimicking routine is the one given above in which transmitter #1 gives that which is to be mimicked, transmitter #2 mimics the animal if he does not give back what is expected and #3 corrects if there is any water noise interferring with the exchange.

The second major program pairs up these elements and combinations of these elements with objects, actions and other kinds of symbols. Material is projected with a rotary slide projector so that a picture is projected on a translucent screen which pairs up a picture (of the object, action, or person or dolphin) which that particular combination is to symbolize. For example, the letters that we are using here can be paired up with the sounds representing those letters, then combinations of the letters can be paired up with the combinations of the sounds. Alternatively, the various actions controllable by the computer can be done such as the food-reward-mechanical-feeder, a gate which can be opened to allow the animal to leave the computer area or to enter the computer area, links can be established over a dolphin telephone to another tank including another dolphin, etc.

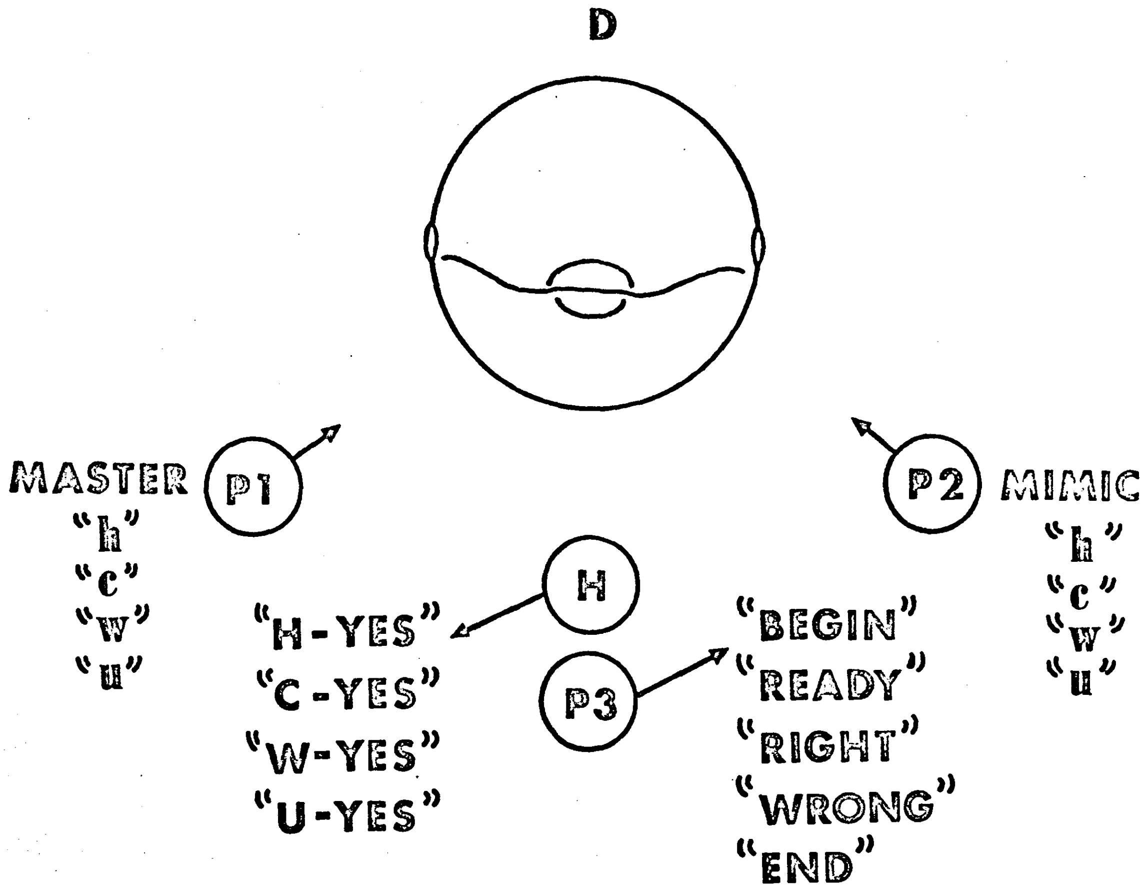
The final program which we hope to develop is that in which these elements are used in exchanges with human operators in attempts to establish communication not just with the computer but with a human being. The computer is used as an intermediary between the dolphin and the human who is posing problems for the dolphin to solve.

An alternative method of using the computer is to set up subroutines on tape where the signals are now laid out in an ordered fashion on one channel of the tape and the dolphin replies on the other channel of the tape. This was tried and abandoned as lacking the necessary flexibility for correcting the dolphin's errors. A dolphin will tend to correct his own errors as fast as he detects them but will operate very much faster if he is reminded that he has made an error. Therefore we put in the transmitter #2 to remind him of an error by mimicking him. As the program develops we hope that we will be able to have (from transmitter #3) in addition to the directions that it is already giving the outputs which are to be symbolized by elements of the code, i.e., natural dolphin recordings, natural human recordings, names in English for objects shown by a slide projector which is also computer controlled.

In summary there are several programs possible: the one proposed is only a first test program and a test code in the beginning of a long developmental testing of these kinds of aids to investigation of the intelligence and abstracting and communicative ability of the dolphin Tursiops truncatus. As was mentioned above, it will be possible to control by suitable input - output devices these experiments with evaluations of the ability of the human to solve similar problems. The above program will pose problems which are not being discussed in this report. However, we do not wish to emphasize this aspect but more the aspect of the investigation of the psychology and intelligence and communicating ability of these interesting animals.



CONFIGURATION



PROBLEM BLOCK DIAGRAM #4

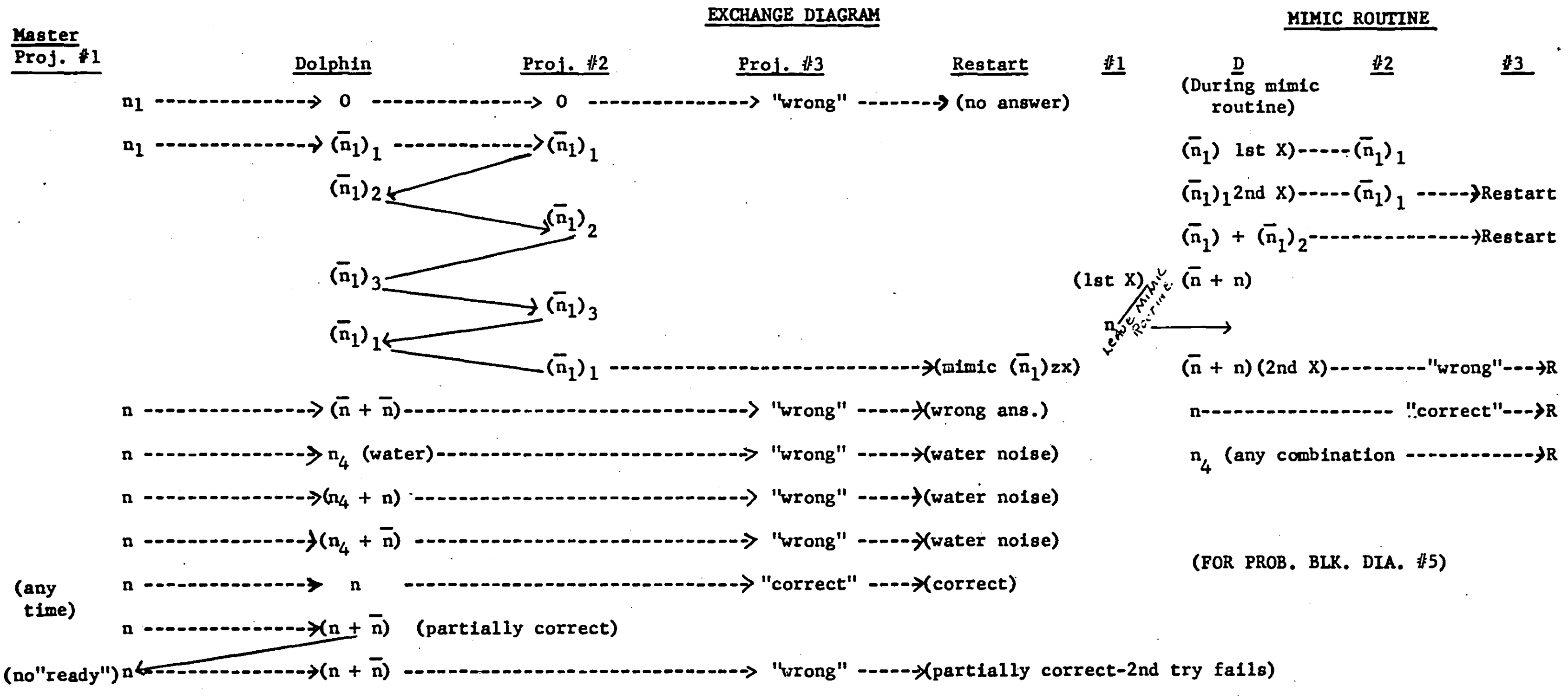
DECODING MATRIX
FOR
SEQUENTIAL PAIRS OF THE FOUR ANALOG DETECTORS' OUTPUT LEVELS, SHOWING ALL POSSIBLE COMBINATIONS AND PERMUTATIONS

First Vocalization

		OOOO	OOOH	OOCO	OOCH	OWOO	OWOH	OWCO	OWCH	UOOO	UOOH	UOCO	UOCH	UWOO	UWOH	UWCO	UWCH
	OOOO	O	H O	C O	CH O	W O	WH O	WC O	WCH O	U O	UH O	UC O	UCH O	UW O	UWH O	UWC O	UWCH O
	OOOH	O H	H H	C H	CH H	W H	WH H	WC H	WCH H	U H	UH H	UC H	UCH H	UW H	UWH H	UWC H	UWCH H
	OOCO	O C	H C	C C	CH C	W C	WH C	WC C	WCH C	U C	UH C	UC C	UCH C	UW C	UWH C	UWC C	UWCH C
	OOCH	O CH	H CH	C CH	CH CH	W CH	WH CH	WC CH	WCH CH	U CH	UH CH	UC CH	UCH CH	UW CH	UWH CH	UWC CH	UWCH CH
	OWOO	O W	H W	C W	CH W	W W	WH W	WC W	WCH W	U W	UH W	UC W	UCH W	UW W	UWH W	UWC W	UWCH W
	OWOH	O WH	H WH	C WH	CH WH	W WH	WH WH	WC WH	WCH WH	U WH	UH WH	UC WH	UCH WH	UW WH	UWH WH	UWC WH	UWCH WH
	OWCO	O WC	H WC	C WC	CH WC	W WC	WH WC	WC WC	WCH WC	U WC	UH WC	UC WC	UCH WC	UW WC	UWH WC	UWC WC	UWCH WC
Second Vocalization	OWCH	O WCH	H WCH	C WCH	CH WCH	W WCH	WH WCH	WC WCH	WCH WCH	U WCH	UH WCH	UC WCH	UCH WCH	UW WCH	UWH WCH	UWC WCH	UWCH WCH
	UOOO	O U	H U	C U	CH U	W U	WH U	WC U	WCH U	U U	UH U	UC U	UCH U	UW U	UWH U	UWC U	UWCH U
	UOOH	O UH	H UH	C UH	CH UH	W UH	WH UH	WC UH	WCH UH	U UH	UH UH	UC UH	UCH UH	UW UH	UWH UH	UWC UH	UWCH UH
	UOCO	O UC	H UC	C UC	CH UC	W UC	WH UC	WC UC	WCH UC	U UC	UH UC	UC UC	UCH UC	UW UC	UWH UC	UWC UC	UWCH UC
	UOCH	O UCH	H UCH	C UCH	CH UCH	W UCH	WH UCH	WC UCH	WCH UCH	U UCH	UH UCH	UC UCH	UCH UCH	UW UCH	UWH UCH	UWC UCH	UWCH UCH
	UWOO	O UW	H UW	C UW	CH UW	W UW	WH UW	WC UW	WCH UW	U UW	UH UW	UC UW	UCH UW	UW UW	UWH UW	UWC UW	UWCH UW
	UWOH	O UWH	H UWH	C UWH	CH UWH	W UWH	WH UWH	WC UWH	WCH UWH	U UWH	UH UWH	UC UWH	UCH UWH	UW UWH	UWH UWH	UWC UWH	UWCH UWH
	UWCO	O UWC	H UWC	C UWC	CH UWC	W UWC	WH UWC	WC UWC	WCH UWC	U UWC	UH UWC	UC UWC	UCH UWC	UW UWC	UWH UWC	UWC UWC	UWCH UWC
	UWCH	O UWCH	H UWCH	C UWCH	CH UWCH	W UWCH	WH UWCH	WC UWCH	WCH UWCH	U UWCH	UH UWCH	UC UWCH	UCH UWCH	UW UWCH	UWH UWCH	UWC UWCH	UWCH UWCH

1 1 1 1 Binary

UWCH Vocalization



n_1)
n_1) max. # req. for same cat. in a row (except when n_1 category is only category remaining
n_1) that has not been responded to 5 X correctly)

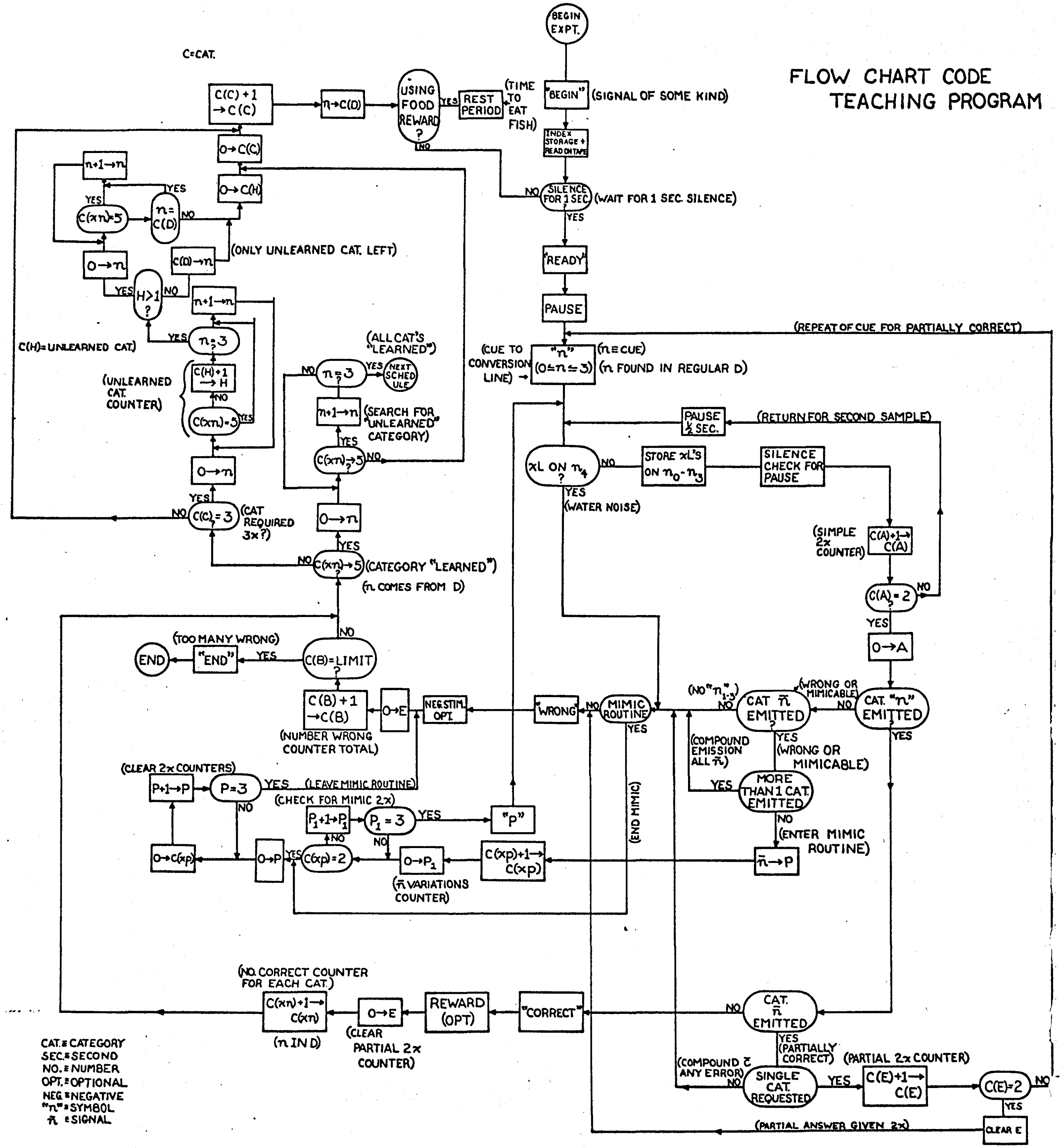
5 correct responses = criterion for not asking for this category again (by itself)(see below)

all correctly answered ↑ single category
↓ compound

n_o ---------------> n_o ----------------> $(n_o$---$n_o)$ -------> "correct" -----> (correct sequential)

n_o ---------------> n_o ---------------> anything <u>but</u> $(n_o$--$n_o)$ ---> "wrong" ---> (orig. imperfect sequential)

FLOW CHART CODE TEACHING PROGRAM



CAT.≡CATEGORY
SEC.≡SECOND
NO.≡NUMBER
OPT.≡OPTIONAL
NEG.≡NEGATIVE
"n"≡SYMBOL
n̄ ≡SIGNAL

EVALUATION OF THE LINC COMPUTER IN COMMUNICATION RESEARCH INSTITUTE

Miami, Florida

Dr. John C. Lilly and Benjamin Locke

This report will be divided into three sections: (I) the familiarization program given by the Center Development Office in July 1963, (II) the initial setup at the Communication Research Institute and (III) the subsequent operation there. In considering these three aspects, I will attempt to evaluate the amount and value of the help and information supplied by the Center Development Office and the ease of execution of each of the operations necessary for the use of the LINC. And, in conclusion, an evaluation of the resultant scientific yield as a function of the total investment by the Communication Research Institute.

(I) The Familiarization Program

Three features were emphasized in this program: (1) the programming of the LINC, (2) the theory of how the LINC works, and thus how to trouble shoot it, and (3) the technique of operating the LINC.

(1) Most of the necessary rudiments for programming were conveyed, although I think considerably more supervised practice was called for.

(2) The true test of one's knowledge of how the LINC operates is to be able to locate or repair anything that goes wrong with it. At the end of the program I did not feel at all fully competent to locate or repair anything that might go wrong with the LINC. This I think is the result only of insufficient information, the problem of which I think could be easily surmountable in the future by the creation of a trouble shooting manual giving step-by-step procedures for locating a trouble spot and containing pictures of the waveforms that occur at key places during correct operation.

(3) As far as the operation is concerned, the booklet entitled

The Console appeared to be quite adequate for explaining the operation of the console, although description of the other units were less adequate. Simultaneous with the instruction courses were the assembly, calibration, and checkout of the LINC. The assembly presented very few difficulties. However, the absence, again of a procedure manual for calibration precluded an objective analysis of the problems of calibration. The same holds true for the checkout procedures, and, as a result, I think too superficial a checkout was done on the LINC and as a consequence it was shipped off without being thoroughly and rigorously checked in all its aspects.

(II) Initial Setup

The LINC was shipped by commercial van and arrived on time and apparently unharmed. We were able to unpack, hook up and accomplish a checkout with the Killam and Hance checkout program, within one and a half hours after arrival. As far as our particular setup goes, we did not discover any malfunction of the LINC directly attributable to our adverse environment, that is, high temperature and high humidity coupled with a close proximity to our relatively open salt water tanks. However, since problems of this kind often take quite a while to develop, and since we were constantly plagued by a marginally malfunctioning LINC, it was very difficult to determine the effect, if any, of temperature, humidity and salt water vapor. The LINC was shortly put in an air-conditioned environment, anyway, as a precautionary measure.

(III) The Operation of the LINC at Communication Research Institute

As mentioned during the entire period which we have had the LINC we have been plagued by a large number of somewhat subtle and sophisticated operational malfunctions. Initially this consisted of erratic faulty operations, characterized by an apparent inability to operate correctly until the LINC had been on for five or ten minutes, by a highly sensitive tape tension, and by associated magnetic tape unit problems. As the weeks progressed these problems became more and more severe until finally it was almost impossible to read a program onto or from the tape and Lap 3 would not work at all. During this time a very large number of attempted fixes were tried, and the machine was completely recalibrated to no avail. Finally on the 23rd of May 1964, Charles Molnar came down and discovered that one of the problems was a faulty fix which was done by Severo Ornstein while the computer was at MIT. This consisted of tying off part of the magnetic tape circuitry to ground via a capacitor. This capacitor effectively short-circuited a whole element of the logic, with the result that tape transfers were never checked. With the removal of this capacitor the LINC was improved but still not fully operational. The problem then appeared to resolve into a complicated timing problem with the whole magnetic tape circuitry. The nature of this problem was recognized by S. Ornstein at MIT and also by Charles Molnar down in Miami, however; and it appears that the magnetic tape section of our LINC computer has not ever, and is still not now performing at its maximum capability. Another problem contributing to this was the fact that all of our tapes had creased edges. However, the use of brand new tapes did not eliminate the problem. In the interest of fixing these difficulties we were supplied with a new and fully calibrated magnetic tape unit from the Center Development Office and this appeared to work well for several months, however, in September again the marginal problem with tapes began and another investigation was begun. One feature that was discovered was the severe abrasion of the underside of a new tape, used for Lap 3, to the extent that this tape would no longer work. Several of these abrasions appeared to coincide with the edges of the gulleys at the inner margins of the tape shoes. This, however, appears to be a development of the new unit, and although it is an additional reflection on this section of the computer, it does not explain any of our earlier problems.

As a result of all this, the trouble shooting and repair techniques as developed up at the MIT course were rigorously tested. It soon became evident that without a really thorough engineering knowledge of the events that occurred during a given instruction and their optimum sequencing, and thus without knowing what the proper waveforms ought to look like, it was extremely difficult to do a repair on any sort of a really sophisticated problem. Faulty cords or connectors or the like could often be very quickly located but the ascertaining of maximum performance was quite difficult without some sort of a reference as suggested above.

Since, in my opinion, the LINC is not yet perfected to anywhere near its maximum capability, it has been almost impossible to determine its dependability or its stability, that is, how long do the various components last, how often recalibration is necessary and what is the effect of environment and line voltages variations.

It is recognized that 1) the LINC has a small memory capacity, and thus limited abilities as far as using a <u>compiler</u> program, and the resultant more bulky <u>compiled</u> programs, and 2) many biological uses require strict control over the size and construction of a program. But "advantages" and "disadvantages," "possible" and "impossible" to the contrary, it was found to be immensely difficult to create a large, complicated program in assembly language, rather than in a compiler or algorithmic type language. Assembly language programming, I think, necessitates a high and often unnecessary degree of knowledge concerning the operation of the machine, far more than would be necessary for algorithmic programming and, in addition, is a much more tedious, a much more lengthy and a far more complex procedure. Large and complex,but <u>constructionally uncritical</u> programs must be continuously created and continuously modified for the biological sciences, and therefore I feel that the lack of a compiler program facility to deal with this need is a severe drawback.

No difficulties were experienced with the techniques of operating of the LINC. However, I think that this was probably to a large extent the result of the thorough knowledge necessitated for trouble shooting and for programming, and therefore the case for the naive operator might be different.

Two programs are presently under development, the first one is a Click Generator Program. This program, with the aid of a random number of table occupying three quarters of memory, generates a random number of clicks from one click to 30 clicks in a strict time format. The click information is supplied to the external world via the "Operate pulse," one pulse corresponding to one click. The external apparatus generates the desired waveform from the pulse, and at the termination of the waveform generation the computer is restarted via the External Level input. This program was successfully run and recorded on tapes for use in psychological studies on dolphins.

Also under development is a conditioning program which is of a highly flexible nature. The goal of this conditioning program, initially, is to condition the dolphins to identify and differentially respond to various categories of auditory stimuli. This is done through a semi-random presentation of the stimuli, an analysis of the response including an auditory prompter mechanism for aiding in the improvement of partially correct responses, and the resultant presentation of signals, rewards or punishments to indicate to the animal the machine's final analysis as to the "correctness" or "incorrectness" of a response. Associated with this program will be a statistical analysis program to evaluate the dolphin's patterns of responses and to attempt to determine the degree of learning which has been accomplished by the dolphin.

If this first part is successful then the long-range goal is to modify the program and to use it in the measuring of cognition, i.e., the degree of abstraction and communication of which a dolphin is capable. Although this is initially designed as an auditory experiment, the connections to external equipment can be easily modified to change to other kinds of stimuli. The development of this latter program has been hindered by the aforementioned difficulties involved in using only an assembly language.

In conclusion, however, I feel that given the programming knowledge and the necessary computer technology coupled with thoroughly documented trouble shooting and calibration procedures, the naive scientist would be sufficiently equipped to productively utilize LINC computer provided that this computer is fully debugged and dependable. Unfortunately, the experience that we have had with the LINC is not especially convincing that a relatively trouble-free and dependable state is easily attainable, and even at that state the often unnecessary time and effort spent programming is so significant that it detracts from the computational capabilities of the LINC, which appear impressive.

CHRONOLOGY OF INSTRUMENTATION TAPE USED ON LINC

1. Item #3, Information Bulletin #1 dated 14 February 1964 from S. Ornstein of Center Development Office suggests use of 3M Tape Cat. #(489-3/4-150-24066).

2. 3M cost and spec sheet M-I-28 dated 1 February 1964 describes tape and quotes price. Note: No warnings as to humidity or environmental conditions are stated on this sheet.

3. On 25 March 1964 20 reels of the above described tape were ordered by Communication Research Institute thru Magnetic Products Division, 3M Co., Chamblee, Georgia. (P. O. #1400. Tape was delivered on 17 June 1964).

4. LINC logbook (page 34) shows all programs were transferred to tape in question on 20 June 1964. Old tape was discarded. Further: Perusal of logbook showed no prior tape problems, other than frayed edges, as of this date.

5. Next entry in logbook (page 35 dated 8 September 1964) describes problem in that the tape would not store information. The tape was then sent to C. Molnar for diagnoses. There is no record of a diagnoses available to us.

6. A few days prior to November 23, 1964 I experienced difficulty and frustration during an attempt to read an assembly program into the LINC. The trouble was tracked to the tape, and an examination of the block in question with a dissecting microscope showed a regular flaking of the oxide.

7. On 23 November 1964 David Peterson of Miami received the tape to submit for lab inspection.

8. On 17 December 1964, Mr. Don Tomasak of the 3M St. Paul test labs claimed the flaking was due to "high humidity." Mr. Tomasak stated at this time that subject tape was not reliable over 40-45% humidity. He also indicated that an experimental tape spool for use in humidities over 40-45% would be transmitted to Communication Research for evaluation.

9. On 17 December 1964 I spoke with the Miami representative, Mr. David Peterson, who claimed that 55% was not too high and since we were using the tape under 65% conditions, he did not think the 20 rolls could be replaced.

3M COMPANY

GENERAL OFFICES • 2501 HUDSON ROAD • ST. PAUL, MINNESOTA 55119 • TEL: 733-1110

Magnetic Products Division

January 5, 1965

Mr. F. Grissman
Communications Research
3430 Main Highway
Miami, Florida

Dear Mr. Grissman:

This letter will summarize our lab findings and recommendations on your reported problem with #489 Sandwich Tape.

The one roll of #489, 3/4" x 150', we received from you for evaluation of your problem, showed that the oxide and sandwich layers had been pulled out in sections. We recognize this type of damage and it usually occurs when the tape operates in a high humidity environment. Moisture in thickness of only a few molecular layers acts as an adhesive. The adhesion can take place between the sandwich layer and epoxy fill spots in the head and it can occur between the sandwich layer and metal surfaces.

We generally consider relative humidity to be on the high side when it is above 50%. The exact RH that begins to become problematical is not easy to pinpoint because tape damage of the type you experienced is a function of tape-to-head (or guide) pressure. For example, buildups on the tape from scratched tape surfaces cause high pressure points that aggravate the deleterious affect of high humidity. To sum up then, high humidity is to be avoided. A clean machine-tape system is essential.

Through our local representative, Mr. D. J. Peterson, you were supplied with a sample of #8972 that is less susceptible to the type of damage you experienced. We recommend you consider the product on future procurements. It is magnetically the same as the #489 you have been using.

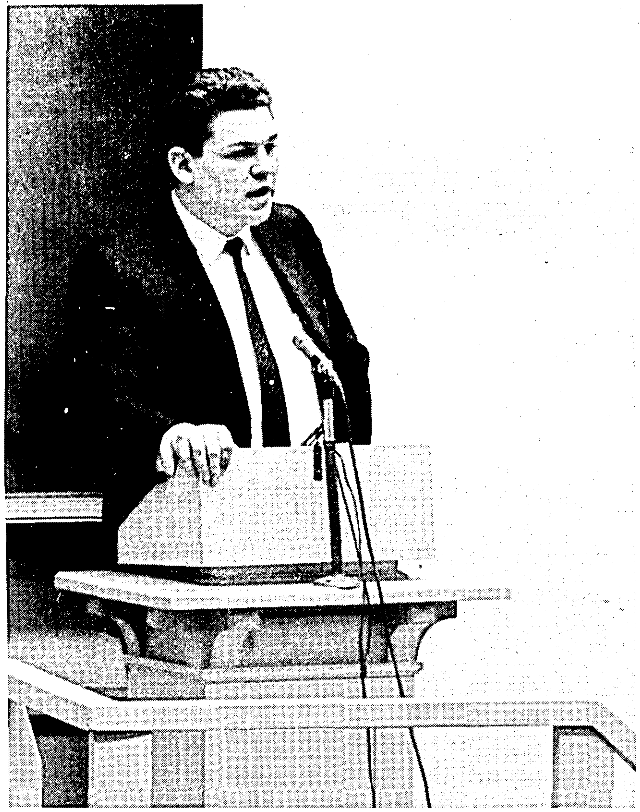
I've been advised by the Sales Department that you will be hearing from Mr. Peterson in a week or two on the #489 you have on complaint. In the meantime, we hope this clarifies the solution to the problem.

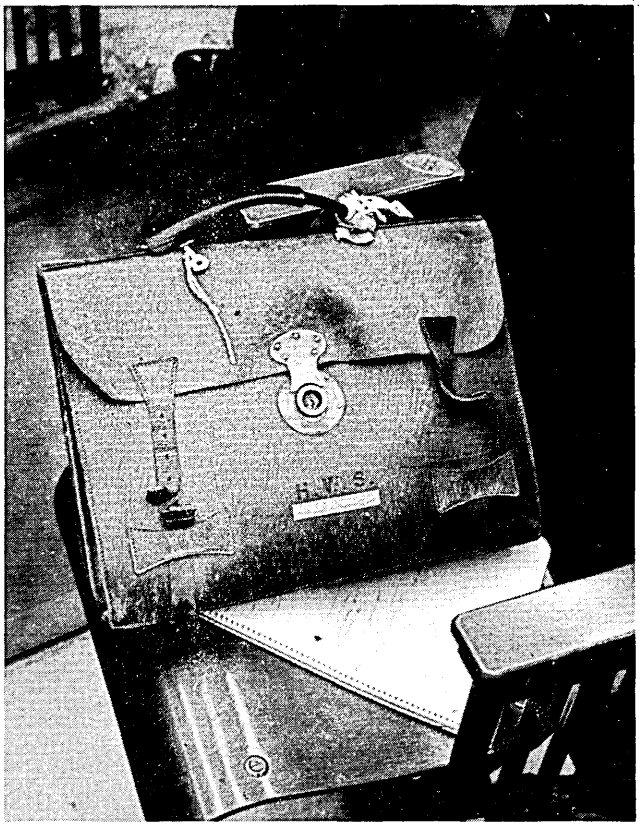
Yours truly,

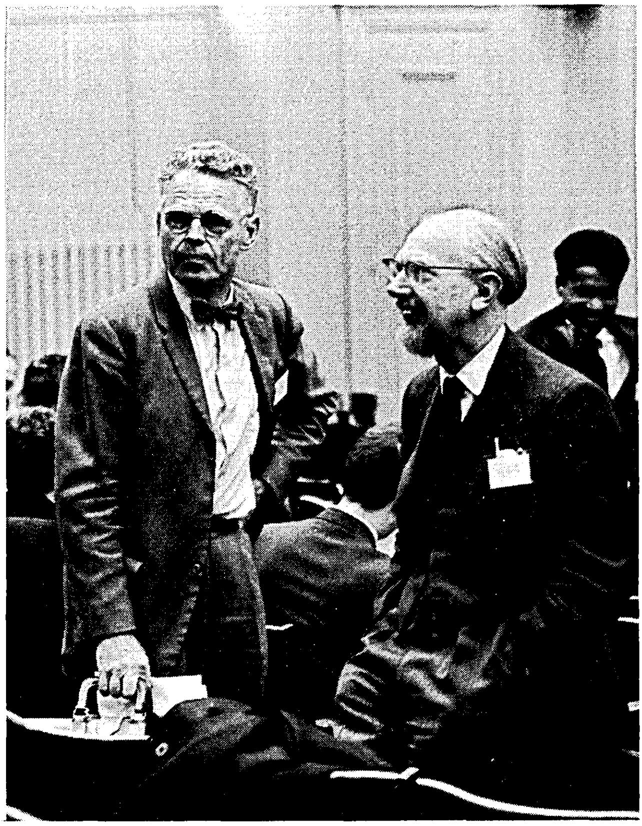
Donald F. Tomisak
Technical Service Supervisor

DFT:jn

cc: D. J. Peterson, 3M




H. V. S.



Final Report

LINC Evaluation Program

C. Alan Boneau
Department of Psychology
Duke University

5-1

1. History of the LINC at Duke

The computer arrived in September, 1963, at a time when the Department of Psychology was preparing to move into new quarters and construction was still proceeding on the building in which it now resides. It was placed in storage briefly and then a temporary laboratory in which apparatus was being prepared for the research to be described below. Early efforts to integrate LINC into the apparatus revealed engineering problems of such magnitude that LINC was temporarily abandoned while the apparatus was completed on another basis in order to continue the research program. By May, 1964, that work was proceeding well and the new quarters were complete. At that time a move was made into the new laboratory quarters.

With more time available at this point to devote to LINC, the engineering problems (involving shielding, control of ground loops, and arc suppression) were solved enabling LINC to be utilized in the apparatus. This basic system was continued, but the function of LINC was gradually enlarged. It was soon placed in charge of controlling all the timing and data gathering functions, as well as the contingent events in the operant conditioning setup consisting of two identical pigeon boxes. The addition of an external clock at this time aided the process. In addition, wiring was completed to provide an additional 24 LINC-controlled relays as well as another 12 binary input lines. The new relay controls were used in part to control a surplus Flexowriter which was installed and operating in September, 1964. At about this time LINC was on a 24-hour-a-day schedule, reliably running birds overnight, collecting the data from some 12 to 15000 trials per bird, and storing the latency data trial by trial on tape.

Later LINC was used to process and analyze the data gathered in this fashion.

Meanwhile a new set of apparatus was devised and constructed and in February, 1965, LINC was hooked into it. The main features of the new apparatus are somewhat more flexibility, an expanded span of attention (four birds at a time), and a provision for a computer-initiated automatic change from one set of birds to another so that several different groups of birds might be run overnight on several different experiments.

2. Past and present research

Up to the present time, the LINC has been used exclusively as an on-line control and analysis instrument for an operant conditioning setup. It has been utilized primarily on a color discrimination problem, but has been utilized for a number of pilot studies involving training of responses of different latencies, and the effects of various latency contingencies.

Since the apparatus has been the same for all of these projects, and the control program was general enough, we will describe here the main features of both before proceeding to a more detailed description of individual projects.

The essentials of the apparatus consist of boxes of approximately a cubic foot into each of which is placed a pigeon. On one side of the box are two holes, a small round one at head level and a larger (2"x2") at ground level. Behind the smaller hole is a piece of translucent plastic hinged so that it moves when the pigeon pecks into the hole. The bottom hole contains a device for feeding the pigeon at stipulated times in the presence of a light available only at those times. Onto the plastic key, a monochromatic light is projected from the rear. The light source is a variable monochromator to which has been connected a servo motor controlled by a set of twelve potentiometers, which can be selected by feeding a 4-bit code into a relay tree.

Provision is made for a shutter interrupting the projected light at specified times. In an early version of the apparatus, two such boxes were used, the LINC relays being used to control the monochromators, the shutters, and the magazines. The control was rather complex involving 52 of the possible 64 combinations of the six LINC relays, and a system of external holding relays. To monitor the output of the pigeon; SXL lines were in effect connected to switches on the keys.

The basic procedure consisted of a series of short presentations of the various stimuli, food being made available for a short period of time if several conditions were met. Typically, the shutter was opened to present the stimulus value and was closed again at the occurrence of a peck or the end of the period, usually two seconds. If reinforcement (food) was to be given for a peck on that particular trial, the magazine was opened for, say, three seconds, immediately following the occurrence of a peck. At the end of a trial or of a reinforcement, whichever was later, the monochromator code for the next trial was placed in the relays, and a variable inter-trial duration averaging two seconds was introduced. If a peck occurred in this period to the dark key, the delay was started again from that point in time.

The basic program consisted of a large set of subroutines for performing the various functions indicated. The core of the program was a list of JMP instructions and NOP's serving as a central control. Conceptually, the computer stepped through the list jumping to the various subroutines as required, closing the gate to them by inserting a NOP in place of the JMP when the operation was no longer required for a trial, (for example, after the first peck, the gate to the subroutine which looked for pecks in a particular box was closed). When timing operations were initiated, say, the reinforcement period, a JMP to the appropriate subroutine was placed in the table, it being replaced by the original NOP at its conclusion. A trial was over when the list contained nothing but the NOP's and the switch was made to a between trial routine. Typically, the time from the beginning of a trial to the occurrence of a peck (the latency) was measured in one of the subroutines. Quite frequently this time was placed in a location in memory corresponding to its place in the sequence and to the color code prevailing at that time. After 100 trials for each bird, the sequence of trials for both birds was placed on tape. In this inter-block time, a new random schedule of stimulus codes was generated, and placed in memory. The generation program contained a table of the various codes to be utilized and the number of occurrences of each in the 100-trial block. A random number was computed by a simple addition process, the number having the property that its range was the number of places still to fill in the random sequence. After each selection, the number in the location was reduced by one and the code corresponding to that location was placed in the random sequence.

Much of this was done with the original half-memory LINC and space was pretty much at a premium. Consequently there was a considerable amount of reading programs in and out of memory in the interblock time. Controlling all the operations was a special program called the Master-Entry Exit program which read in the appropriate programs in the prescribed quarters for them and then jumped to the starting location.

Once the original program was written it became suprisingly easy to make major modifications in it with the addition of deletion of a few steps. For example, a latency contingency was added simply by inserting a routine which added the complement of the criterion time to the latency and then using an APO for the decision. In the past several weeks, a new piece of apparatus has been completed and tied to the computer. This apparatus consists of a rack holding 4 banks of pigeon boxes in two levels. The two rows on each level are arranged outward. Between them is a movable rack containing a monochromator, and all the paraphernalia, shutters, magazines, etc., for two opposing boxes. A motor at one end of the rack when activated on command from LINC will move the central island from one position to the next. The two levels are independently tied to the computer, and within each level, all functions are independent with the exception of the color projected on the key, since there is only one monochromator linked to the two keys by a beam splitting device.

This setup is basically similar to the previous apparatus, except that it has provision for running 4 birds at once, instead of two, and can be programmed to move to a new set of birds or even a new experiment after a stipulated period of time.

To handle all the control functions, a new set of relays had to be arranged. To this end, two twelve bit registers and control logic were built into one of the data terminal boxes. These are connected through inverters to external relays identical to the original LINC relays. The external lines are sufficient at present to handle the input, consisting of the output of several one-shot multivibrators triggered by the various external events, such as pigeon pecks, positioning switches operated during movement of the boxes, and photoelectric circuits monitoring the lights in the monochromators.

Programming for the new apparatus setup was similar to the previous programs in many respects, but of course, was somewhat expanded. We have found it convenient for our purposes to have a concept of rather short subroutines each of which has a specific function to handle, say record a latency, or determine whether the conditions for reinforcement have been met. These programs have been strung together by means of a central control program consisting of JMPs and NOP's as before. Ordinarily, the jump to a routine is made with a code attached so that the function is assigned to the correct bird by virtue of the code.

The surplus Flexowriter has been connected into the computer by means of one of the twelve-bit registers feeding external relays through inverters. The manner of connecting the Flexowriter to the computer-operated relays may be of interest. The particular model which is in use is one which operates from an eight-channel tape, but upon which all the normal functions are confined to seven of the channels. In effect we have paralleled the Flexowriter's tape reader furnishing relay closures in place of the switch closures which normally would occur if a tape were being read. The eighth channel tape switch has been disconnected from the circuits in the Flex and an external line has been connected in. With the Flexowriter operating, controllable by the computer through the relays, all timing functions are controlled by looking at the external line. When its switch closes, a new code is placed in the relays; when opened, the code is removed. A little difficulty was experienced with arcing since the Flexowriter operates on a 90-volt DC system and has several inductances in operation. This was very effectively countered by placing ITT contact protectors across the 90-volt lines to ground wherethey entered the computer-operated relays. Since the completion of that system, absolutely no computer attributable trouble has been incountered. At one time we thought there was but it was ultimately traceable to some very delicate mechanical adjustments on the Flexowriter keys. Now that the art of adjusting has been mastered, we have had little trouble.

A number of programs have been written to handle Flexowriter output. The most general is one which writes from the LINC keyboard to the Flexowriter and also into memory. The output format including spacing is typed on the Flexowriter and at the same time into memory. Since much of the output desired was data, a provision was made to provide for an output matrix, the entries of which in the initial set up were tagged locations, the tag, indicating that it was the contents of the location, not the location which was to be typed out. In operation the data was read in, was converted from octal to decimal by a simple program and stored one character per word, was converted to Flexowriter code by a table-lookup procedure, the space, carriage return, and other function codes were inserted. This involved a considerable amount of tape churning as the programs to perform the various functions and the data and characters were read in and out. Once the location matrix for a given format of data was prepared, the sets of programs could be shuffled together to get out data in any format and tacked onto the end of any data analysis program. We found this to be very satisfactory.

Other programs have been written for other kinds of read out before the Flexowriter was available, the most elaborate of which displayed the Tape Unit number, Block Number, Location, Contents, and an octal-decimal conversion of the contents of a particular location, with provision to pick locations in sequence or jump to the next location with non-zero contents. One could in addition read any block from either unit from the keyboard. A less elaborate version permitted skipping through memory one word at a time with an octal version of the contents of each location displayed. This program had provision for altering the contents of the memory location from the keyboard and hence was and is our primary means of entering programs initially into memory.

In addition to these, we have had a number of programs giving a graphical display of the results of our experiments. It was possible in the days before single trial recording to monitor the performance of the birds by means of a display of points of light representing the cumulative number of pecks to each stimulus. This was routinely done at the beginning of a run to determine whether everything was functioning properly.

3. Future research

For about a year, much of my time has been devoted to integrating LINC into my ongoing research project in an efficient way and attempting to come to some closure on that research. This work was carried on, however, with the idea of making greater use of LINC's capabilities, and the apparatus constructed was designed to be as flexible as possible. It was obvious to us, also, that having LINC around meant a different look at the problems which we were handling.

Of prime interest from the point of view of this presentation is the work which we would like to do involving the latency of the response in our color discrimination task, and the way the interest in this work was prompted by the availability of LINC. Our experimental procedure had consisted of daily presentations of a random sequence of wavelength values of monochromatic light, each for two seconds with a blackout in between while the color was changed.

A peck to some of the values was occasionally reinforced by the short period during which the pigeon was permitted to eat grain. Other values were never reinforced. After many hours of this training, the birds would respond with highest probability to those values for which they had been reinforced, and responded little if at all to other values. Each day the bird would receive several thousand trials, and the data accumulated. It occurred to us that the lumping of the data was probably obscuring short range temporal effects contingent upon individual reinforcements and long range effects like changing drive level over time as the bird gradually ate enough to near satiation. Consequently we decided to record the latency on every trial, keeping an ordered list of stimuli and the latency of the response of any to them. We would then be able to go back over the data, pulling out various interesting aspects of them as we saw fit. There gradually emerged out of these data a relationship between latency on the one hand and accuracy on the other. The longer he waited before responding, the more likely was the bird to give a response to a reinforced value. We measured such things as the conditional probability of a response to a given value, conditional upon a response having not been made prior, since a response ended a trial. This measure as well as simply the proportion of responses to reinforced stimuli for various latency categories revealed the relationship to exist no matter how the latency values arose. If the pigeon was working hard, there appeared a characteristic distribution of latencies when looked at over a long number of trials. The performance at the longer latencies was always better than at the earlier ones. In fact, accuracy was a monotonic function of latency. If the pigeon was not working hard, because of fatigue, or satiation, or because the conditions were not propitious, he tended to slow down. Thus, fewer pecks per hundred trials, went along with longer latencies, but somewhat paradoxically, the relative accuracy increased. If one takes the human position that the pigeon's "heart just wasn't in the task," the performance was better for these conditions than for the hard working conditions in which there tended to appear a large number of short latency, relatively less discriminating, responses From all we can tell at present, the performance for a given latency is invariable, no matter the circumstances from which the latency came. This has lead us to speculate some upon the mechanisms underlying such a finding. In particular we are interested in looking at the possibility that some sampling process is taking place over time and that the increase in accuracy is simply a function of the fact that a larger sample was taken. In the near future we plan to explore this possibility, starting with an experiment in which the wavelength stimulus appears for a variable length of time, say 1/4 to 1 1/2 second and is replaced for the remainder of the two second trial period by a white light. We will then be able to compare the latency distribution across stimulus values for the various exposure times. We expect that this procedure will interfere with the performance at the shorter exposures, but if it doesn't it will rule out the sampling hypothesis. We also intend to train pigeons to respond with different mean latency distributions to determine whether we might increase discrimination performance by forcing them to wait longer.

It is quite likely that we would hever have become involved in these problems without LINC. There is entailed an an extremely large amount of data, at one time we were sorting through some one million relevant trials. The analyses tended to be somewhat complex, and only because of LINC were we tempted to do them at all. It is possible that some of the data sorts could have been handled by a big computer if we knew what we were looking for. With the LINC, however, we were in a position to change the analysis slightly when something seemed to be coming out, and we spent a few hundred hours at this task zeroing in on hidden meaning, a venture which would have been prohibitive on the big computer, to say nothing of inconvenient.

For example, we experienced a certain amount of difficulty in ascertaining the probability of response at different latency values under different conditions. We were interested in the conditional probability of response, that is,the probability of a response in a trial given that a response had not occurred up to that point. Our straightforward procedure would be to divide the number of responses in an interval by the number of opportunities. If the animal was working hard, there were relatively few opportunities at the end of the interval, but the probability was high that these would be responded to for the reinforced stimuli. When the animal was not working steadily, however, a problem arose. Just what should be counted as opportunities, all trials for a particular stimulus? There was a good chance that the animal was asleep or was not paying any attention to the colors on the key for many trials. It was not unusual to have several thousand trials go by without a response. The more this occurred, the more likely was the opportunity effect to be washed out, particularly if it were to occur on following periods of relatively steady working. A number of data analysis programs were devised to deal with this problem, the most complicated being one which looked at all trials and counted an opportunity for a response to a stimulus if a response occurred within x trials in the sequence from it, where x was a small number, say, 2 to 10. This was accomplished by having two pointers looking at the sequence of trials, one at the present and one x in the future. If a peck occurred in the future, a bit was put in the rightmost position of a reference work. On each trial the word was rotated one bit to the left. If a bit occurred in the last 2x + 1 bits of the reference work, then a response had to occur within x trials of the now trial. This is in effect a sliding window. This procedure is mentioned because it points up the difficulty of making a direct comparison of the relative accuracy at different latencies under different conditions. We hope to get some data to answer this question in part by using LINC to train slow latency responders.

In addition, we intend to examine the force of the peck in much the same way that we have examined latency. The equipment has been prepared and the programs for running have been constructed withthis in mind; the key as been perfected. (The key developed to measure force uses a strain guage as a transducer, the amplified output of which is fed into the A-D inputs of LINC.)

As mentioned above, we have felt the need to train latency discrimination in the pigeon and have done some work along this line. We have set up latency criteria and made reinforcement contingent upon them. A somewhat elaborate experiment along these lines is planned for the near future. It involves training the pigeon to peck on a schedule to which he is reinforced, say, on the average every ten pecks. The distribution of latencies will be obtained and the schedule will now be changed so that the reinforcement is given every time a peck is made in an interval of the latency distribution which contains one-tenth of the bird's responses. If he continues on as before, he will receive the same number

5-8

of reinforcements, but they will all occur in responses having roughly the same latency. The LINC will monitor the resulting distribution, altering the width of the interval in order to keep the probability of reinforcement exactly one-tenth. Under these conditions we predict that the pigeon will not learn to alter his distribution of latencies toward the reinforced interval as reinforcement theory would seem to demand. We have plans to complicate this procedure still further, by changing the relative reinforcement available for different birds. In some the probability of reinforcement would be less, in some higher, for responses in the selected interval after the shift. We are nurturing the hypothesis that it is not reinforcement per se, but an increase in reinforcement relative to some level which is the critical variable in performance change and we would like to see how much this change has to be in order to produce the change. We suspect that it may be possible to train shifts away from the reinforced interval by programming the correct reinforcement contingencies.

4. Training program

One can most meaningfully evaluate the program as a training device in terms of his own advancement. I feel that the month at Cambridge, while hectic, was a very worthwhile experience which netted a tremendous amount of theoretical and practical skills which have stuck with me. At present I experience no qualms about running down some malfunction in the machine (which usually turns out to be something else) and have uncovered defective diodes and transistors with no trouble. I have designed and built supplementary circuitry following the DEC logic requirements for running ancillary equipment and feel that I know enough that if pushed I could design my own computer (at least the logic) at this point. Admittedly, most of these skills came about the hard way, the Cambridge stint furnishing guidelines, but no details, and many the long hour I sweated over difficulties that came up when I was trying to get it to do things that weren't in the original plans.

There were times when I thought the machine was a monkey on my back and resented the uncompromising demands for time. After spending into the wee hours several nights in a row trying to get something to work one develops a perspective verging on despondency which fortunately usually vanishes with a couple of night's sleep.

I suspect that if there is one difficulty with the program it lies in the fact that there just was not enough time. Since I was doing most of the work myself and had other responsibilities, I have not really been able to capitalize on the machine as I hope to do in the future. About all I could do was to tie the machine in as well as I could into an ongoing research program and made the most out of LINC's talents in that context. I think at this point I need a little bit of time to relax and think out the possibilities provided by the machine. The last year and a half has been different because of the LINC. I have felt pressed, harried and stressed, but I wouldn't trade the experience for anything. There has always been a certain amount of ambivolence over the personal investment considering the possibility of having the machine taken away at the end of the period. Many times I felt that the time and energy devoted to the LINC (which had to be pried loose from other activities) would go up in smoke if the stewardship were only temporary, but LINC demands and gets.

As for specific aspects of the training program I think we might have profited by a little more enforced interchange, but realize that this is something which we can and will work out among ourselves. Certainly with such a potentially powerful device with its capabilities literally and figuratively unrealized, any opportunity for cross talk to jog us out of our usual modes of thought would have been of help to us and to the program.

5. Computer performance

Astonishment is the only way I can describe my feeling about the performance of the LINC. Except for a few tape reading difficulties which were adjusted, the machine has given no trouble whatsoever. It has been completely reliable, and as far as I know in 4000 hours has never made an error which wasn't attributable to something else when we finally understood what the difficulty was. On several occasions, we have had the machine running non-stop for as many as three weeks, with no sign of trouble. If only the mechanical equipment were as reliable, my maintenance troubles would be over.

With respect to LINC capabilities vis a vis the problems encountered in operant conditioning, it would seem to be the case that researchers reared on relay circuits are going to have some trouble probing the outer limits of capability space of the LINC. It is at least 3 orders of magnitude faster than the present problems require (not, however if one wishes to utilize specialied transducers or monitor several setups.)

There have been absolutely no procedural changes in the research program necessitated by LINC. Rather, there have been a few (and undoubtedly will be more) which were inspired by the presence of LINC. These have had to do mainly with the utilization of LINC capabilities to squeeze more information out of the experimental situation. Whereas we previously had measured only the number of pecks occurring in an extended number of trials, we soon were measuring the latency of the peck and storing the complete sequence of latencies together with the stimuli and reinforcement information, for example.

But herein lies one possible limitation to the LINC as presently constituted, namely its capabilities of handling large quantites of data. Our present four-box setup seems about the limit for the kind of information we are getting. The control program requires almost all of the first half of memory; a quarter per box of the second half is allotted for the accumulation of data. Since our schedules are stored in half-word format, we can get 100 trials in a quarter, counting 50 words for schedule, and 100 each for latency and force measures. Thus data is stored every 100 trials and then the process proceeds with a new schedule, etc. If we are to continue as at present, it is obvious that four boxes about exhausts the available storage capacity. The program itself is capable of dealing with perhaps another dozen boxes on the same problem, but there is simply no room for the storage.

6. Specifications for ideal laboratory computer

My feeling is that the LINC *is* the ideal laboratory computer. Certainly for operant conditioning setups, it has potential that we have hardly begun to realize. There have been times when we have wished that it was somewhat faster, had a somewhat larger memory, or a bigger word size. These difficulties were overcome quite readily, however, with but little sacrifice in efficiency. I think that one simply has to accept it for what it is. It was not intended to supplant the big computational machines. If one has a problem requiring facilities of that kind, he is probably in a position to use them instead of the LINC. Herein lies one weakness of the LINC system I feel. It is designed as if it were a complete unit, but it is not. If one has a good bit of data gathered on the machine and wishes to do some fairly complex analyses of it he has no recourse but resort to a big machine but has the problem of getting data from LINC into a large-computer-compatible form. As standard equipment, LINC for completeness sake, (perhaps to emphasize the fact that it is what it is) should have some form of output which is capable of easily being utilized on other machines. Admittedly it is possible to do this, and I probably will providing I can scrape up enough money to convert to somethin like this.

7. Publications based upon use of LINC

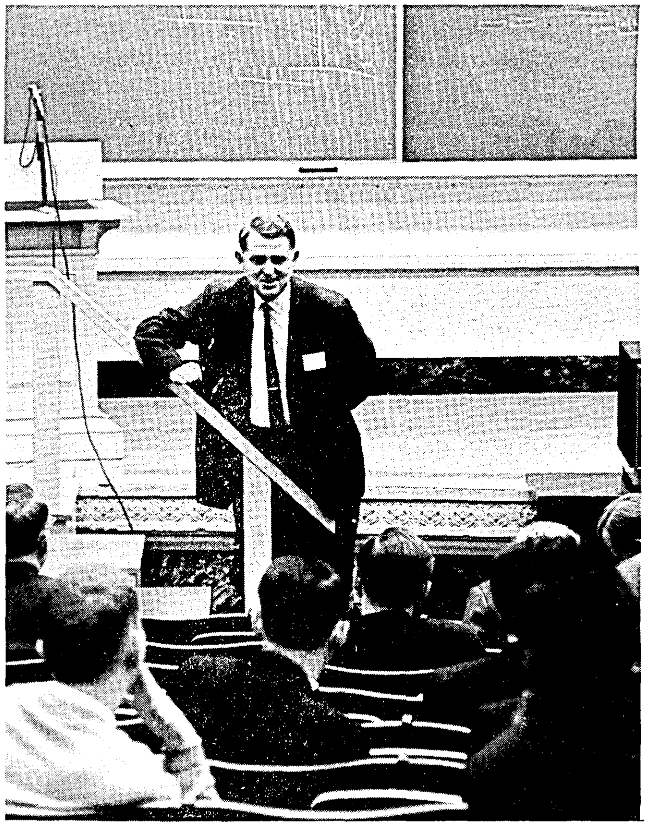
Decision processes in the Pigeon: A model for color-discrimination performance. In preparation.

Latency and accuracy as variables in color discrimination. In preparation.

Generalization gradients and the discrete trial situation. In preparation.

8. Statement on future disposition of LINC

By all means I wish to keep LINC in my possession as an integral part of my research efforts. I would hope that some procedure might be instituted which would insure that LINC would be free to move with me wherever I go in the future. This hope is predicated upon an investment of a fairly concentrated few thousand hours including a year's vacation period on my part in the LINC project.



Department of Physiology
The Johns Hopkins University, School of Medicine

G. F. Poggio, M.D.
V. B. Mountcastle, M.D.

REPORT ON LINC EVALUATION

and

PROPOSAL TO RETAIN THE LINC

6-1

TABLE OF CONTENTS

A. Letter Transmitting Evaluation and Proposal.

B. Proposed Research.

C. Evaluation Report.
    1. Development of Input/Output Facilities.
        a. Hardware.
        b. Utility Programs.
    2. Research Application of the LINC.
    3. Computer Performance.
    4. Comments on the LINC Evaluation Program.

A - LETTER TRANSMITTING EVALUATION AND PROPOSAL

THE JOHNS HOPKINS UNIVERSITY
SCHOOL OF MEDICINE
725 N. WOLFE STREET
BALTIMORE, MARYLAND - 21205

DEPARTMENT OF PHYSIOLOGY

March 13, 1965

Dr. T. T. Sandel
LINC Evaluation Board
Computer Research Laboratory
Washington University
700 South Euclid Avenue
St.Louis, Missouri 63110

Dear Dr. Sandel:

With this letter we wish to transmit to you the attached report on the use of the Linc computer in our Neurophysiological laboratories. It is our intention that this report serve as a formal proposal to you, and the National Institutes of Health, that the computer remain in our laboratories after termination of the Linc evaluation period. We make this proposal because we believe, as we hope the report will document, that the availability of Linc and the associated circuitry will allow us to undertake Neurophysiological investigations which would be impossible, or at the best extraordinarily difficult, without it. We emphasize that the capability provided by Linc is much more than that of an elegant piece of hardware, which allows one to do a tedious job with ease; for it is evident to us now that the integration of Linc into our experimental arrangement influences the design and execution of experiments to a considerable degree, and promotes the planning of an experimental investigation at a level more sophisticated by a step-function than was possible before Linc became available to us.

As you know, Dr. Gerhard Werner joined us in our initial proposal that a Linc be placed in our laboratories for evaluation. Dr. Werner has recently accepted the position of Professor of Pharmacology in the University of Pittsburg, and will be leaving this department on June 30, 1965. We suggest, therefore, that should our proposal meet with favor the award be made to Gian F. Poggio, M.D. as principal investigator, and Vernon B. Mountcastle, M.D., as co-investigator.

The presence and evaluation of Linc has occurred at a time when extensive renovations have been carried out in the Department of Physiology. We have taken this opportunity to make physical arrangements for the use of Linc by either of our two laboratories. Linc and its associated equipment now occupy a large "data-reduction" room, which is flanked on one side by Dr. Mountcastle's laboratory in which studies of the neural mechanisms in somesthesis are under way. On the other side we have created a new laboratory for Dr. Poggio's studies on the neural mechanisms in vision. We thus believe we have made physical arrangements for the most efficient use of our system.

We would like to emphasize the important role Linc has played in our training program in Neurophysiology. During the past year two post-doctoral fellows, Drs. Frank H. Baker and William Talbot, have become expert in the use of Linc, and have in fact contributed heavily to the development of our system and to the programs which are described in the body of our report. In addition, Dr. Robert DeVoe, a member of the staff of the department, has purchased recording equipment and arranged to collect his own experimental data in a form suitable for Linc analysis. We believe that the use of Linc can only increase, and its significance for the research programs in Neurophysiology is, we believe, recognized by all.

This letter is meant to be the introduction to our final report on the evaluation of the Linc and our formal proposal that Linc remain in our laboratories. There then follow, in the first part of the body of the report, brief descriptions of the research programs underway and those proposed in which Linc will be of considerable importance. Next we describe the input-output facilities we have developed, then the utility programs. Then we give some specific examples of the application of Linc to actual experimental problems. Finally we give our evaluation of computer performance and of the Linc Evaluation Program itself.

In closing this letter we wish to express to you and to your colleagues of the Linc Evaluation Group our great appreciation for the help you have given us during this evaluation. You have been untiring in assistance, and ready with skill and foresight when called upon. Above all you have striven successfully to understand our point of view, that any instrumental system of whatever complexity finds its greatest usefulness when designed to fit the needs of the experiment, and not independently of it.

Sincerely,

Gian F. Poggio, M.D.

Vernon B Mountcastle

Vernon B. Mountcastle, M.D.

GFP/VBM/mh

B. PROPOSED RESEARCH

All of the neurophysiological research in which we are now engaged, and which we propose for the immediate future, involves the use of the method of single unit analysis in unanesthetized monkeys. The philosophical basis from which we work is simple, and old. It is the premise that if the objective events evoked by sensory stimuli in the brains of experimental animals can be observed under as normal conditions as possible, they may then be compared with the sensory performance of human beings in response to sensory stimuli. The expectation is that increasing knowledge of the former will lead to a steady closure of the field of sensory neurophysiology and psychophysics, and thus lead to further understanding of the neural mechanisms of sensation and perception.

The method of single unit analysis, which we employ, is a powerful one, and its success in recent years in elucidating certain aspects of CNS function has led to its widespread use. There is, however, to our knowledge no other method which for its success depends so critically upon the way in which it is used, and upon the conditions of the experiment. This is particularly true if one wishes to study the dynamic, the time-dependent aspects of neural activity. It is less so if one's experimental objective is to study the geographic, or static properties of central neurons: such things as receptive fields, modalities of driving, etc. But if one wishes to go further, to study for example the spontaneous activity of central neurons and the way in which it is complexed with evoked activity, or to tackle the problem of whether information transmission in the central nervous sytem depends to any degree upon impulse-interval modulation, then the condition of the animal at the time of the recordings becomes of great importance.

It became clear several years ago that studies of these time-dependent aspects of the action of central neurons is useless in animals which are generally anesthetized. The effect of the anesthetic drugs influences most profoundly the natural rhythm of central neuron discharge, whether that occurring in the absence of or in response to specific sensory stimuli. We have therefore recently devised an implanted double-chamber method of recording. With it microelectrodes can be passed into the brain of the unanesthetized monkey, who is under neuromuscular block and artificially respired. The head is positioned in Horsley-Clarke space by fixation of the implanted outer chamber, and no painful stimuli are delivered to the animal. Fluid balance, CO_2 concentration, body temperature, etc., are regulated within normal limits, and the animals remain awake in what we believe is a comfortable condition for many hours.

We believe that no one would deny that observations upon a single neuron of the central nervous system are of more than anecdotal value. What is required for the successful application of the method of single unit analysis is that for each nuclear grouping a very large number of single cells be studied, ad seriatim, under as identical conditions as possible. With these data in hand, it is then possible to reconstruct the population events occurring in the nuclear region under the imposed conditions. Until methods are developed for observing many cells simultaneously, this remains the only valid application of the method.

If ones aim is to make some quantitative study of the relation between sensory stimuli and the neural activity they evoke, a precise control of the stimulus is required. For work in the somatic system we have recently developed a stimulator which allows controlled indentation of the skin (controlled to within about 2μ) for different distances, at different rates, repetitions, durations, etc. For experiments on the visual system, Dr. Poggio and his colleagues have developed a multiply-beamed visual stimulator. This allows positioning of a total of four individually controlled beams, two in each eye. For each beam, independently, there is control of position, size (down to 100μ spots), shape, contour, intensity, and color, as well as temporal pattern, of the stimuli.

Our plans for the study of these two sensory systems are nearly identical, so we shall consider them together.

1. Studies of the spontaneous activity of thalamic and cortical neurons.

Studies of this type have already been made of neurons of the ventrobasal nuclear complex. The temporal ordering of impulses has been described, and both the slow and rapid oscillations in the likelihood of discharge characterized (Werner and Mountcastle, J. Neurophysiol. 26: 958-977, 1963; Poggio and Viernstein, J. Neurophysiol. 27: 517-545, 1964). Dr. Poggio intends to extend these studies to the lateral geniculate nucleus, and in particular to study the influences upon this "internal structure of the neural message" of variations in the intensity and color of weak background illumination. Studies of the spontaneous activity of cortical neurons, of both systems, remain to be done, though some preliminary studies of this type have been made in other laboratories. Our long-term aim in these studies is to be able to completely characterize spontaneous activity, to correlate it with the level of awareness of the experimental animals, and with the on-going spontaneous slow-wave activity of the thalamus and the cortex.

2. Interaction of spontaneous and evoked activity.

Our aim in studies of this type is to discover the laws governing the interaction between spontneous and evoked activity. Evidence suggests that the spontaneous activity itself depends upon a complex input from spontaneously discharging sensory receptors and input from those "intrinsic" CNS systems concerned with maintaining levels of excitability. The question iswhether the additional input evoked by a stimulus is treated additively or multiplicatively. Some observations on thalamic cells suggest the former, and this fits with the fact that precise estimations of stimulus magnitudes, and intensity discriminations, are made by humans over a wide range of awareness, from drowsiness to agitation - which is interpreted as meaning over a wide range of spontaneous activity levels. However, for other cells this is less certain, and the possibility that multiplicative interaction occurs is open. We hope to determine which is most common, and the circumstances governing the occurrence of one or the other.

3. The quantitative relations between sensory stimulus and central neuron response.

The aim here is to order stimuli along a scale of the physical dimensions of the stimulus, the response in some reliable way - by frequency, interval sequence, etc., and to discover the laws governing the relation. This will then be compared with that controlling the human sensory performance. The result should allow some inferences about the cascaded neuronal transformations which intervene between the input (volleys of impulses in first-order afferents) and the response (subjective estimate of stimulus magnitudes, etc.). For the somatic system studies of the stimulus-response relation for cutaneous (Werner and Mountcastle, J. Neurophysiol. 28: 1965) and joint primary afferents, and for third-order joint neurons of the thalamus (Mountcastle, Poggio and Werner, J. Neurophysiol. 26: 807-834, 1963) have been completed. Studies of the thalamic neurons responsive to cutaneous stimulation are under way, and those of cortical neurons of the two types are planned for the coming year. Explorations of the stimulus-response relation in the visual system have begun, and promise to be of particular interest, for there the stimulus can be controlled most precisely.

4. Interaction of excitation and inhibition in the nervous system.

While studies of this subject have been made earlier in the somatic afferent system, in a geographic and qualitative manner (Mountcastle and Powell, Bull. Johns Hopkins Hosp. 105: 201-232, 1959), it remains to study this interaction under precisely controlled stimulus conditions in the somatic and in the visual system. In the latter not only can the interaction depending upon position and intensity be examined, but that between opponent colors as well. The implications for the general phenomenon as well as for the central neural mechanisms in color vision are obvious.

All these studies involve the handling and measurement of very large amounts of data. In a typical successful experiment it is not uncommon to record the electrical signs of 100,000 to 200,000 nerve impulses. We require measurement of all these impulse intervals, as well as an extended series of subsequent statistical analysis. We wish to use pre-programmed control of stimulus sequence patterns, and simple on-line analyses to guide the subsequent course of experiments. We believe that we have achieved this capability by incorporating the Linc into our present data anlysis system, and by producing compatible IBM tape with it when used in the data reduction mode. That the system is of central importance for our entire experimental program is certain.

C. EVALUATION REPORT

1. DEVELOPMENT OF INPUT-OUTPUT FACILITIES

The installation of the Linc in the Department of Physiology of the Johns Hopkins University School of Medicine was designed to allow for on-line analyses from each of two adjacent neurophysiological laboratories, and to make the instrument available for off-line computation to any member of the department.

The ongoing research in those laboratories which utilize the Linc is concerned with the study of some quantitative aspects of neural activity in the central nervous system as described above. The experimental data upon which these studies are conducted consist of (a) impulse activity from single neurons of the central nervous system recorded with microelectrodes, or from peripheral nerve fiber isolated by microdissection. Observations are made on the ongoing activity ("spontaneous activity") as well as on the activity evoked by physiological stimulation of peripheral receptors, and (b) activity from a population of neurons recorded with large electrodes (electroencephalogram, electroretinogram, "evoked" responses).

In order to utilize the Linc for our research purposes we found it necessary to design and build suitable input-output interfaces and to write a series of programs for experimental control and data reduction.

a. HARDWARE

Introduction

Input-output facilities were built to perform the following operations:

1. Input neural activity of the nature described above, together with code signals used to define periods of stimulation, parameters of the stimulus, etc. Data processing and analyses may be performed with the Linc either on-line with the experiment or at a later time from records collected on analog magnetic tape.

2. Output signals for the control of external equipment and for on-line selection of parameters of the stimulus delivered to the animal in the course of the experiment.

3. Input-output with Teletypewriter.

4. Write-read IBM compatible digital tape.

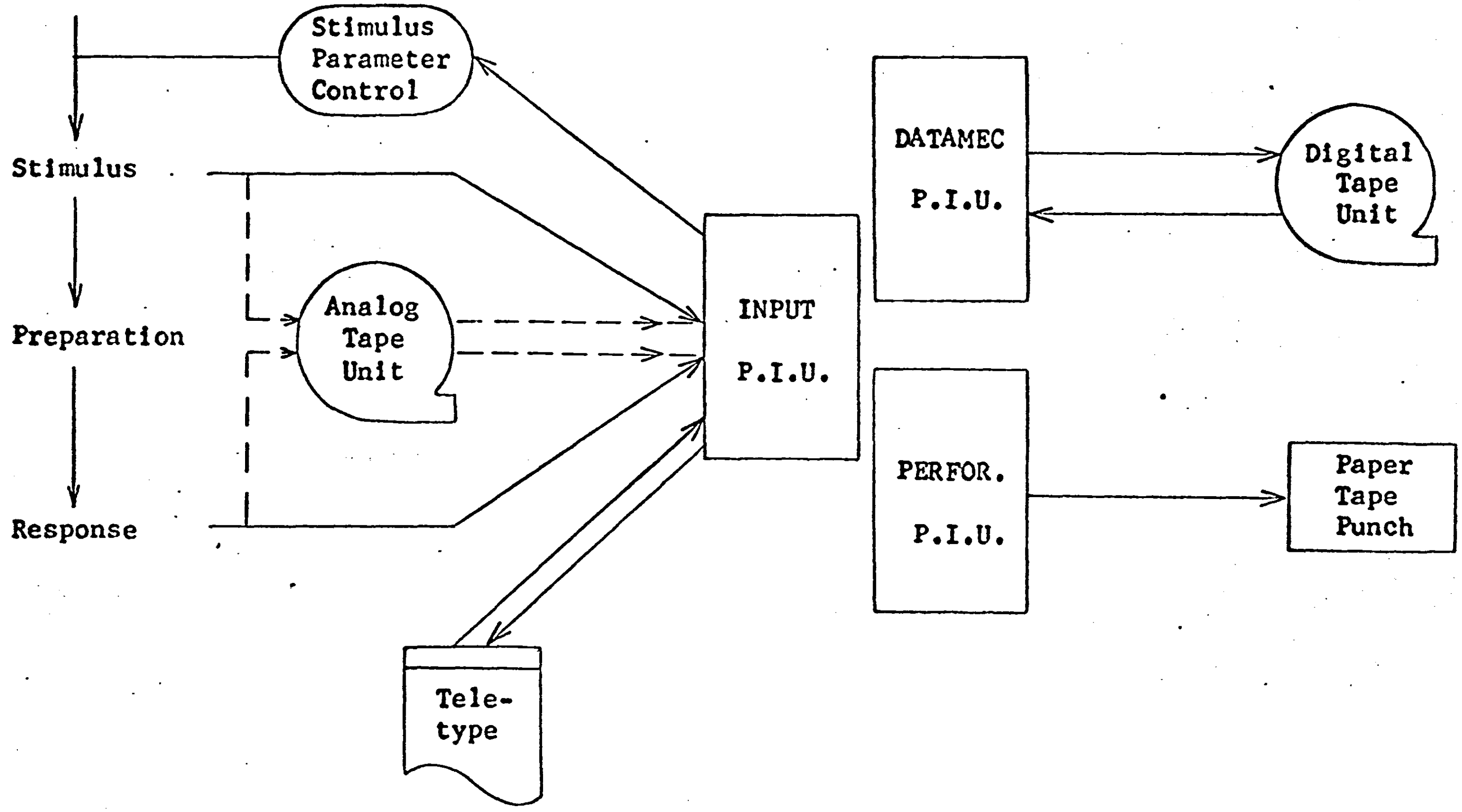
5. Output to high speed paper tape perforator.

The additional hardware necessary for these operations was constructed in three plug-in-units (P. I. U.) for the data terminal box, arranged as follows:

INPUT P. I. U.	Input neural activity and code signals Input-output connections with Teletype M33TA Output Relays External Clock
DATAMEC P. I. U.	Connections with Datamec D 2020 Magnetic Tape Unit.
PERFORATOR P. I. U.	Connection with paper tape punch set Teletype BRPE.

Datamec P. I. U. and Perforator P. I. U. can neither be used alone nor simultaneously, but only in conjunction with Input P. I. U.

In the following page is given a diagrammatic representation of


Stimulus Parameter Control
Stimulus
Preparation
Response
Analog Tape Unit
INPUT P.I.U.
Tele-type
DATAMEC P.I.U.
Digital Tape Unit
PERFOR. P.I.U.
Paper Tape Punch

the general arrangement of this system.

To obtain greater flexibility the system is designed to make the input of data continuously possible (operation 1 above), while any one of three other operation (2, 3, 4) may be simultaneously performed after being selected under program control. Each of these three operations utilizes the Linc BUFFER RELAY REGISTER outputs at the data terminal box. These outputs are used:

1. to drive the external relays located in the INPUT P. I. U. (BR_{0-5}) for operation 2.

2. to provide output code signals to the Teletypewriter (BR_0) for operation 3.

3. to provide control levels to the digital tape unit (BR_{0-4}) for operation 4.

For each operation, logical gates are employed to connect the relay register outputs with the external equipment. The selection is made by executing an OPRn instruction. The hardware and the circuitry used are detaildd in drawing #2 and #8. The following table summarizes the operational arrangement of these logical gates.

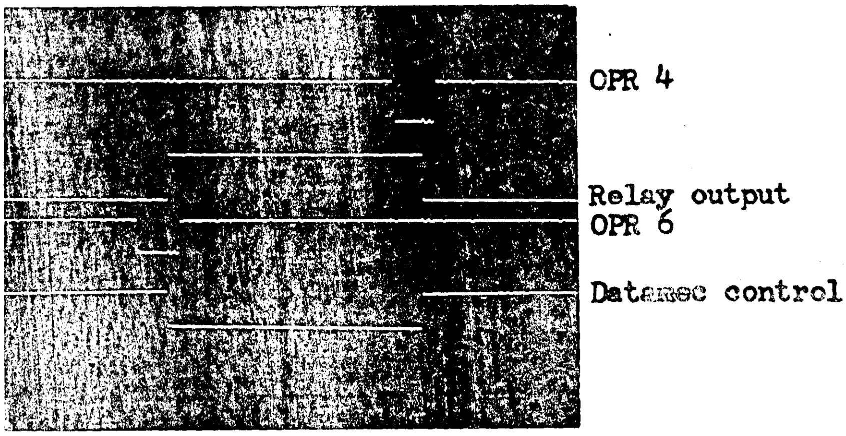
Linc Relay Register Output Selection

Register outputs	Connections			
	Following OPR 4 (Relays Select)	Following OPR 5 (Teletype Select)	Following OPR 6 (Datamec Select)	
BR_0	Relay 0	Teletype write	Forward drive	
BR_1	Relay 1	n. c.	Reverse drive	1=enabled
BR_2	Relay 2	n. c.	Rewind drive	0=disabled
BR_3	Relay 3	n. c.	Write permit	
BR_4	Relay 4	n. c.	Read threshold	(0=low ; 1=high)
BR_5	Relay 5	n. c.	Parity select	(0=odd ; 1=even)

Since no RELAY REGISTER outputs are used in paper tape perforation, when Input and Perforator P. I. U. s are used together OPR 4 and OPR 5 operate as shown above while OPR 6 simply disables both the output relays and the teletypewriter.

In the figure shown on the following page is an example of the operation of the logical gates. At OPR 4 . BOPR 2. 2 relay register outputs are connected to external relays and disconnected from Datamec control. The opposite occurs at OPR 6 . BOPR 2. 2.

Detailed drawings of the circuitry employed in the three plug-in-units may be found following a brief description of the main aspects of this circuitry.


OPR 4
Relay output
OPR 6
Datamec control

INPUT P.I.U.

The partially wired plug-in-unit given to us with the original LINC contained 8 input channels for A/D conversion and 6 output relays controlled by the BUFFER RELAY REGISTER. As noted above we have introduced logical gates between the relay register and the output relays. In addition we have made provision for three external level input channels, for input-output with the Teletypewriter and also added an external clock for program timing. No modifications were made to the Analog Input.

Level Input

At each of three level inputs signals in the range ± 10 V. are fed into a difference amplifier (DEC 1501 level standardizer) the output of which is - 3V. whenever the input level is more positive than a fixed reference voltage. The outputs of the three amplifiers are connected as follows: (see drawing #1)

i. each of them directly to an XL line (XL 4, XL 5, XL 6) and output 2 also to TN 10.

ii. the outputs for input 1 and 2 to flip-flops (DEC 4209) in such a way that one flip-flop is set by positive going level changes at input 1 (event sense flip-flop 1) and the other by either positive or negative going changes at input 2 (event sense flip-flop 2). The outputs of these flip-flops are connected through an OR-gate to XL 3 (event sense line). The output of event sense flip-flop 1 is also connected to TN 11.

Level output of OPR 3 provides TNEL causing the status of event sense flip-flop 1 and the condition of input 2 to be transferred into the two leftmost bits of the Accumulator. After transfer has occurred the same instruction clears both event sense flip-flops (OPR 3 + BOPR 2.2 through a capacitor diode gate).

The logical design of the level input was dictated in large part by our wish to perform serial interval data processing at high temporal resolution. The buffering flip-flops, "hold" an input event until it can be detected by the program, the OR-gate permits testing two inputs with a single SXL instruction, and the OPR 3 connections allow the rapid jam transfer of a 2-bit coding of the nature of the event detected.

A detailed description of the application of this logical design is given under Utility Program #4, Serial Interval Reduction Program.

Teletypewriter

The Teletype input-output connections were made as recommended in CDO Information Bulletin #6, May 26, 1964, with the addition of the logical gates on the relay register output as previously described. (see drawing #2).

As an output device the Teletypewriter has performed very well. We have experienced, however, considerable difficulties in the use of the instrument for input to the Linc. Firstly, the 200 msec turn-around time needed to obtain a simultaneous hard copy of the input was found inconveniently long. Secondly, and more important, the mechanically generated serial code is sufficiently noisy so that input read erros often occur. A much more satisfactory arrangement is to use the Linc Keyboard as the input device with direct type-out on the Teletype. (see description of Keyboard to Teletype Routine under Utility Programs).

External Clock and QKRESTART control.

The output of a DEC Variable Clock 4401 is connected to a pulse amplifier (4606) through the gated pulse input. This input is enabled by the -3V. level of the OPR 1 instruction (see drawing #3). The output of the pulse amplifier furnishes QKRESTART.

The use of the clock is primarily as a simple means to adjust program loops to a standard but variable duration. It was found particularly useful in slow rate sampling and counting operations. The clock is consulted with an OPR i 1 instruction.

QKRESTART pulses are also generated in the Input P.I.U. in response to signals from the Datamec P.I.U. (drawing #6) or from the Perforato P.I.U. (drawing #10).

DATAMEC P.I.U.

The system we designed and built to write-read-check IBM compatible digital tape is very similar to that developed by Dr. Joseph E. Hind at the University of Wisconsin. Since a complete description of that system was sent to the participants in the Linc Evaluation Program (CDO Information Bulletin #7), May 7, 1964), a short description of our system will suffice.

The main differences between the two systems stem from our use of Datamec cards to generate and check the lateral parity bit, instead of a single circuit built in the P.I.U. as in the Wisconsin system. This was done so that the read after write mode of operation could be utilized. Details of the circuitry used are given in drawings #3-#9.

6-14

Write: Bits 0-5 of the accumulator are written on tape tracks 1-6. Write clock command is given by the negative going transition of the level output of OPR 13. Due to the time requirements of the Datamec electronics, the bit configuration of the character to be written must remain in the accumulator for at least 26 microseconds after the onset of that negative level.

Read: Six-bit characters written on tape are read into bits 0-6 of the accumulator by executing any one of the read instructions (OPR 10, OPR 11, OPR 12 each of which provides transfer enabling level SNEL). The output of the laterial parity check character is connected to bit 11 of the accumulator (SN 11) and also to a flip-flop (parity error flip-flop)(drawing #6) the status of which is tested with an XL line (XL 13). Lateral parity check bit is thus read into the accumulator together with the six-bit character to which it refers. Test of the parity error flip-flop will detect the occurrence of at least one parity error in any given series of characters (one data record for instance). The parity error flip-flop can be cleared before beginning to read a record by an OPR6 instruction.

Datamec tape unit status is not tested with individual XL lines as in the Wisconsin system, but the five sense lines from the tape unit are connected to bits 6-10 of the accumulator (SN6-SN10) and the status of these lines is transferred into the accumulator with any READ instruction. Transport status is checked by testing the configuration of the appropriate bits of the accumulator. In addition the END OF TAPE sense line is connected to XL 10 in order to simplify the tape test program subroutine.

PERFORATOR P.I.U.

This unit contains the logical hardware that allows transfer of the configuration of the 8 rightmost bits of the accumulator through power relays to a Teletype BRPE High Speed Tape Punch (drawings # 10 and # 11).

The perforator emits a synchronization pulse every 9 milliseconds. This pulse may be gated by the OPR 12 output line into a series of logical 'and'-gates so that the contents of the accumulator are transferred to the punch during the 3 msec following the synchronization pulse. Because it is necessary for the desired character configuration to remain in the accumulator for the full 3 millseconds following the synchronization pulse from the perforator, programming practice is to load the accumulator with the desired configuration and then to execute an OPR12 instruction. The Linc then pauses until 3 milliseconds after the next perforator synchronization pulse at which time a QK Restart pulse is generated (drawing #3) and control returned to the program.

Summary of LINC Input/Output Connections

OPR LINES

OPR O - LINC output to teletype
OPR 1 - External clock timing
OPR 2 - N.C.
OPR 3 - Provides TNEL (to read into bit 11 of A the state of event sense flip-flop 1 to read into bit 10 of A the condition of XL 5)
Clears event sense flip-flops.
OPR 4 - RELAYS select
OPR 5 - TELETYPE select
OPR 6 - DATAMEC select; clear lateral parity error flip-flop.
OPR 7 - Read Reset
OPR10 - Provides SNEL
OPR11 - Provides SNEL for reading digital tape.
OPR12 - Provides SNEL
Paper tape perforator control
OPR13 - Write clock pulse
OPR14 - Write reset
OPR15 - KBD input control
OPR16 - Right Switches into A
Retrigger 2-1/2 character delay
OPR17 - Left switches into A

XL LINES

XL 0 - Teletype input to LINC
XL 1 - N.C.
XL 2 - N.C.
XL 3 - External event sense
XL 4 - External input #1 (-3V = yes; OV = no)
XL 5 - External input #2 (-3V = yes; OV = no)
XL 6 - External input #3 (-3V = yes; OV = no)
XL 7 - N.C.
XL10 - Eng of tape sense (-3V = end; OV = not end)
XL11 - 1-1/4 character delay (-3V = expired; OV = not expired)
XL12 - 2-1/2 character delay (-3V = expired; OV = not expired)
XL13 - Lateral parity error sense (-3V = Error; OV = no error)

TAPE STATUS AND PARITY CHECK (Left half of accumulator following OPR10, OPR11, OPR12)

A6 - (octal 100) - Auto-local sense - 1 = auto; 0 = local
A7 - (octal 200) - Density sense - 1 = high; 0 = low
A8 - (octal 400) - Write enabled sense - 1 = not enabled; 0 = enabled
A9 - (octal 1000) - Rewind sense - 1 = not rewinding; 0 = rewinding
A10- (octal 2000) - End of tape sense - 1 = end of tape; 0 = not end
A11- (octal 4000) - Lateral parity error - 1 = error; 0 = no error

DRAWINGS OF PLUG-IN-UNIT CIRCUITRY

(Note: in all drawings the circled numbers indicate the pins of the P.I.U.-DTB connectors or of the connectors to external equipment. The circled letters refer to the pins of the specified DEC card.)

b. UTILITY PROGRAMS

1. General purpose display subroutine (KIN)
2. Keyboard to Teletypewriter I/O Routines (KBDTTY)
3. Program manuscript type-out (MANOUT)
4. Digiral tape program (DATAMEC)
5. Serial Interval Reduction Program (SERINT)

1. Subroutine KIN

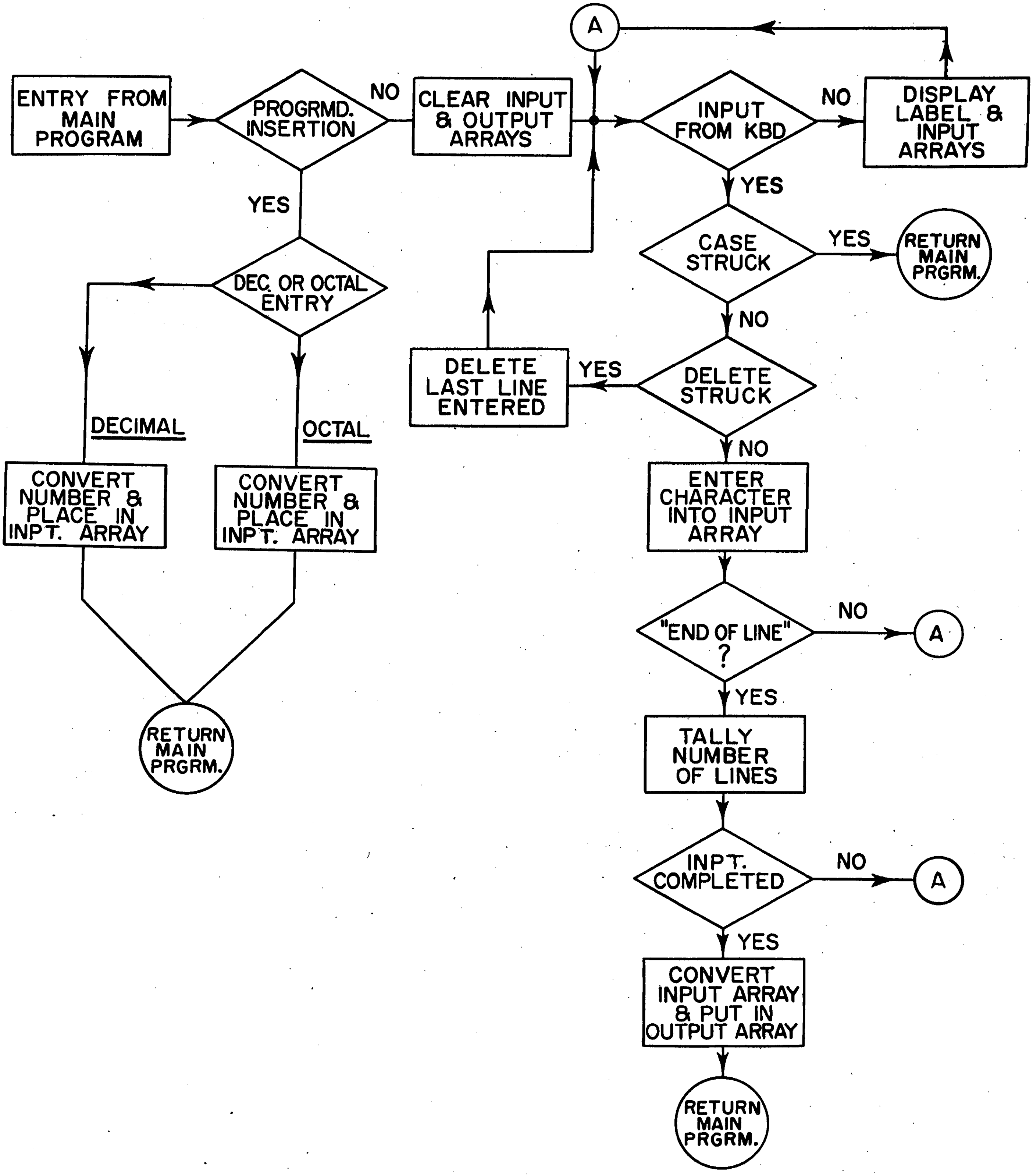
Kin is a general purpose subroutine for keyboard input with display of titles and numerical parameters; alternatively, it can be used for the display of constants and parameters from the main program. In either case, the main program which calls KIN in must contain a series of half word keyboard representations of letter occupying two lines of title, and the number of labels (n) which are to be filled by numerical parameters. The main features of the routine appear in the accompanying flow chart.

a) Kin for display of keyboard input: Kin displays two lines of title (obligatory) which can be followed by a list of labels (such as for instance: S_1 = ; etc). Numerical values for the labels can be entered from the keyboard either in octal or in decimal numbers. They must not exceed double word length and must be entered in the order in which the labels appear on the oscilloscope screen. The typing of each value is terminated by EOL if it is an octal number; a "D" followed by EOL is typed after decimal numbers. The numbers are displayed as they are entered. The last line entered may be deleted by typing DEL. When the number of numerical values entered equals to n and when the last EOL is typed, Kin converts the numerical values entered into binary numbers and leaves them in an array of double length numbers in KIN, starting with location 1350. These numbers can be utilized by the main program. Control is then automatically returned to the main program two locations after the point in the main program from which Kin was entered (JMP KIN + 2).

b) Kin for display of numerical values from the main program: Numerical values are read from the main program and ae inserted into the appropriate locations of the title-label display.

6-19

SUBROUTINE KIN



2. KBDTTY--Keyboard to teletype I/O routines.

This is a package of subroutines designed to make the keyboard and teletypewriter an effective pair of devices for the initialization of programs and the display of results. The following are included:

Teletype alphanumeric output routine: This routine takes a 6-bit keyboard code character from the accumulator, translates it to teletype code and prints it out. Upper-case characters are printed by entering this subroutine consecutively with 23 and the character code. The routine is suitable for printing out characters as they are typed in through the keyboard, or for printing lists of characters stored in memory.

Octal loader: Thispermits keyboard loading of positive or negative octal numbers without leading zeros. Digits are typed out on the teletype as they are entered. When a space or EOL is struck control returns to the calling program with the octal number in the accumulator. Only octal digits, spaces, and EOLs are accepted by this routine. The type-in may be limited to 3 or 4 characters. Numbers may be deleted at any time before the terminator is struck.

Decimal loader: This is like the octal loader except that a decimal-to-octal conversion (single precision) is performed after the number is terminated and the octal number loaded into the accumulator. Not more than four decimal digits will be accepted. If the octal conversion exceeds 7777_8 an error indication is typed out.

Octal typeout from the accumulator with leading zeros replaced by spaces or with leading zeros printed. All numbers are treated as positive octal numbers. The printout can be limited to the rightmost 2, 3, or 4 digits.

Decimal typeout from accumulator: This is like the octal typeout except that an octal to decimal conversion is performed before the typeout. Two routines are available, one of which converts octal numbers greater than 4000_8 to positive decimal equivalents, the other to negative decimal equivalents. When used the minus sign is moved next to the first printed digit.

6-21

3. MANOUT--Program manuscript typeout

This program uses the teletypewriter to print out a copy of Lap manuscripts. Initialization from the Linc keyboard provides options for printing a 20 character name on each 100_8 line page, for printing the binary conversion of each manuscript line, for typing a list of tags and equalities, and for restricting the type-out to a specified span of lines. To simplify annotation of the manuscript the program will pause at the end of the current line whenever a keyboard key is struck. Thereafter KBD input results in teletypewriter output until a meta or case-G is struck. The former restarts manuscript type--out after a line-feed and carriage return; the latter causes return to Guide.

4. DATAMEC Program

This is a general purpose program to write, read and check IBM compatible digital tape, low density, odd or even parity. For specific purposes other digital tape programs are used in the laboratory (see, for instanae, SERINT.)

Datamec program occupies two quarter memory (QN 2+3) and consists of several subroutines and subprograms. The entire Datamec program uses 761 octal memory locations and index registers 0, 1, 2, 3, 4, 5, 6, 7, 10. A brief descritpion of the program is given below, and its manuscript is appended.

Tape format used with Datamec program

Linc bit A_0 = Track 1 Datamec unit
" bit A_1 = track 2 " "
" bit A_2 = track 3 " "
" bit A_3 = track 4 " "
" bit A_4 = track 5 " "
" bit A_5 = track 6 " "
Lateral Parity bit = track 7 " " (ODD for binary; EVEN for BCD)

Inter character spacing, low density = 0.00544" (184 CPI, nominal 200 CPI)
Inter character time = 120 μs (fifteen, 8 μs Linc cycle time)
Longitudinal check character (LRCC) gap = .0216"
Inter record gap (IRG) :
write after write = MIN .720", NOM .810", MAX .900"
write after read = MIN .750", NOM .790", MAX .840"
write after backspace = IRG increases MIN 0, NOM .050", MAX .100"
End of File Mark (EOF) = 17_8 even parity + LRCC
Write gap with tape at load point = 3.65" 0.05" to first record.

Subroutines and subprograms

All subroutines and subprograms other than delay subroutines are entered with tape at rest. With the exception of Rewind return to calling program will be with tape motion halted.

a. Programmed time delays:

Initial Tag 1A = for 2.7 msec delay (free tape travel .123")
" " 1B = " 5.0 msec " (" " " .225")
" " 1C = " 5.9 msec " (" " " .265")
" " 1D = " 11.7 msec " (" " " .525")
" " 1E = " 17.0 msec " (" " " .764")
" " 1F = " 66.4 msec " (" " " 3.000")

b. Initial advance: Initial Tag 2A
Generates initial write gap (erase) with tape at load point.

c. Backspace: Initial Tag 3G
Backspace one record of any length. When motion stops the tape is properly positioned for any other operation to follow.

d. Rewind: Initial Tag 2R

Initiate rewind of tape to loadi point. Control is returned to calling program about 17 msec after rewind command is given. Return configuration: c(A) = 20

e. Write record: Initial Tag 2B

This subprogram writes one record of 2000_8 six-bit characters taken successively from upper and lower half of 777_8 twelve-bit Linc word stored in QN 6 and 7. During writing process the running Longitudinal Check Sum is computed using the character configuration loaded in the Accumulator. After writing the LRCC the program waits (read after write mode) for the end of record (first 2.5 delay expiration), retriggers 2.5 delay (OPR 16), then checks status of lateral parity error flip-flop (XL 13). If no error the program waits for LRCC or for expiration of second 2.5 delay and then checks longitudinal sum by testing Computed Long. Sum = LRCC.

If no errors are detected tape motion is halted in the required way, for generation of the after-write portion of the IRG (*), and control returned to calling program. Return configuration: Accumulator = 0.

If lateral parity error is present orlongitudinal sum does not check, motion is stopped and the tape backspace-erased (WRITE PERMIT left on) the length of the record just written. The 2000_8 stored characters are then re-written over the same portion of tape. Re-writing may be repeated up to six times. If after these six trials an error is still detected, the tape is backspaced once more and then erased forward for about 3.0". Writing of the same record is then tried again. The entire procedure may be repeated up to three times. If error is still present, at the end of the third 3.0" advance erase the tape motion is halted as above, and control returned to calling program. Return configuration: Accumulator = 1.

An outline of this program is given in flow-chart #1.

(* - Note: Removal of WRITE PERMIT generates a spurious character on the tape stationed over write head. To eliminateit, at the end of a write operation WRITE PERMIT is removed at the time STOP command is given. Once the tape has stopped it is backspaced the length of one START+STOP motion (JOG). The spurious character is thus positioned in front of the write head and the successive write operation will erase it before writing the first valid character.

f. Write End of File: Initial Tag 2K

This program writes an EOF mark (17_8 even parity + LRCC), checks it in read after write mode, and generates the after-write portion of the IRG (STOP+JOG). Erros detection, automatic re-writing and return configuration are similar to those used in the Write Record subprogram and they are detailed in flo-chart #2.

g. Read Forward: Initial Tag 3A

This subprogram reads and identifies: (a) one record, defined as any two consecutive characters separated by less than 2.5 intercharacter spaces.

6-24

1 character record is considered as a read error. (b) one End of File mark.

Characters are read into bit 0-5 of the Accumulator and store successively in left and right half of core registers QN 6+7. Record cannot be longer than 2000_8 characters. If longer records are to be read set initial storing address (location Tag 3B+1) to beginning of QN 5 (for records up to 3000_8 characters) or QN 4 (to 4000_8 characters).

(Note: With each character, the tape transport status senses and lateral parity check bit are read in bit 6-11 of the Accumulator. See description of Datamec P.I.U.).

The program tests for lateral parity error, missing characters, and longitudinal check sum. If no error, the tape is stopped within the IRG at a position that then allows execution of any other tape operation to follow, and then control returned to calling program.

Return configuration: (a) Record: Accumulator = 0; Register 1 = no. of characters stored (octal); unused portion of QN 6+7, if any, filled with negative zeros. (b) EOF: Accumulator = 1; Register 1 = 3000_8

If any error is detected the tape is stopped, backspaced the length of the record just read and the reading operation tried again. This procedure is repeated up to 12_8 times, alternately with low and high read threshold. If at the end of this series of trials an error is still present, the tape is stopped within the IRG as above and control returned to the calling program. Return configuration: for permanent read error: (a) Record: if lateral parity error, Accumulator = 4002; if missing character error, Accumulator = 2002; if longitudinal check character error, Accumulator = 1002. (b) EOF: Accumulator = 4002, Register 1 = 3000.

An outline of this subprogram is given in flow-chart #3.

h. Skip EOF Forwards: Initial Tag 3L
Backwards: Initial Tag 3K

This subprogram will skip any given number of End of File marks then enter READ FORWARD subprogram to check the last EOF mark identified and to position the tape past it, to read or write the first record of the next file. Return to calling program will be from READ FORWARD and therefore return configuration will be the same as that of the latter subprogram. In addition Register 10 will contain the number of records that constitute the last file skipped. The complement of the number of End of File marks to be skipped must be filled in register 5 before entering the program.

An outline of this subprogram is given in flow-chart #4.

5. SERINT: Serial interval reduction program

This program is the latest of a series of systems used by this laboratory for continuous data reduction. All previous methods have required that data be recorded on analogue tape and replayed at greatly reduced speed for the reduction itself. The present method can, when desired, be used for on-line reduction or for off-line reduction in real time. It can even be used to reduce data played back from analogue tape at speeds higher than recording speed whenever a unit time longer than 240 microseconds is permissible.

The data for which this program is intended consist of one channel of nerve impulses and a second channel carrying a three level code designating the analysis period (analysis gate) and time of stimulation (stimulus codes). The program generates a list of time intervals between successive 'events'. Significant events for this reduction are the onset times of nerve impulses, in the first channel and the onset and offset times of stimuli in the second channel. The program produces a sequence of double word entries in which the duration of an interval is marked by the ten low order bits in each word (in one word these bits show a count of 240_{10} microsecond time units; in the other they show a count of 2000_8 X 240_{10} microsecond units). In real time the longest possible interval is slightly greater than four minutes, the shortest possible interval is 240_{10} microseconds, and the accuracy is +240, -0 microseconds.

As each interval is registered it is marked (in the two high order bits of the word containing the high level counter) with a code identifying the nature of the event which terminated that interval. These codes are brought into the Accumulator via TN 10 and TN 11 (See Input P.I.U.). They are suitable for direct use by LINC programs designed to operate on this kind of data; they are also recognized by an IBM 1401 program which converts our double precision octal expressions for the intervals to fixed point decimal multiples of milliseconds. These converted expressions in turn serve as data for the laboratory's library of IBM 7094 analysis programs.

This method of reduction is made possible by the characteristics of the Datamec tape recorder and by the buffering flip-flops on the external level inputs through which data are brought into the LINC. Although writing data on the Datamec requires a program loop more than half of which is committed to the bookkeeping and mechanics of tapewriting, some time is left over in which an input process can continue even while data are being put out. The LINC core memory serves as a buffer. The buffering flip-flops and their connections (see INPUT P.I.U.) were designed to reduce the time needed for examination of the external levels sufficiently so that a check of external levels on two lines could be made and intervals recorded within 240 microseconds when 6-bit characters were being written on tape every 120 microseconds.

6-26

The general pattern of the program is revealed in the attached flow sheet. The teletypewriter is used to generate a permanent copy of the name of the tape, the file number, and the title of the data transfer as this information is typed in from the Linc keyboard. (Up to ten 72-character lines of descriptive title can be entered at this time. This title is associated with the data in all subsequent data handling procedures.) The data transfer title is translated into IBM compatible alphanumeric code. Our basic tape subroutines (DATAMEC program) are used to write the title on the digital tape. Then the 240 microsecond loop program is called in to make the transfer itself. To help insure reliability in transfers and to minimize false starts, the routine begins with a triple pause; data collection begins with the first event following the onset of an analysis gate which in turn follows the striking of a keyboard key.

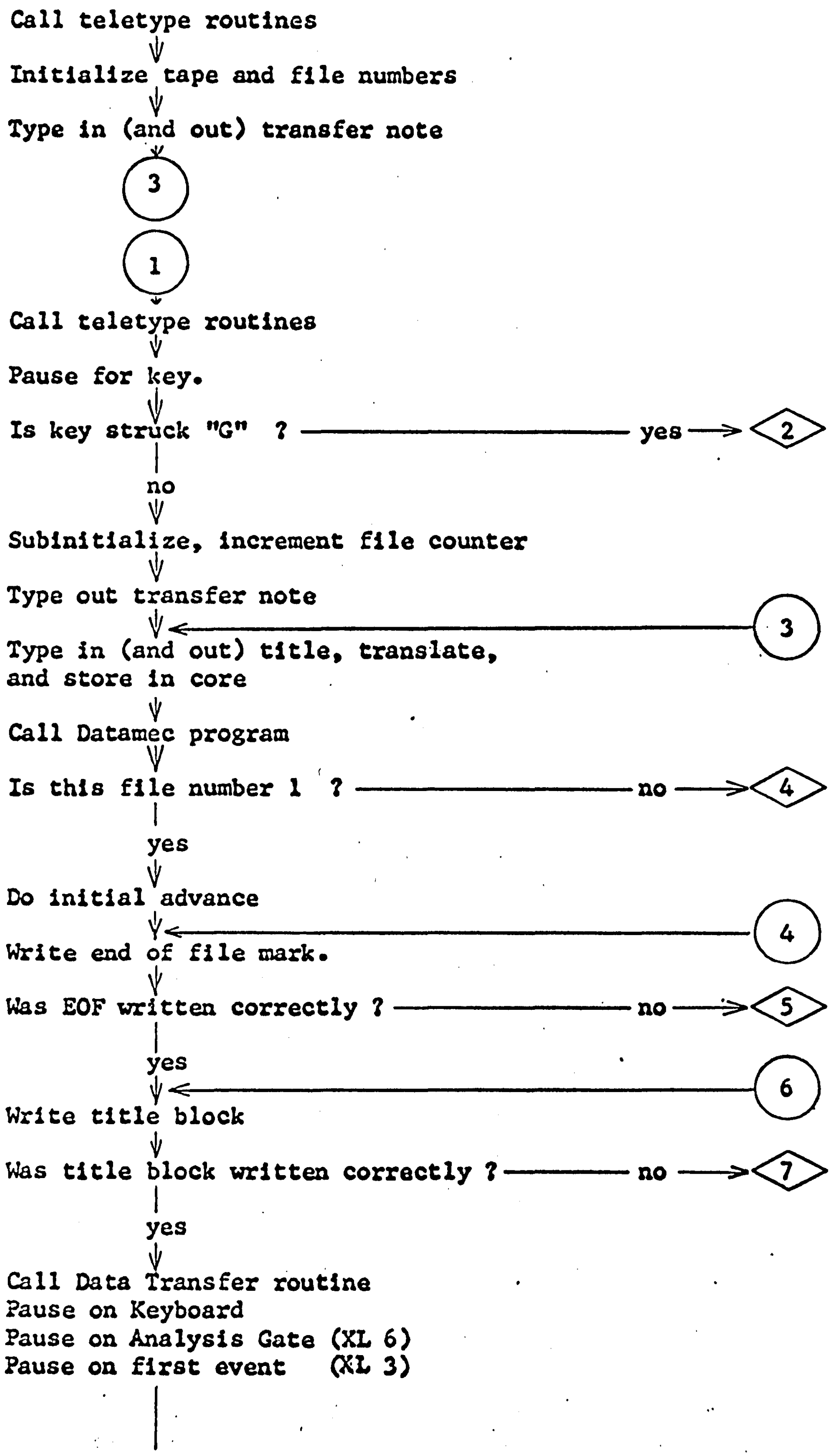
The initial data collection occurs in a relatively simple loop from which exit is possible only when the analysis gate is removed or half of upper memory is filled. Counting of time is done in terms of 240 microseconds units because this corresponds to twice the character time in Datamec writing. When control leaves the initial loop because half of upper memory is filled, it enters the loop where simultaneous input and output of data occur. All counting in this loop is done in terms of pairs of character (6-bit) dumpted on tape. Two characters are put on tape for each tally of the '240 counter' and therefore for each test of external events. The only exit from the loop occurs when the entire half of upper memory from which writing is occurring has been written on tape. If the writing proves to have been successful, control returns to the simple data input loop until the other half of upper memory is ready to be dumped. If the writing produced a lateral parity error (detected on the laterla parity flip-flop before the LRCC has passed the read head of the Datamec) the tape is baskspaced through the faulty record, erased forward, and another attempt made to write the data. Throughout this process data continues to be collected. Alternate writing and erasing will continue until success occurs or the alternate half of upper memory is filled. In the latter case the program halfs with an error type-out. When in the simple data collection loop the 'off' state of the analysis gate is detected a final data block is written on the digital tape and control returned to a sub-initialization routine which types out a transfer note with the name of the tape and the new file number inserted into it. An exit to guide is provided just before this type-out so that the run may be terminated at the end of any transfer.

Before taking the data tape generated by this program to the IBM computing center or before using it for LINC analysis the data are transferred from Datamec tape to LINC microtapes and checked for errors in longitudinal parity and for missing characters. (There is not time to check these during the data transfer, only lateral parity then being checked.) The LINC tapes form the permanent storage medium for these data. During the transfer a permanent transfer note which includes the transfer title is made as an index to the contents of the LINC tapes.

The basic SERINT program consists of 11 blocks of LINC tape which are read into LINC core as follows:

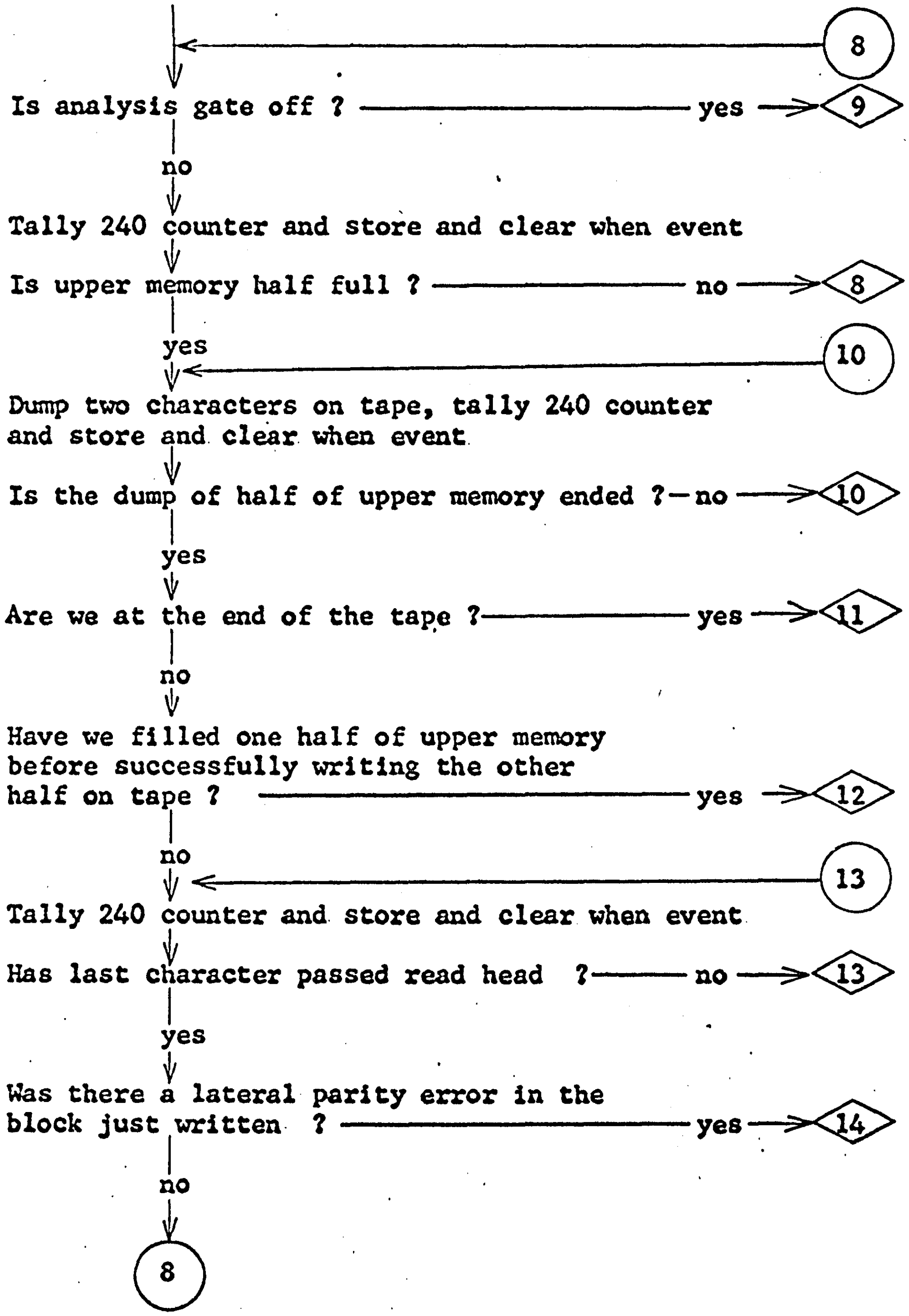
MAIN DRIVER	SERINT	QN 0
TELETYPE ROUTINES	TITLIN	QN 1
	KBDTTY	QN 2, QN 3
DATAMEC ROUTINES	DATMEC	QN 2, QN 3
DATA TRANSFER ROUTINE	240 SR	QN 1, QN 2, QN 3
END RUN	SEREND	QN 1
DIAGNOSTICS, etc.	WORDS	QN 4

SERINT FLOW CHART -- 1

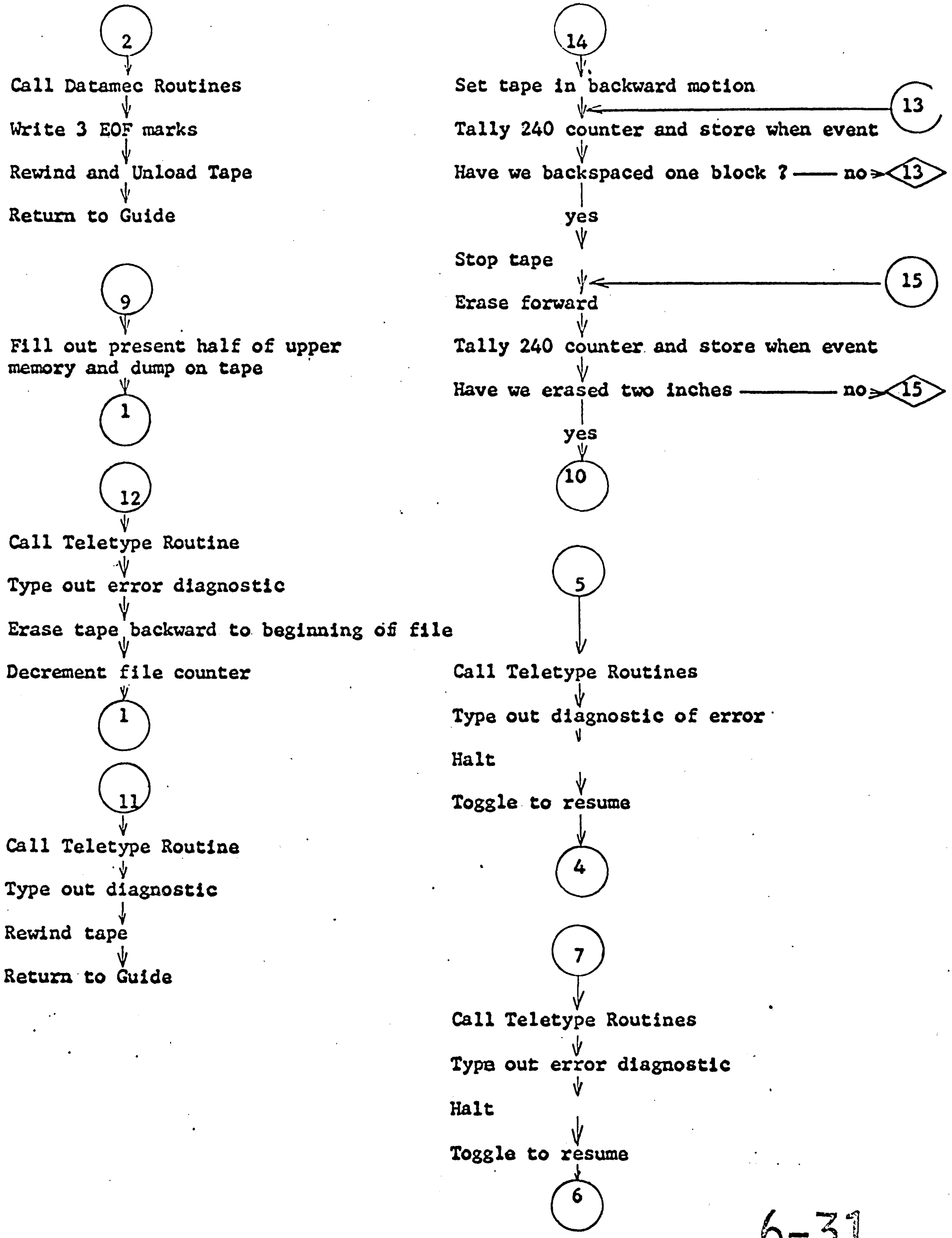


Continued on pg. 2

SERINT FLOW CHART - continued -- 2



SERINT.FLOW CHART - continued -- 3



2. RESEARCH APPLICATION OF THE LINC

a) ANALYSIS OF IMPULSE ACTIVITY OF SINGLE NEURONS

The neurophysiological investigation in which we are now engaged have been described at length under "Proposed Research". Specific computational problems we are concerned with relate to: 1) Choosing measures for neural activity and testing their significance; 2) In particular devising useful measures for 'spontaneous' or on-going activity so that variations of this activity may be correlated with other measurable events; 3) discovering the functional relationships between neural responses and the eliciting stimuli; 4) overcoming the signal-to-noise difficulties in weak responses of units exhibiting ongoing background activity.

In this section we wish to give some examples of our use of the Linc in connection with these problems. The nature of the experimental data and the arrangement to enter them in the computer have been outlined previously (see Input/Output facilities).

General survey programs

We have written a series of simple programs providing a rapid survey of the experimental data. All these programs utilize the oscilloscope display. Since these programs perform well known analyses no lengthy description is necessary. They include the following:

- a program for measuring and storing on Linc tape intervals between events. This is used for short series of intervals; longer series are collected using utility program SERINT described above.

- a program which constructs and displays a histogram of the numerical values of the intervals between neural events.

- a post-stimulus histogram program.

- a program which counts the number of events occurring within a selectable gating time following a stimulus, and displays the number of events vs. trial number for each of a series of stimuli.

- a program to construct and display as estimate of the expectation density function of a sequence of events. This function reveals certain temporal aspects of the internal structure of the sequence of neural impulses.

Examples of the results of some of these analyses are shown in the following figures. They refer to data collected from single neurons of the lateral geniculate nucleus.

Figure 1 - Expectation density. Abscissa scale: 8 msec./div. Ordinate scale in this figure and those following indicates number of counts for the histogram class. This figure demonstrates the existance in the impulse sequence of periodically changing probability of discharge; period of about 15-20 msec.

Figure 2 - Expectation density. Same data as in Figure 1. Abscissa scale 32 msec/div. Note periodicities of about 160 msec.

Figure 3 - Interval histogram. Same data as in Figure 1. Abscissa scale 8 msec/div. Note that there are modes separated by 15-20 msec. and also a peak at 160 msec.

Figure 4 - Time histogram of the discharges of single units before during and after stimulation with light pulse. Abscissa scale: 32 msec/div. Stimulus occurs 100 msec. after t=0 and lasts 40 msec.

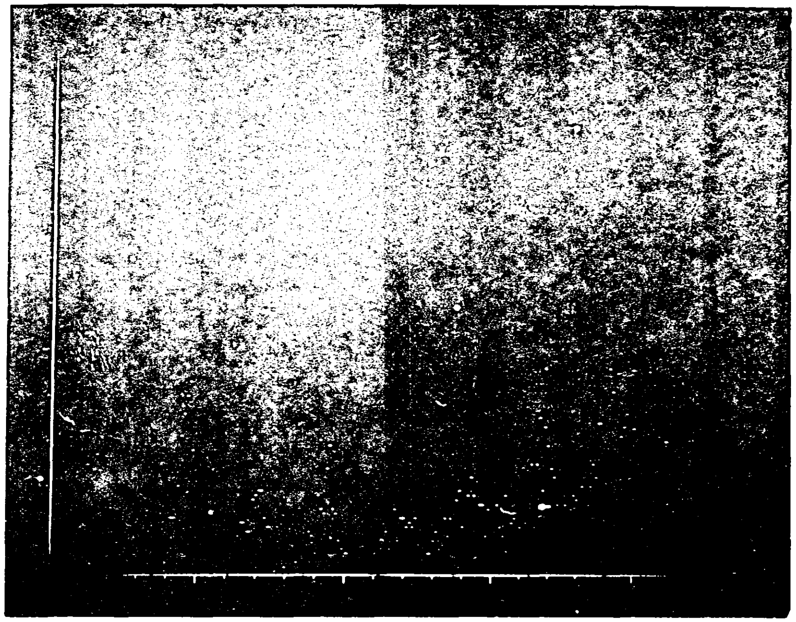


Fig. 1

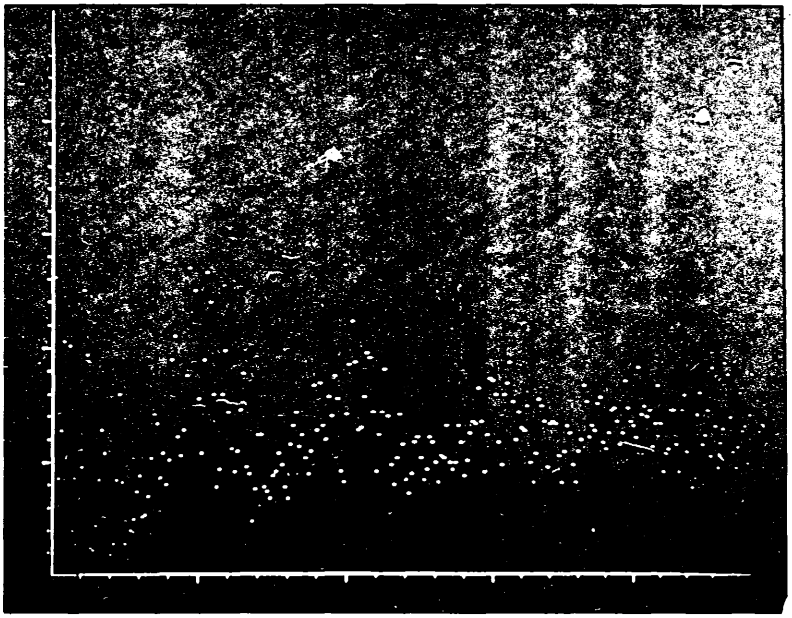


Fig. 2

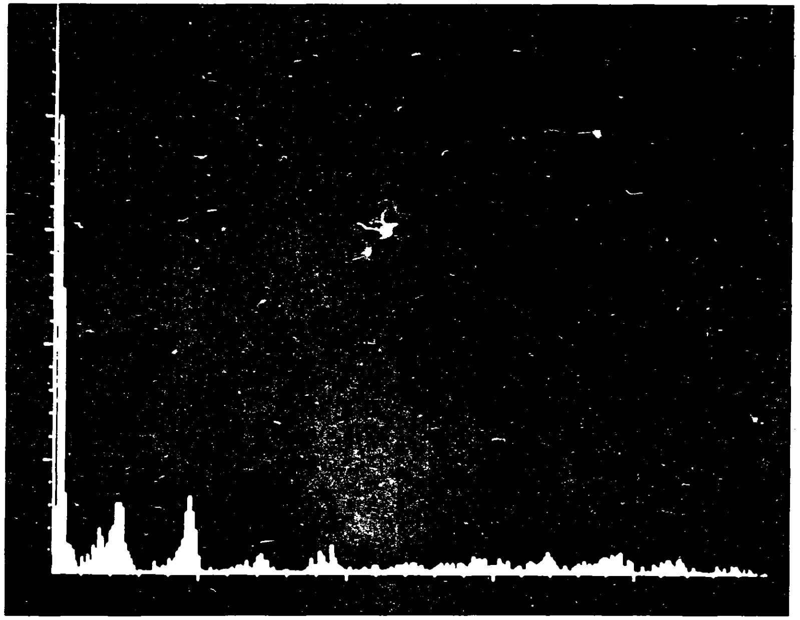


Fig. 3

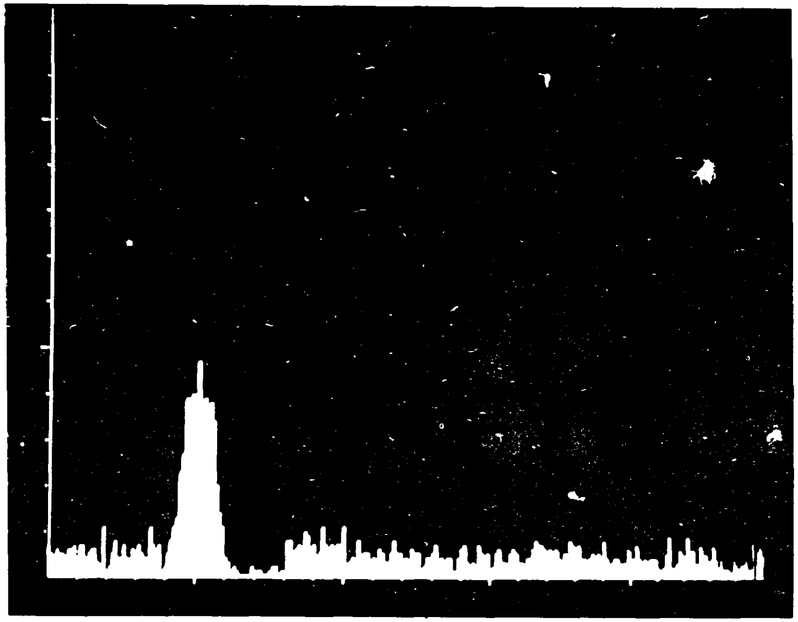


Fig. 4

RESPONSE DISCRIMINATION - 2 ALTERNATIVES (Sign Test)

1: Purpose: The program permits determining the Weber function

$$\frac{S_1 - S_2}{S_1} = f(S_1)$$ for stimulus evoked neural activity, on line with

the experiment. Pairs of stimuli of different intensity (S_1 and S_2) are applied in alternation; the single neuron responses to these stimuli (R_1 and R_2, respectively) are counted; and a non-parametric statistical test (Sign Test) is performed.

2: External circuitry and data input: $32_{(10)}$ different stimulus intensities, ranging in equal steps from 0 to 31 (31 = maximal intensity) are provided in external circuitry, and can be addressed from the relay registers. For each test series, S_1 (the larger base stimulus) and S_2 (the smaller test stimulus) are typed in from the keyboard. The stimuli of the selected intensities are applied in alternation, at a regular rate determined by external circuitry. At the beginning of each stimulus, a code signal is available.

3: Operation of this program: Mount data collection tape on unit #1. The "observation period" (i.e., the time after the code pulse during which responses will be counted) is set by RSW. The parameters for the test are typed in from keyboard in this order:

Identification Number of Test:
Intensity of stimulus S_1 and S_2:
A value for the response difference for which the Weber function will be evaluated (c):
The number of stimulus pairs to be applied:
A critical value of the test statistic at the desired confidence level.

4: Display of output statement: Depending on the outcome of the test statistic, the following output statements may occur after the desired number of stimulus pairs has been applied:

INCREASE S_2

DECREASE S_2

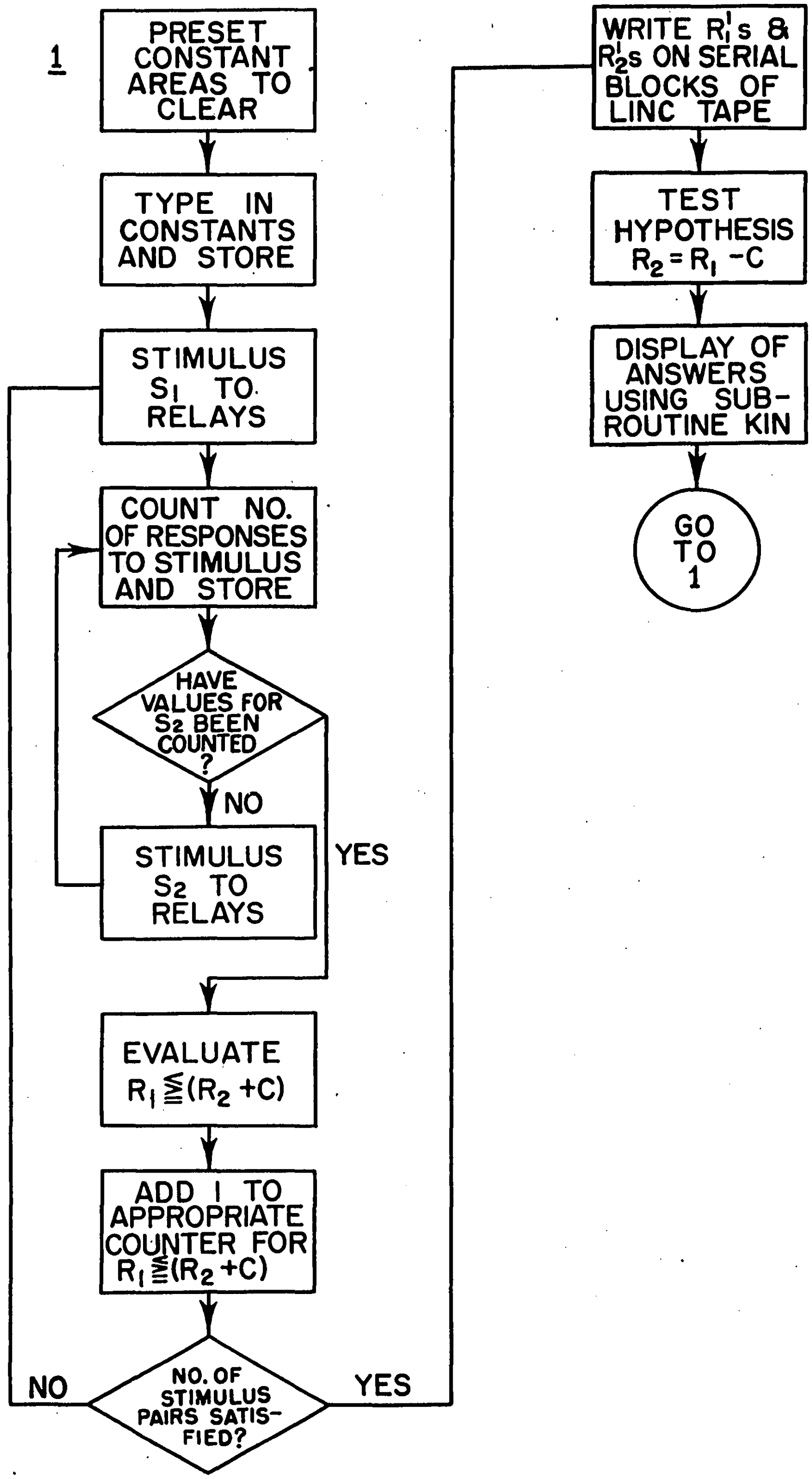
HYPOTHESIS CONFIRMED (in this case: $(R_1 - (R_2 + c)) = 0$), and the pair S_1 and S_2 define one point of the Weber function.

5: Tape storage of neural response counts: After application of the desired number of stimulus pairs, the neural response counts are stored on LINC MAG tape, serially on two blocks, first for S_1 and then for S_2. After octal to decimal conversion, these values can be punched out on paper tape.

6-35

6: Results obtained with this program: This program was used to determine the Weber functions for neural activity in certain first order, mechanoreceptive afferents. The objective was to test different assumptions concerning "c" (e.g.: constant additive response difference; constant fractional response difference), and to compare the Weber functions so obtained with that typical form of the Weber function that is descriptive of human touch discrimination. This condition was found to be best fulfilled when c = const. (see: G. Werner and V.B.Mountcastle, J. Neurophysiol., 1965, to appear in March or May issue.)

SIGN TEST - PROGRAM



RANDOM SELECTION OF STIMULUS INTENSITIES AND CATEGORIZATION OF SINGLE NEURON RESPONSE COUNTS

1: Purpose: This program is designed to obtain on line with the experimental preparation the data required to calculate the information transmission in a neural pathway: stimuli of a certain (optional) number of intensities are applied in a random sequence, and the single-neuron responses evoked by these stimuli are counted. The response counts are stored on LINC mag tape in such a way that their association with the respective stimulus intensities remains preserved on retrieval. On retrieval, the response counts can be grouped into categories of optional size. From the data saved on LINC mag tape, a matrix of joint occurrences of stimulus intensities S_i and response R_j can be obtained which permits evaluation of the numerical expression:

$$U(S,R) = -p(S,R) \log p(S,R)$$

where U(S, R) is a measure of "joint uncertainty". The marginal probabilities of S and R are used to determine the "maximum joint uncertainty" for the case that the R_j's and S_i's are orthogonal ($=U_{max}$). The difference $U_{max}-U(S,R)$ is a measure of the uncertainty reduction through information transmission; it is a function of the number of different stimulus categories chosen (Fig. 1). This program is used to collect and save data from experiments performed to determine this function for single neuron activyt. The main features of this program appear in the accompanying flow chart.

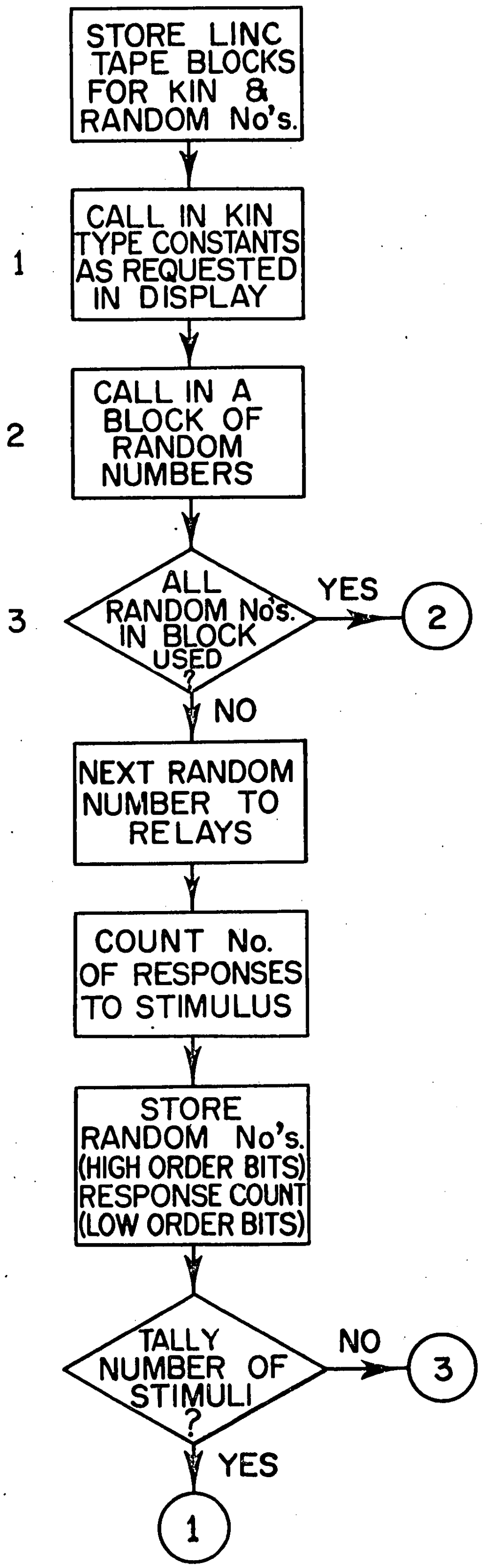
2: Prerequisite: Storage of 4 blk's of rectangularly distributed random numbers on Linc Mag tape.

3: Operation: See sign test program.

4: Use of the program: On start of program, a display appears in which the numberical variables are to be typed in from the keyboard. Tiis display appears as follows:

RANDOM STIMULUS-EPUT COUNT	(title)
RUN NO.	(type identification number, in decimals, followed by EOL)
BLKG.TIME	(time in msec after the code for stimulus onset)
SAM TIME	(Time after blkg time during which responses should be counted)
NO STIM	(total number of stimuli to be applied)
S_1	(this is the minimal stimulus intensity that should be applied, out of the available intensities-1 thru 3[illegible].

RANDOM STIMULUS - PROGRAM



STIMULUS CATEGORIES – maximal = 30

Fiber 25-1, Random Series
Stimulus - Response Matrix

Response Categories-Impulses Per Response

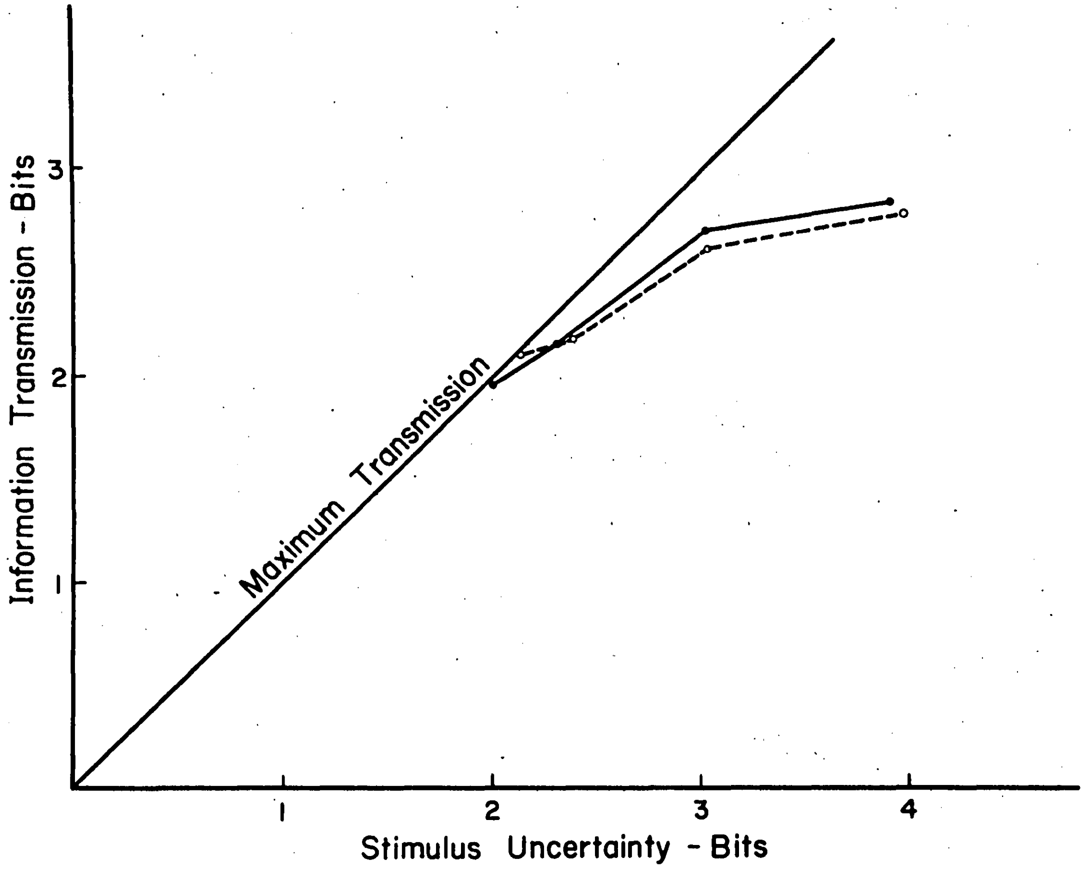
	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Σ ↓
0-1	29	15	3																									47
2-3	3	11	3																									17
4-5		5	2	1																								8
6-7			8	1																								9
8-9			6	7																								13
10-1			1	10	1																							12
12-3				7	7																							14
14-5				1	7		1																					9
16-7					4	10	3																					17
18-9						8	9	2	1		1		1															22
20-1						2	16	3	2	1	1																	25
22-3							5	8	7	7	11	6	4		1													49
24-5								3	9	6	4	5	9	2	4	2												44
26-7									4	3	5	7	5	8	6	2	5	4										49
28-9									1		1	2	8	2	2	5	6	9	4									40
30-1										1				5	6	7	4	6	7	5	7	3		1				52
32-3															2	4	2	6	7	11	9	7	8	5	1			62
34-5																	2	1	5	2	5	9	11	4	3	4	3	49
36-7																1				1	3	4	6	11	9	4	8	47
38-9																	1					1	5	7	6	7	4	31
40-1																							1	2	5	8	3	19
42-3																									1	3	1	5
44-5																									2	1	1	4
Σ →	32	31	23	27	19	20	34	16	24	18	23	20	27	17	21	21	20	26	23	19	24	24	31	30	27	27	20	644

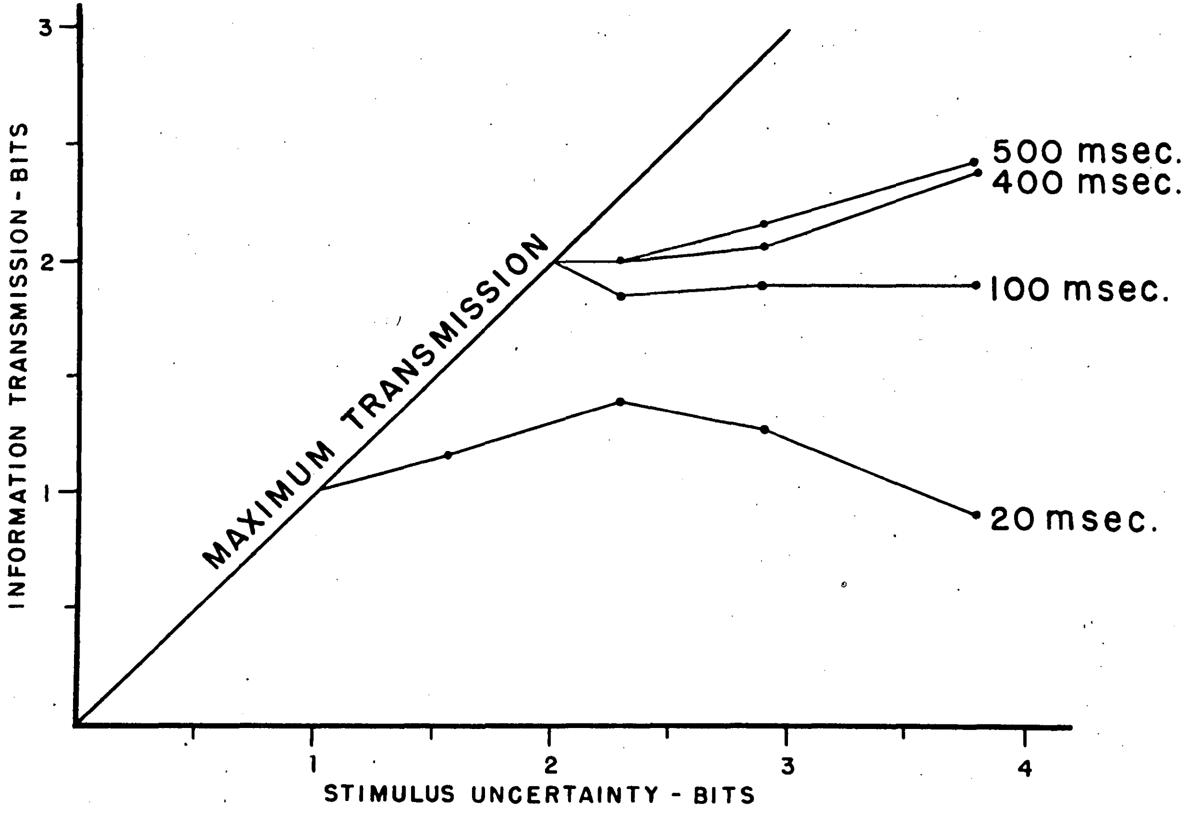
(1) $U(S) = -\sum p(S) \cdot \log \quad p(S)$ [Stimulus Uncertainty]

(2) $U_{ob.} \quad (SR) = -\sum p(SR) \cdot \log \quad p(SR)$ [Observed Joint Uncertainty]

(3) $U_{max.} \quad (SR) = -\sum p(SR) \cdot \log \quad p(SR)$ [Maximal Joint Uncertainty On Random Assumption]

(4) $U_c \, (SR) = U_{max.}(SR) - U_{ob.}(SR)$ [Contingent Uncertainty =Information Transmitted]


Information Transmission - Bits
Stimulus Uncertainty - Bits
Maximum Transmission
1
2
3
4
1
2
3


INFORMATION TRANSMISSION - BITS
STIMULUS UNCERTAINTY - BITS
MAXIMUM TRANSMISSION
500 msec.
400 msec.
100 msec.
20 msec.
1
2
3
4
1
2
3

Explanations: Blanking time (blkg.time) and Sam time define the observation period, that is that phase of the response over which the discharges are to be counted; it is referenced to stimulus onset.

Specification of S_1 allows the user to limit the range of the random numbers addressing the relay registers (e.g.: only stimuli above intensity 7 are admitted).

When the last value is typed, the program will store the first acceptable random number in the relay register, and write the identification block inblk. 1, unit 1. The program will then wait for sense switch 0 to be put in the "up" position; from there on, stimuli are applied at the rate determined by an external rate generator. When the total number of stimuli is applied, the stimulus response blocks are written. In each word, the first five (high order) bits contain the stimulus intensity, and the other seven (low order) bits contain the response count, with a maximum of 177_8.

5: <u>Ancillary programs:</u> Programs are available for perforation of the saved data on to paper tape. Figure 2 displays the print out of a stimulus-response matrix with the responses categorized in classes of two responses each.

6: <u>Results obtained:</u> From stimulus-response matrizes such as that of Figure 1 (top) information transfer as regards stimulus intensity was calculated for mechanoreceptive cutaneous afferent nerve fibers, using the computational procedure outlined in Figure 1 (bottom). For each nerve fiber, the computation was repeated for different sets of stimuli, with different values of stimulus uncertainty. An estimate of the maximal information transfer was thereby obtained (Figure 2 (top). In 14 different fibers, this value averaged to 2.5 bits. Maximal information transfer was found to vary with the "observation time" after stimulus onset, in a manner shown for one typical example in Figure 2 (bottom). These results are in the process of publication (G. Werner and V.B.Mountcastle, J. Neurophysiol., 1965).

b) ANALYSIS OF POPULATION ACTIVITY

Response averaging and amplitude and phase measurements of noisy signals

This program is used in a research project on the biophysics of photoreception. One of the least understood aspects of vision is the way that absorption of light by visual pigments causes membrane potential changes in the visual receptor cell. This project seeks to quantitate the dynamic relations between the amount of light illuminating the eye and the evoked visual cell potentials. These potentials are measured as a massed response of many cells, the retinal action potential, recorded from an intact spider in response to various modulations of light intensity with time. From the simple ocellus of the spider, it is thought that only electrical responses of primary visual receptor cells will be measured; thus, these potentials may be considered the initial electrical events whose elicitation by light is being studied.

The dynamic relations between modulations of light intensity and retinal potentials are most easily studied with small modulations (small signal approach), whereby amplitudes of the retinal potentials are linear functions of light modulation. In particular, it is possible to fit linear transfer functions to amplitudes and phases of retinal potentials elicited by sinusoidal light modulations. From these transfer functions, satisfactory predictions of transient (i.e., incremental flash) responses have been made. More generally, present work indicates that large, non-linear responses may be quantitatively predicted by a model whose parameters are, in part, denied from the linear transfer function.

It is for the averaging of small amplitude (10 microvolts, or so) linear or non-linear responses from much noise that this program was written. It permits measurement of responses to sinusoidal light modulations over frequency ranges longer than would otherwise be feasible and it permits accurate measurement of very small amplitude DC potentials, as in the responses to incremental steps of light, that were hitherto not possible. This program also allows the amplitudes and phases of these averaged responses to be determined, thereby making possible the transfer function analysis described above.

Program Description: The program occupies the four quarters of lower memory and is logically divided into four subroutines, portions of which may be skipped under sense switch control.

1. ARC (Average Response Computation) samples the response waveform at a preset rate while a Compute Gate (external level) is present, adds the sampled points of successive responses in double precision, and averages the preset number of responses in single precision.

2. ASC (Average Stimulus Computation) performs four functions:

a) It samples and averages in single precision a preset number of stimulus waveforms; this may be skipped by SNS 0.

b) It displays both the averaged and response and stimulus waveforms (see Figure 1).

c) Under control of SNS 5, it allows each display to be rotated (see Figure 2) and points added to or subtracted from the end of the (unrotated) display. All subsequent displays are affected.

d) It writes the averaged response and stimulus points; each occupying one block, on tape unit #1; the last word in each block contains the number of data points displayed on exit from (c) above.

3. RAP (Response Amplitude and Phase)

a) allows the amplitudes and zero-crossings of displayed responses and stimuli to be measured by displayed horizontal and vertical lines, respectively; moved under POT control. (see Figures 3 and 4).

b) it computes the peak-to-peak response amplitude and the phase of the (sinusoidal) response with respect to the stimulus; amplitude computation alone may be chosen by SNS 1.

4. DDAP (Decimal Display of Amplitude and Phase) computes the decimal values of amplitude and phase computed by RAP and displays their values (see Figure 5).

Legends for Figures:

Figurel:	Display by ASC of averaged stimulus and response at 20 cps. The upper trace is the stimulus. Number of responses averaged = 256; number of stimuli averaged = 8.
Figure 2:	Rotated averaged response displayed by ASC. Vertical scale double that in Fig. 1. Beginning and end of response do not meet, as the discontinuity near the center shows; this is presumably due jitter in the length of the compute gate.
Figure 3:	Display by RAP of averaged response and a line set by POT 2 to be at its positive peak. Had this peak been found by the computer, the isolated point above the time peak would have been chosen.
Figure 4:	Display by RAP of the averaged response, its zero value (horizontal line), and a vertical line controlled by POT 3 placed at the positive-going zero crossing.
Figure 5:	Display by DDAP of amplitude and phase computed by RAP.

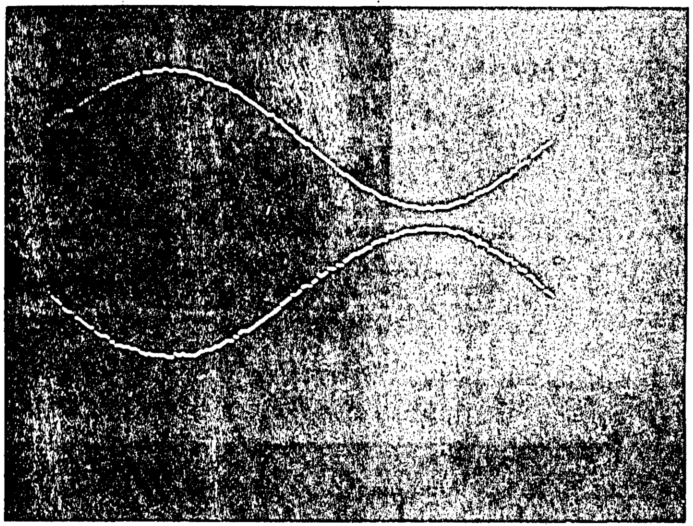


Fig. 1

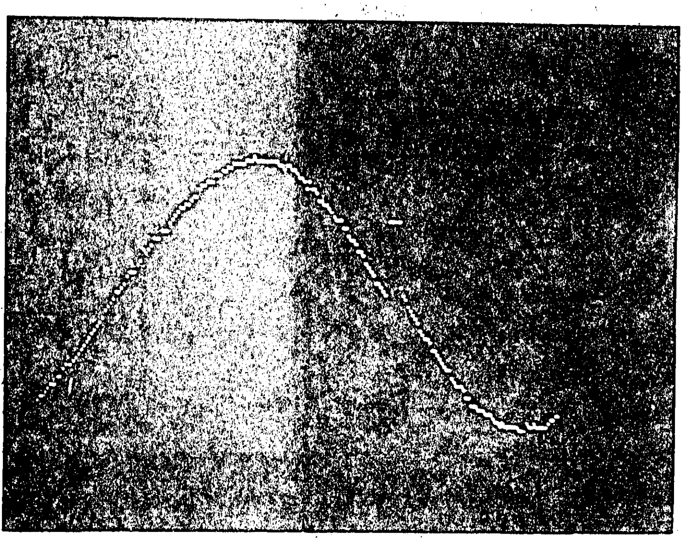


Fig. 2

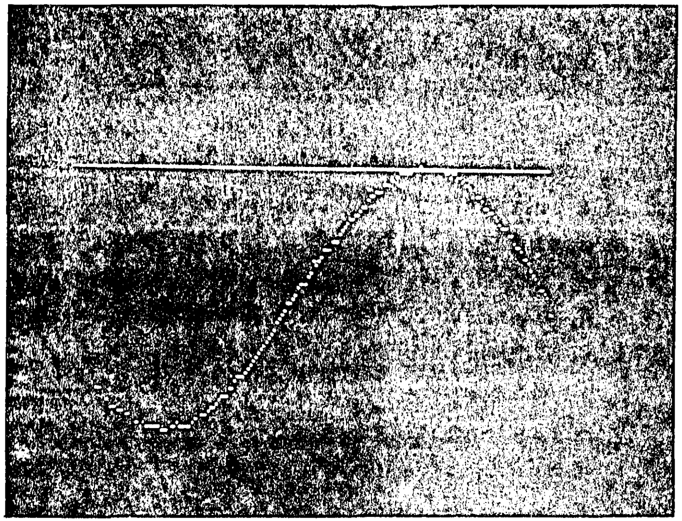


Fig. 3

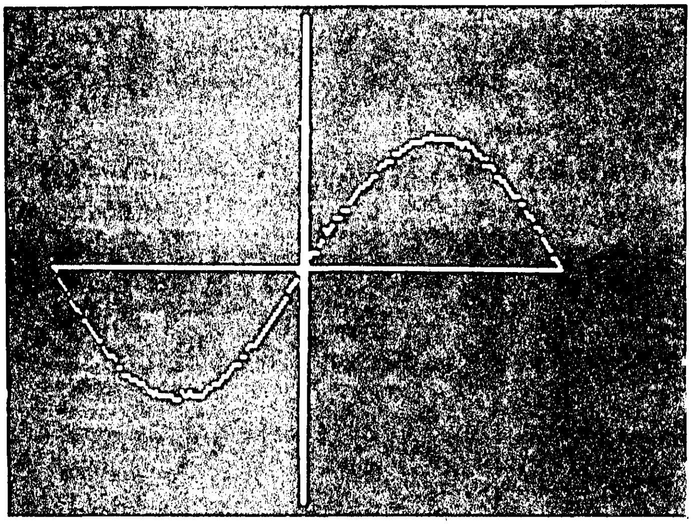
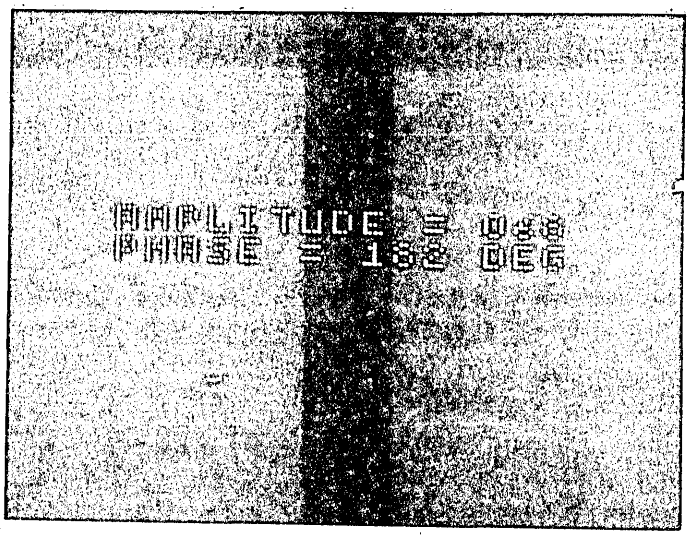


Fig. 4



AMPLITUDE = 0[illegible]8
PHASE = 182 DEG

Fig. 5

6-46

3. COMPUTER PERFORMANCE

All modifications to the computer which have been recommended in CDO equipment change bulletins #1 through #11 have been made and checked. The installation of the upper half of memory was accomplished with no particular difficulties in August, 1964. With the exception of the sense amplifier failure noted below the memory system has been working most satisfactorily.

Maintenance

Only two major maintenance problems have occurred during the entire time we have had the Linc. In each case the trouble was localized quickly.

On March 3, 1964, a transistor of the bit-9 flip-flop (DEC 4205) of the B-register failed. The card was replaced.

On October 12, 1964, bit-2 sense amplifier (DEC 1571) failed. The cause of this failure is unknown to us. The card was promptly repaired by DEC and reinstalled.

Microtapes units have given us quite satisfactory service. Initial difficulties with the tension of the magnetic tape were greatly reduced by adjustment of the drive belts of the motors.

Occasional difficulties with LAP3, particularly with insert and remove meta commands disappeared when the first recommended change of 4221 DEC packages was made.

Sporadic difficulties with the Keyboard input, in particular misreading of the 'tag' key, occurred in September, 1964. They disappeared spontaneously before becoming a serious problem.

General Comments:

We have no major suggestions to make concerning hardware changes in the Linc. We feel that the instrument as it stands is an extemely able laboratory computer in almost every respect. Our use of the computer has been so greatly expanded by the addition of upper memory that we now consider the 2 K memory to be essential. Further increase in memory size would probably be useful, especially if it were to make available additional locations for program storage.

Our few comments on the limitations of the Linc refer mainly to software. They are discussed below. One member of our group does, however, feel that it would be desirable to extend, if possible, the convenience of programming the scope display to conventional analog X-Y plotters. He feels that for routine graphic display the cost of the Calcomp digital plotter is excessive and the effort of programming for it might be objectionable. He feels also that it would be desirable to make use of the X-Y plotters often found in biological laboratories.

In our experience investigators with computational inclinations pick up Linc programming quickly and with a minimum of personal instruction. However, we believe that some didactic material on the art of programming the Linc would be useful. Of particular use would be a manual describing the function and use of Linc operation codes including illustrative annotated programs for some of the more sophisticated instructions.

In spite of this relative ease of programming the Linc we share a common experience of finding that a great amount of our time is spent in programming and otherwise caring for the computer. Because of the pressures of other considerations during our experiments, we must plan our programs so that they may be easily used by any member of our group. Therefore initialization procedures must be straight forward and, as much as possible, self-checking. Writing programs to fit this set of needs is more time consuming than writing programs for oneself alone. In the past year we have been able to devote time to producing several large and fundamental programs because construction work in the laboratory has reduced somewhat the numberof experiments we could carry out. With these as the principal working programs we hope to be able to use available time to write shorter special purpose programs which will add diversity to our Linc operation.

We have found the two major programs that CDO provided us with (LAP and GUIDE) to be very useful. The further development of these programs along the lines that have been discussed with us should make them even more useful. It does not seem to us that there is any particular need for the development of additional complete programs, because, by the very nature of the computer, the most useful programs are those written to meet specific needs of one's own experiments. It would be helpful, however, if some general purpose subroutines could be developed, in particular a library of arithmetic subroutines. This recommendation is based in part on the following reaction to the Linc made by one of the members of our group:

> For the purpose for which I had intended to use the Linc, namely the solution of statistical decision problems on line with the experiment, I found the following difficulties and limitations:
>
> 1) The Linc's limited memory and slow speed made it impossible to solve the problems in the form of conventional algorithms in the time available.
>
> 2) I am not sufficiently inventive to replace the conventional algorithms with approximation procedures which can be carried out in the available time.

3) I believe that most attempts to replace conventional algorithms with procedures in non-parametric statistics limit the scope of the application and curtail severely the precision of the results.

In retrospect, we feel that it would have been more efficient and helpful to us if we had had available certain general purpose subroutines that we had to write for ourselves. These are not necessarily difficult routines, but they are basic and widely used. They include: octal to decimal and decimal to octal conversion routines, alphanumeric oscilloscope display, and keyboard to teletypewriter alphanumeric translation. It has been our experience with subroutines we ourselves have written that programming time is greatly reduced when adequate subroutines are available.

4. THE EVALUATION PROGRAM

a. Initial training: The assembly phase of the Linc was an invaluable experience for the two participants. Firstly, it led to a rather intimate acquaintance with the technical aspects of the machine, its design, and the facts pertinent to check out and maintenance. Secondly, and perhaps more importantly, it provided the opportunity for intensive personal contact with the designer of the instrument This greatly helped to give the participants a frame of mind and a way of thinking about machine computation and data processing which was previously considerably less vivid and specific in their minds. This interaction with the personnel of the CDO fostered the development of ideas of ways in which computational methods, in general, could be used in research, and of ways for their implmentation in the laboratory. To that extent the initial assembly phase not only reflected on the Linc alone, but more generally, on the use of computers in biological research. Thus this program served a wider purpose than merely setting the stage for the use of the Linc per se. This initial experience of the two participants in the assembly phase was transmitted to other investigators in this laboratory, after the installation of the Linc, with consequences similar to those described above. Some of the contagious enthusiasm of the designers of the Linc was thus imparted on a larger group, and with this a wider group of investigators was drawn into the fascination for machine computational methods in physiological research. It is our opinion that a program similar to that conducted during the Summer of 1963 in Cambridge is to be recommended for any biologist who wishes to use laboratory computer.

b. Concerning the period from installation of the Linc at home base to date: At times we felt that during this period some advice and better communication with the CDO would have been very useful. In particular it was felt that some information service concerning programs of other groups would have been advantageous. Or else, some period distribution of more advanced examples in solving programming problems that should have served as a follow-up teaching device. The idea here is that of a kind of "correspondence course" to keep the participants alert of certain possible, typical "tricks" and general routines. This, we believe, would have saved us considerable "development time" which, valuable as it has been for better understanding of the machine and its applications, had to be detracted from the time we devoted to physiological experimentation. For these reasons we think it would be profitable if in the future, brief adjournment meetings were organized for Linc users. These meetings, to be held no more often than once a year, would give the opportunity for exchange of programs and ideas about laboratory application of the instrument and help greatly to reduce duplications that are costly both in money and time.

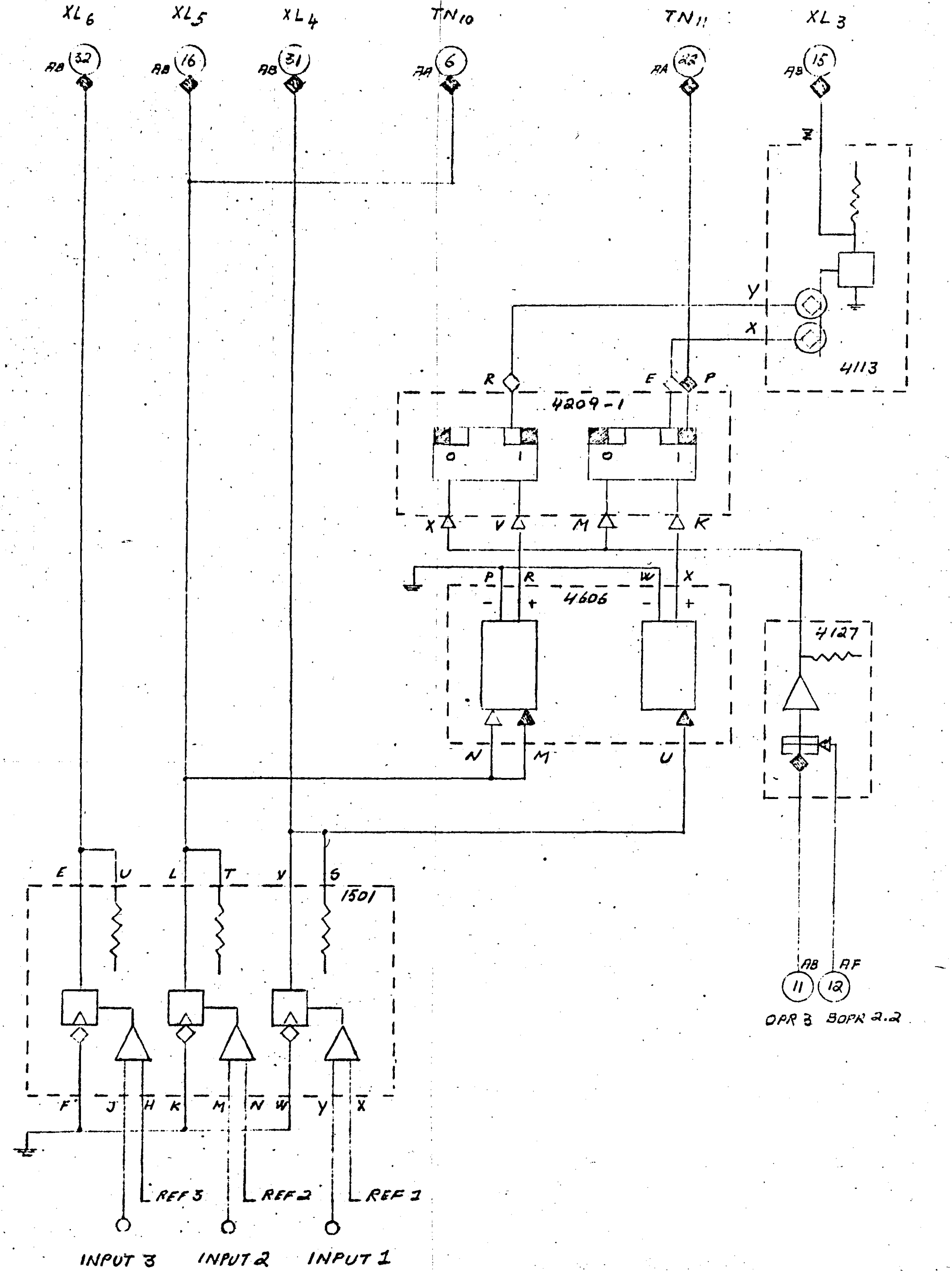
Publications of research in which the LINC was used:

1) Gerard Werner and Vernon B. Mountcastle: Neural activity in cutaneous mechanoreceptor afferents: Weber function and information transfer. J. Neurophysiol., (in press)

BY ______ DATE 9/10/64 SUBJECT INPUT P.I.U. SHEET NO. 1 OF 3

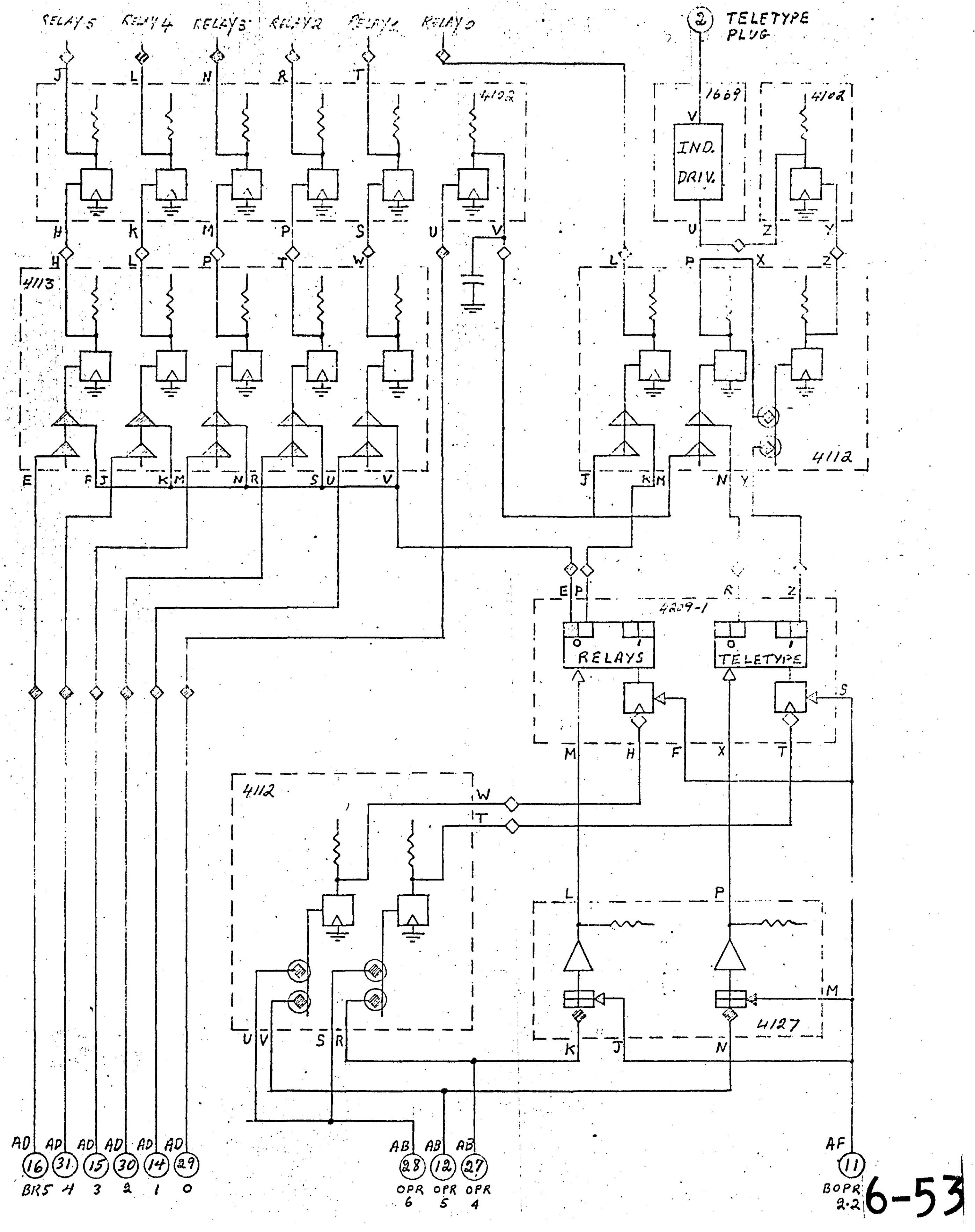
CHKD. BY ______ DATE ______ External Level Inputs JOB NO. 1

Event Sense Flip-Flops



BY WT DATE 7/26/64 SUBJECT INPUT P.I.U. SHEET NO. 2 OF 2

CHKD. BY ______ DATE ______ Relays and Teletype Selector Flip-flops gates and driver. JOB NO. 2



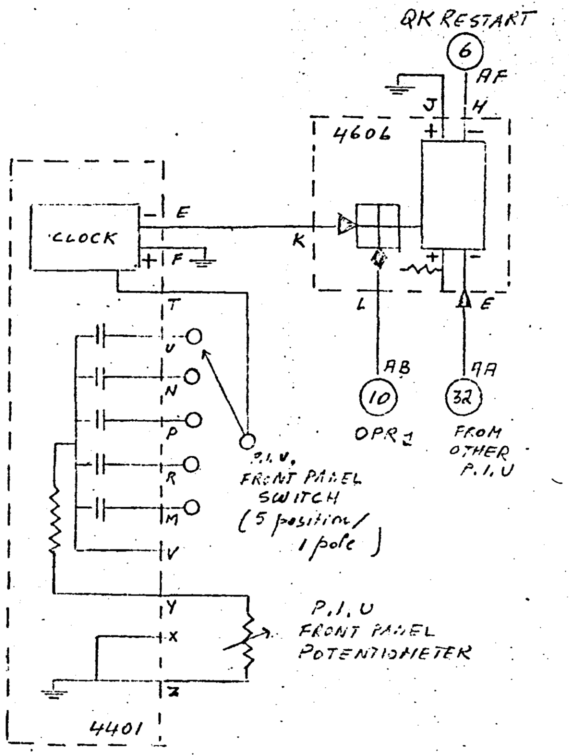
BY............ DATE 5/31/67

CHKD. BY............ DATE............

SUBJECT INPUT P.I.U.
External Clock - Quick Restart
DEC Cards lay-out

SHEET NO. 3 OF 3

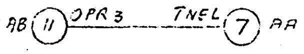
JOB NO. 3

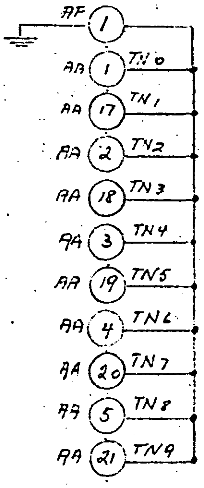


REAR

4113	A11
4102	A10
4112	A9
4127	A8
4209-2	A7
4209-1	A6
4401	A5
1501	A4
1669	A3
	A2
	A1

FRONT





BY .... DATE 10/23/64

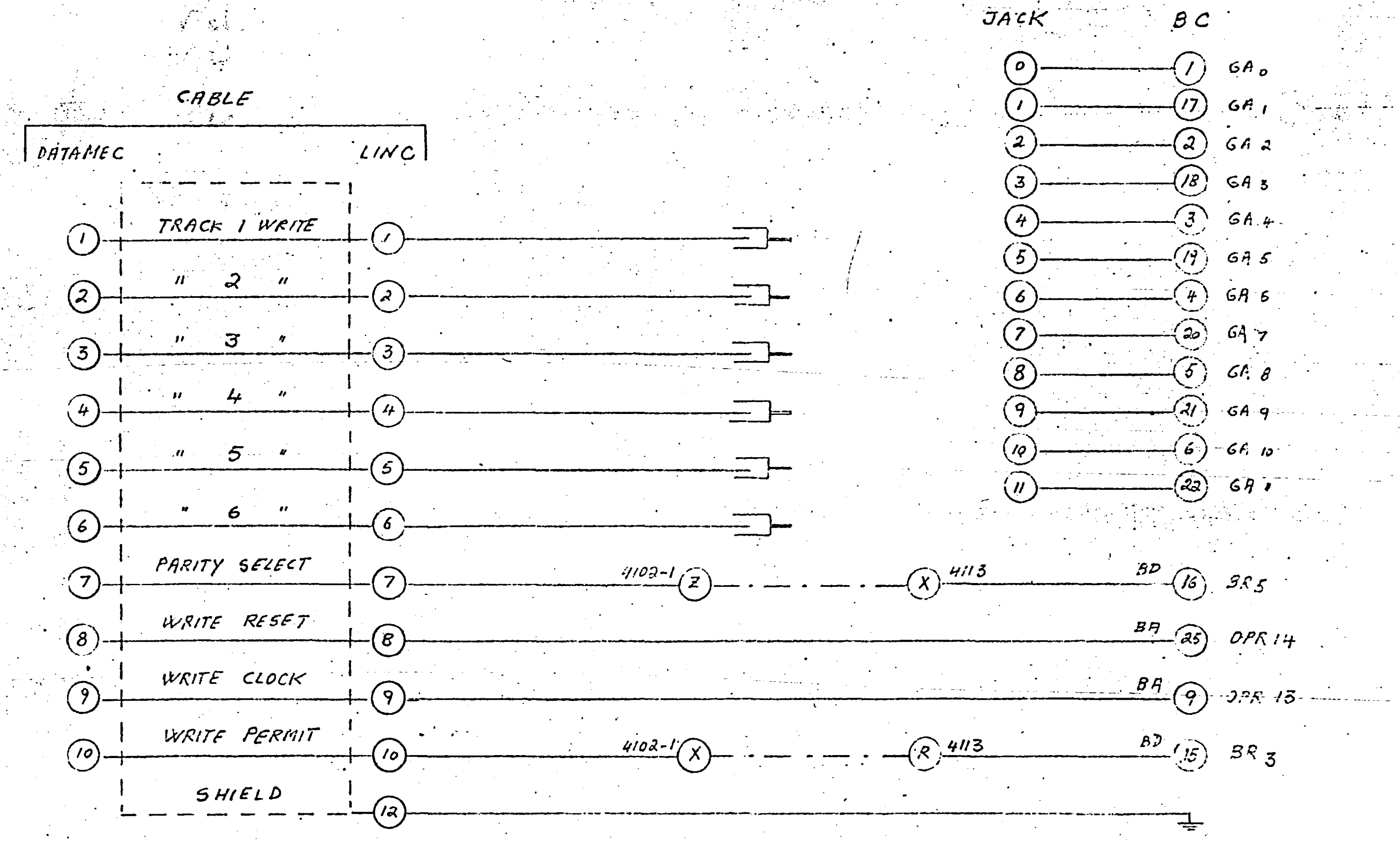
CHKD. BY .... DATE ....

SUBJECT DATAMEC P.I.U.

WRITE System - Datamec/Linc connections

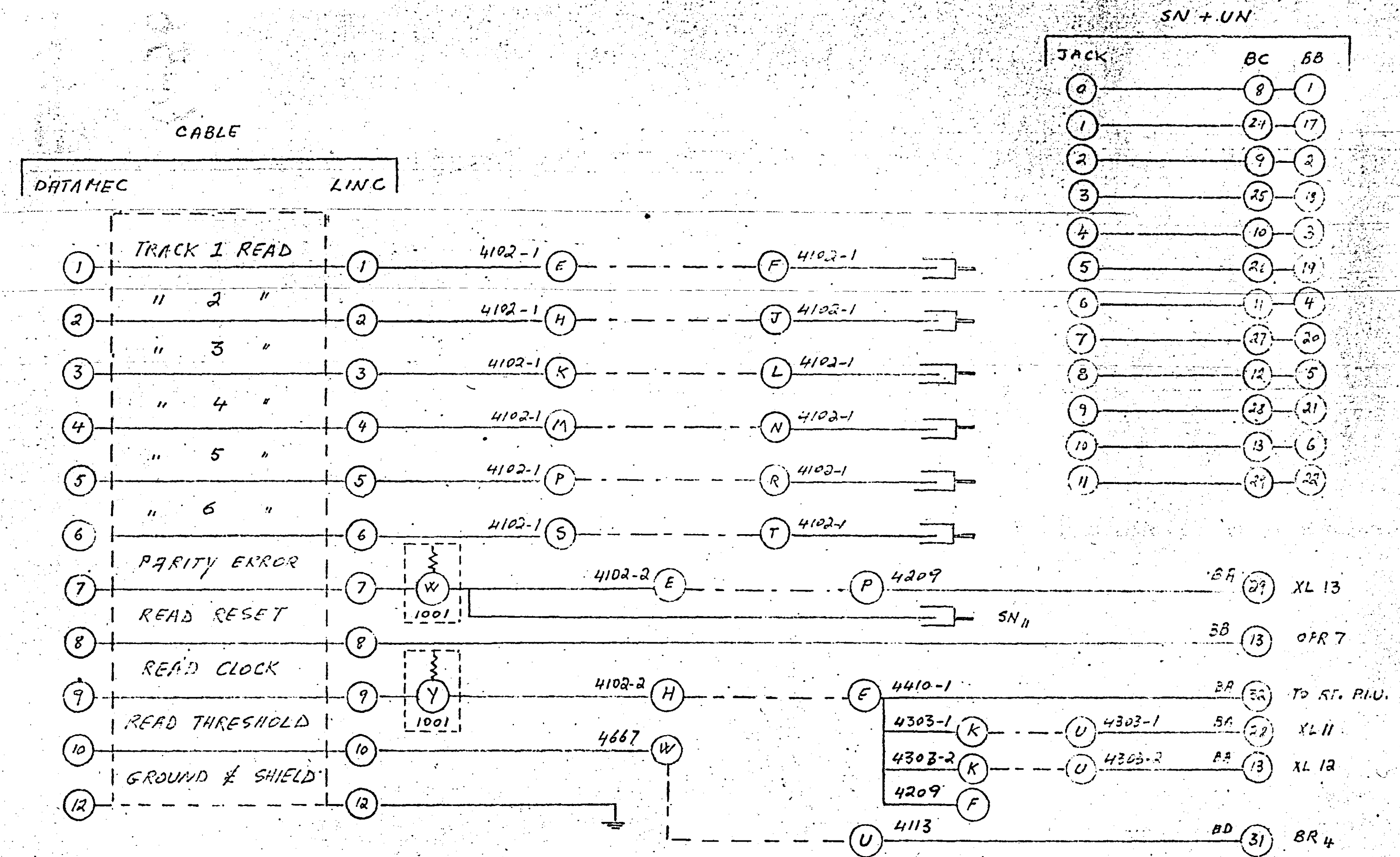
SHEET NO. 1 OF 6

JOB NO. 4



DATAMEC J6 PLUG = E. Co RF 11224

LINC WRITE PLUG = AMPHENOL 26-4501-16 S



DATAMEC J7 PLUG = ELCO RM 11224

LINC READ PLUG = AMPHENOL 26-4501-16 S

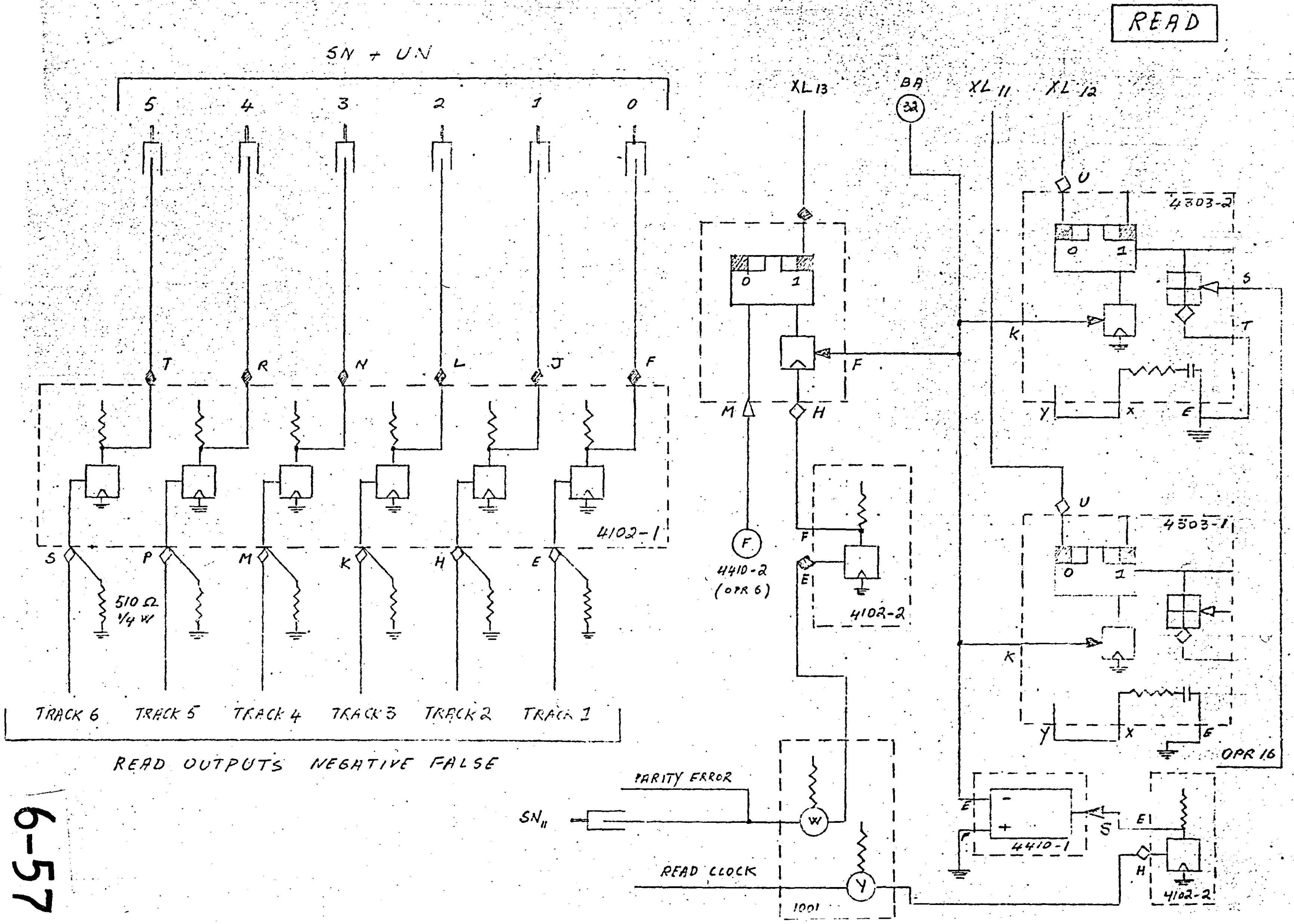
BY........ DATE........

CHKD. BY........ DATE........

SUBJECT DATAMEC P.I.U.
READ System - Datamec/Linc connections

SHEET NO. 2 OF 6

JOB NO. 5

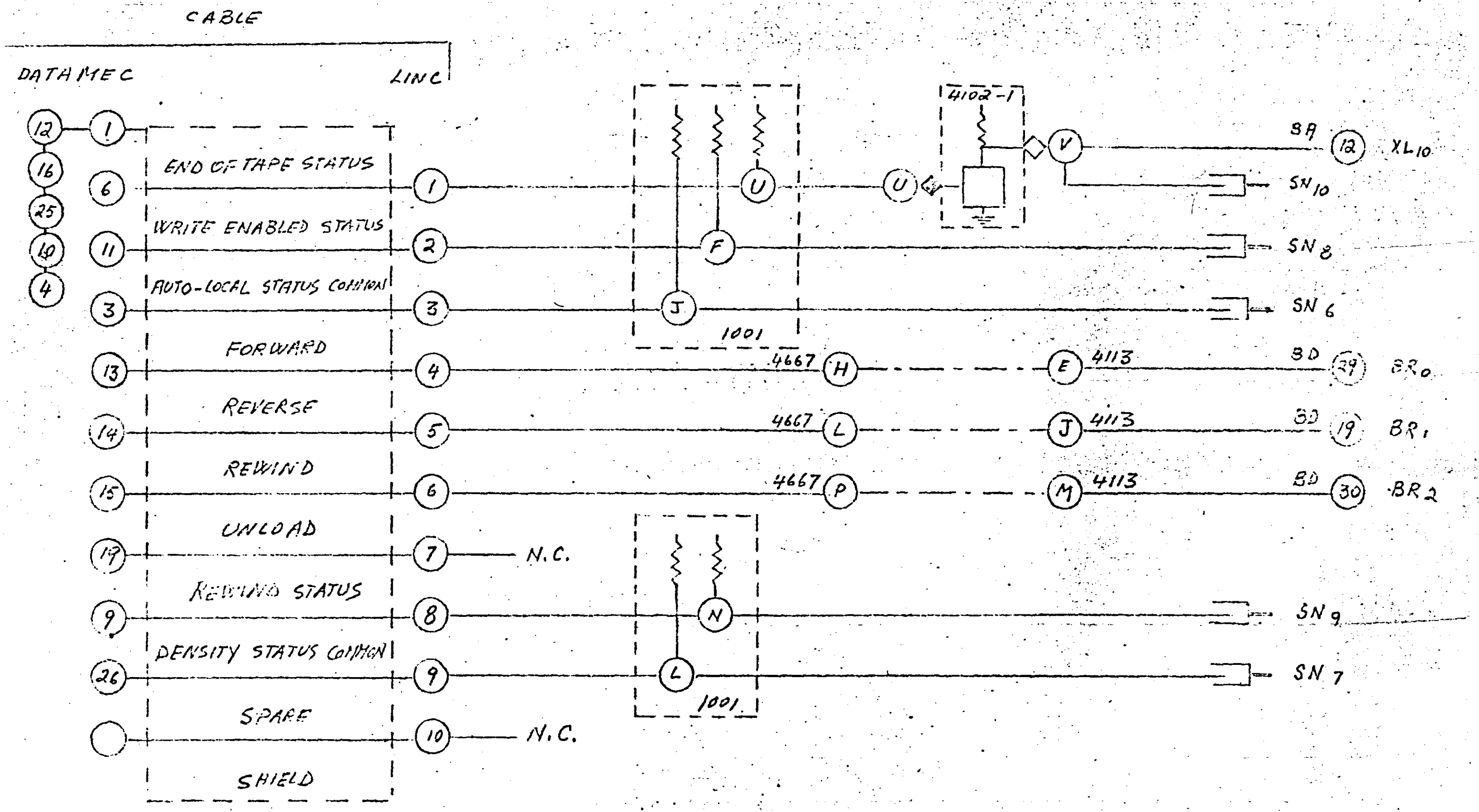


BY.............. DATE 10/29/62
CHKD. BY........ DATE..........

SUBJECT DATAMEC P.I.U.
READ System logic

SHEET NO. 3 OF 6
JOB NO. 6

6-57



DATAMEC J2 PLUG = ELCO RF 22824

LINC CONTROL PLUG = AMPHENOL 26-4501-16S

BY.......... DATE [illegible]

CHKD. BY...... DATE..........

SUBJECT DATAMEC P.I.U.

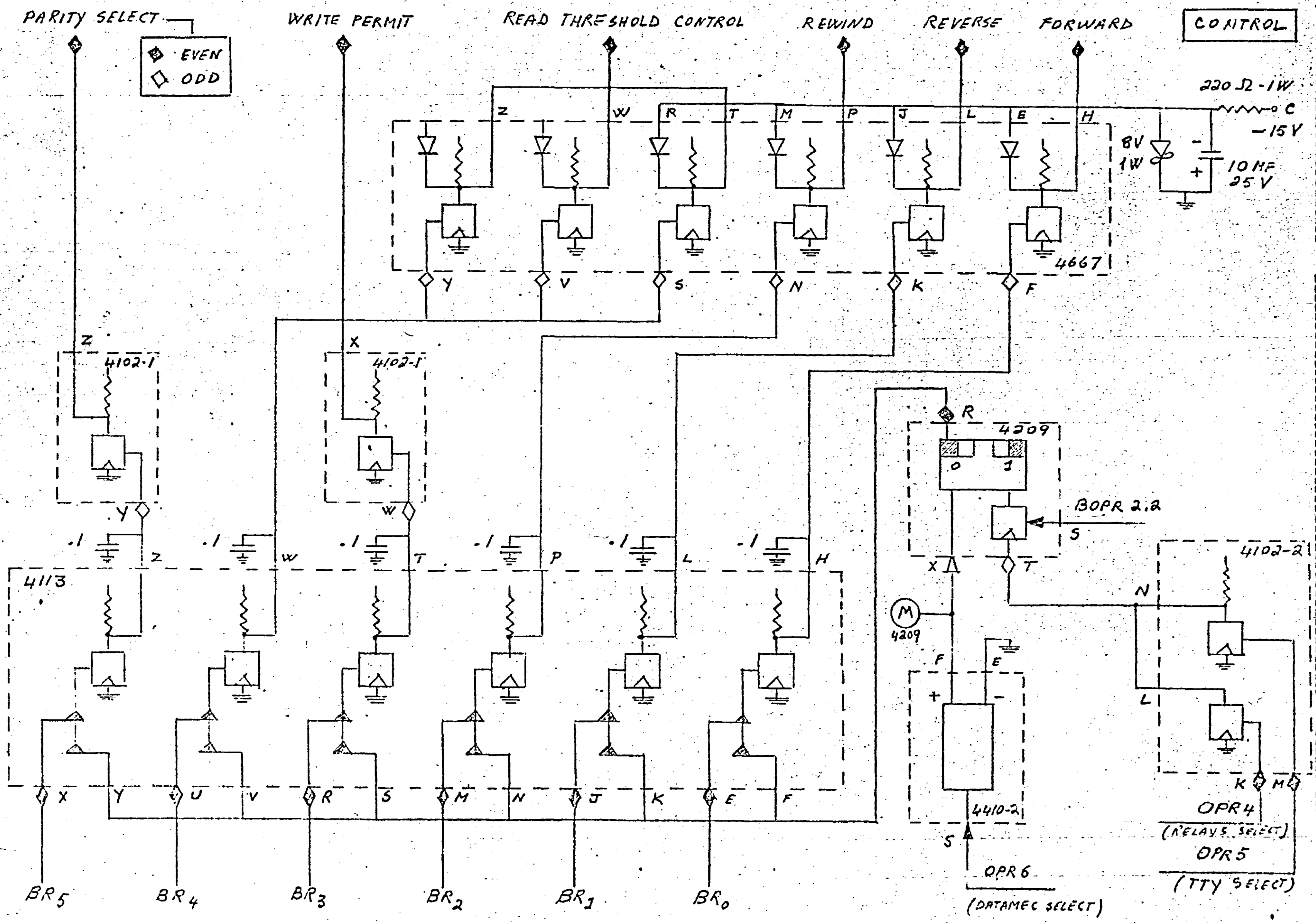
Tape CONTROL System

Datamec/Linc connections

SHEET NO. 7 OF 8

JOB NO. 7

BY____ DATE 10/20/64
CHKD. BY____ DATE____
SUBJECT DATAMEC P.I.U.
CONTROL System Logic
Datamec Selector Flip-flop
SHEET NO.____ OF____
JOB NO.____ 8



BY .... DATE ....

CHKD. BY .... DATE ....

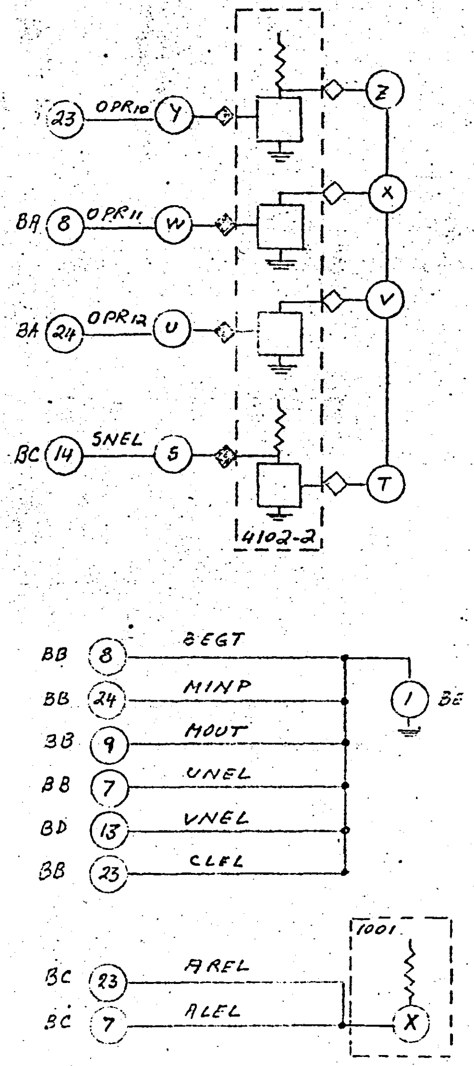
SUBJECT DATAMEC P.I.U.

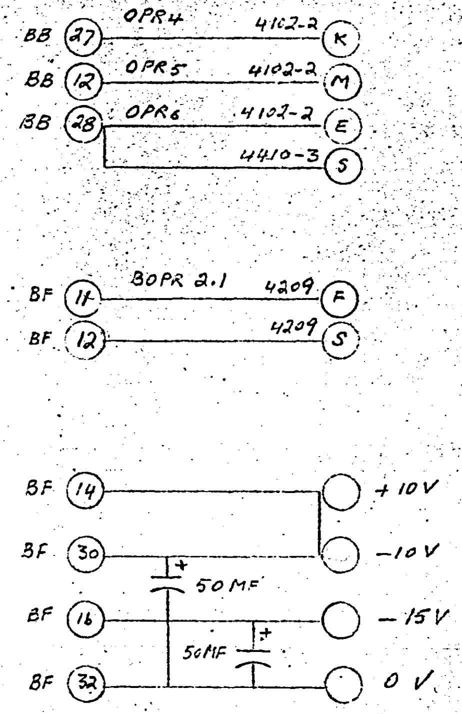
Miscellaneous connections

DEC Cards lay-out

SHEET NO. .... OF ....

JOB NO. 9





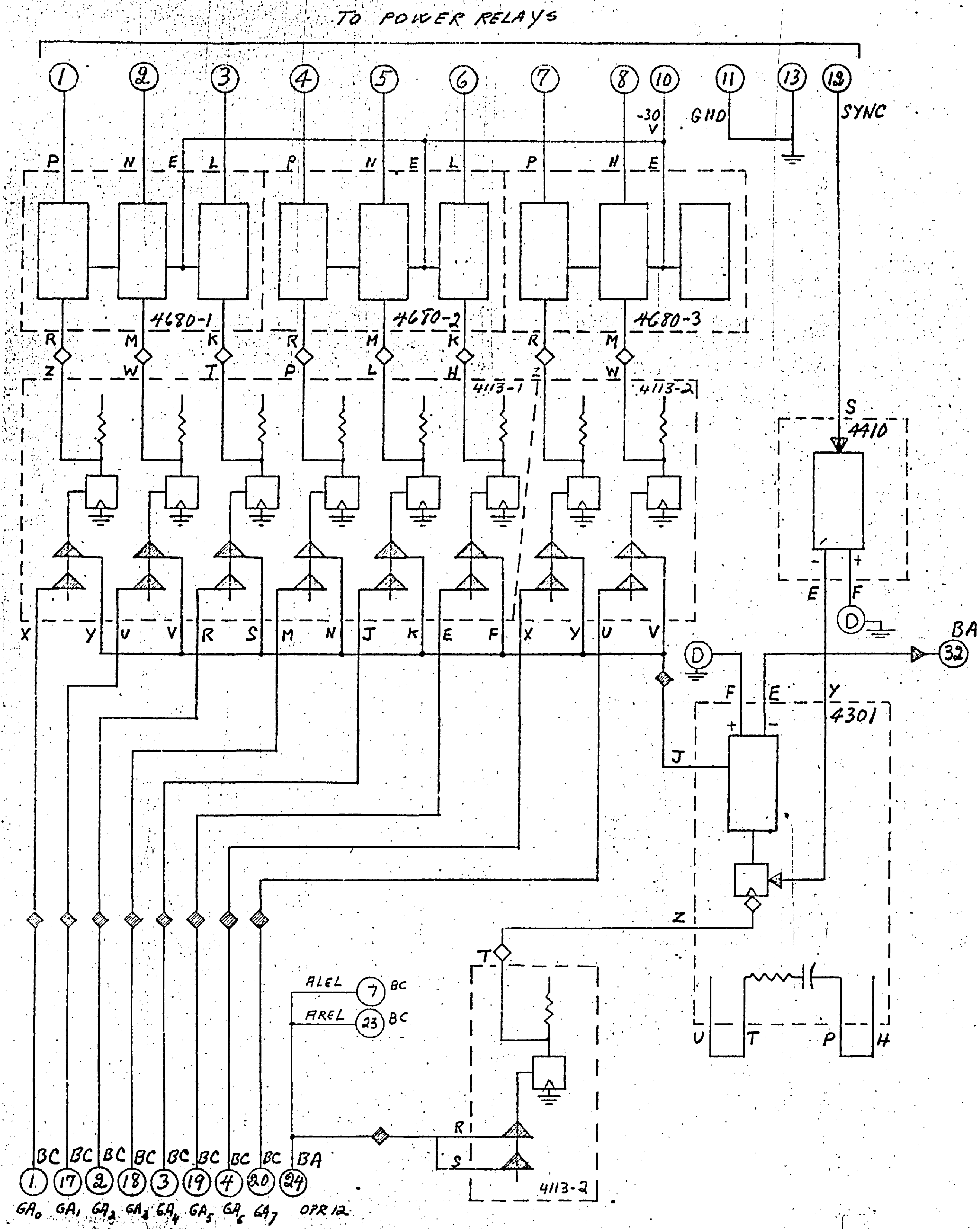
REAR

4102-2	C11
4410-3	C10
4209	C9
4303-2	C8
4303-1	C7
4410-2	C6
4410-1	C5
4113	C4
4667	C3
4102-1	C2
1001	C1

FRONT

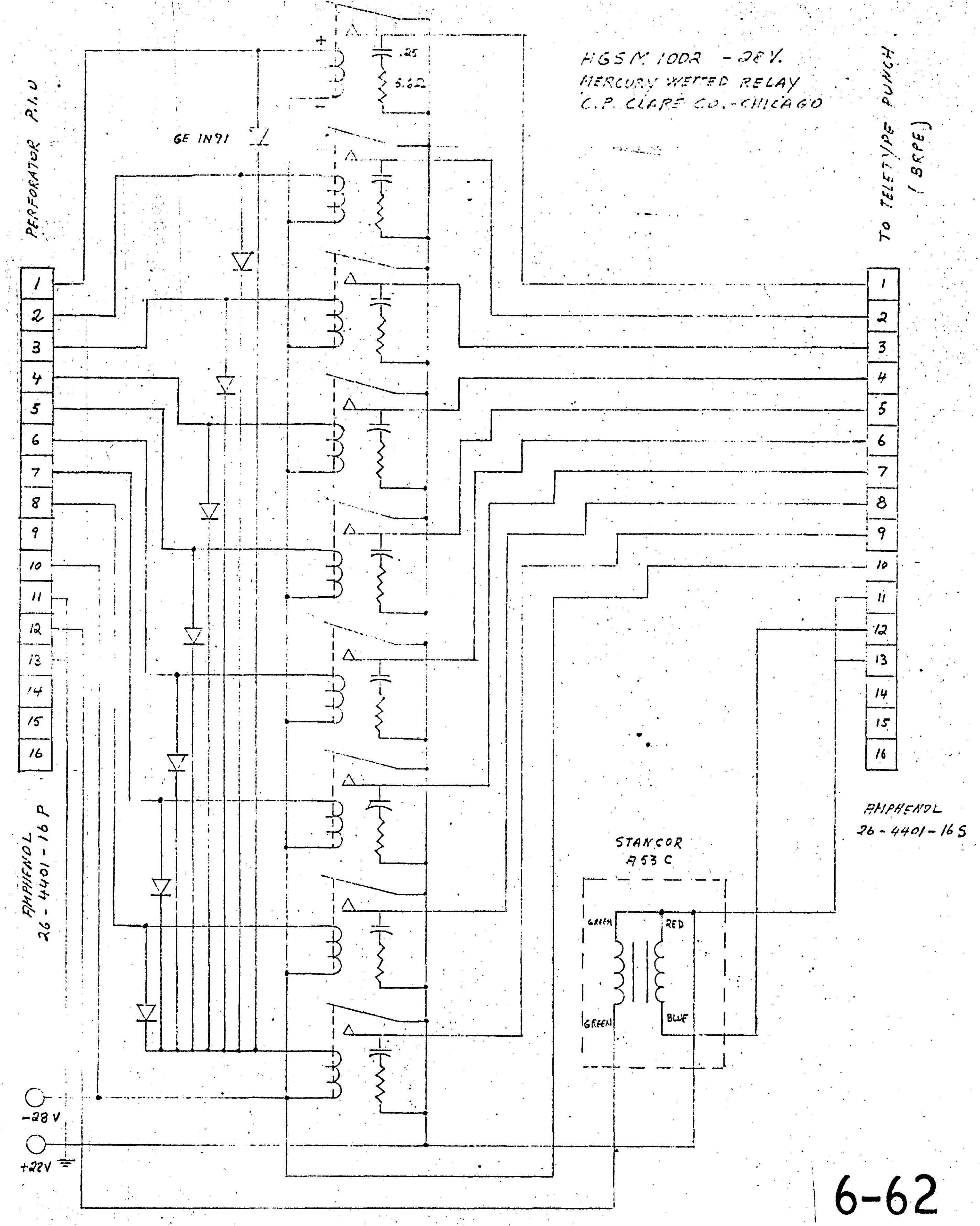
BY WT DATE 8/31/64 SUBJECT PERFORATOR P.I.U. SHEET NO. 1 OF 2

CHKD. BY DATE (to be used with INPUT P.I.U.) JOB NO. 10



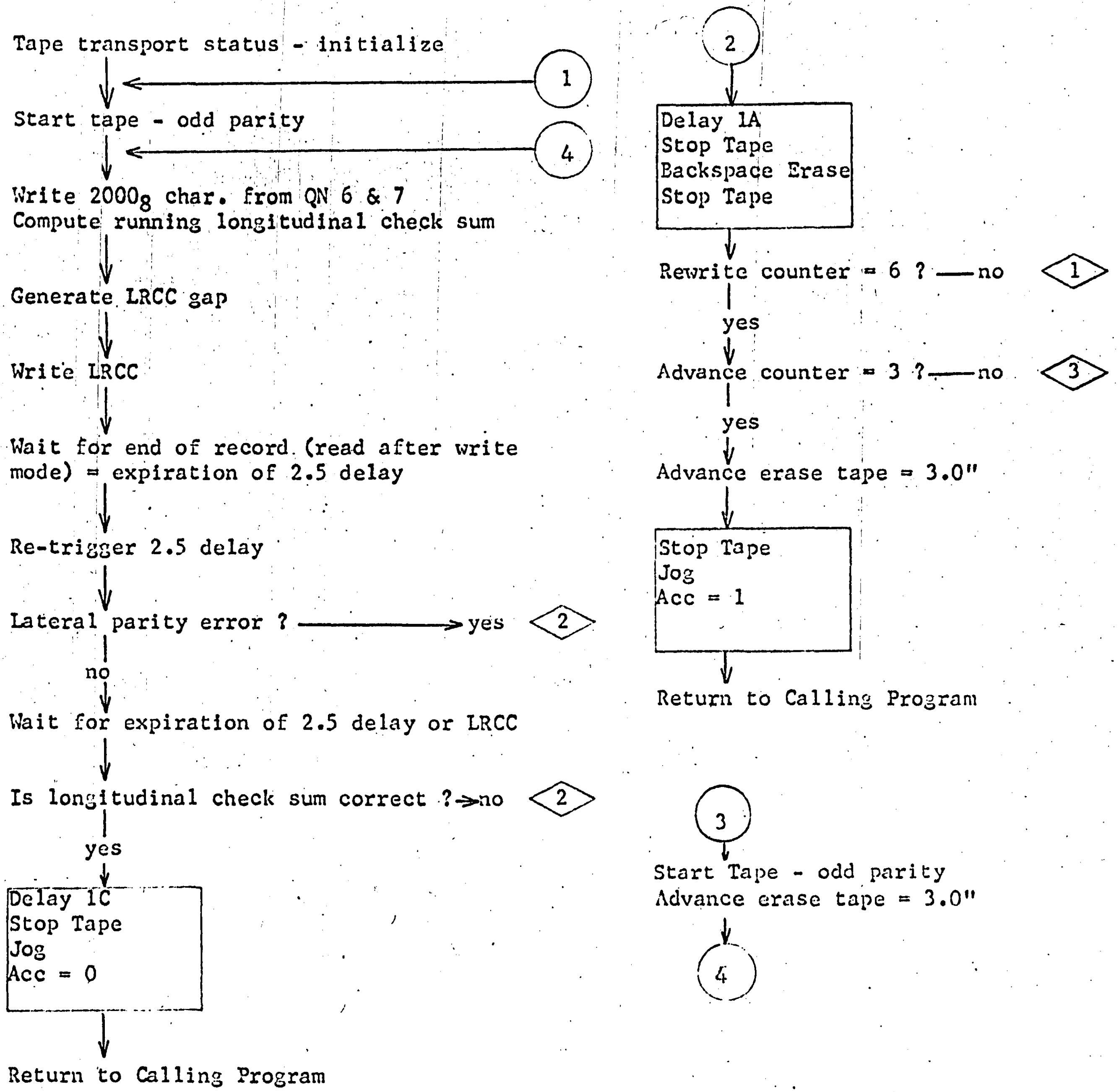
BY ____ DATE 6/3/64 SUBJECT PERFORATOR System
CHKD. BY ____ DATE ____ Power Relays - Connection Perforator
P.I.U. to Teletype Paper Tape Punch

SHEET NO. 2 OF 2
JOB NO. 11



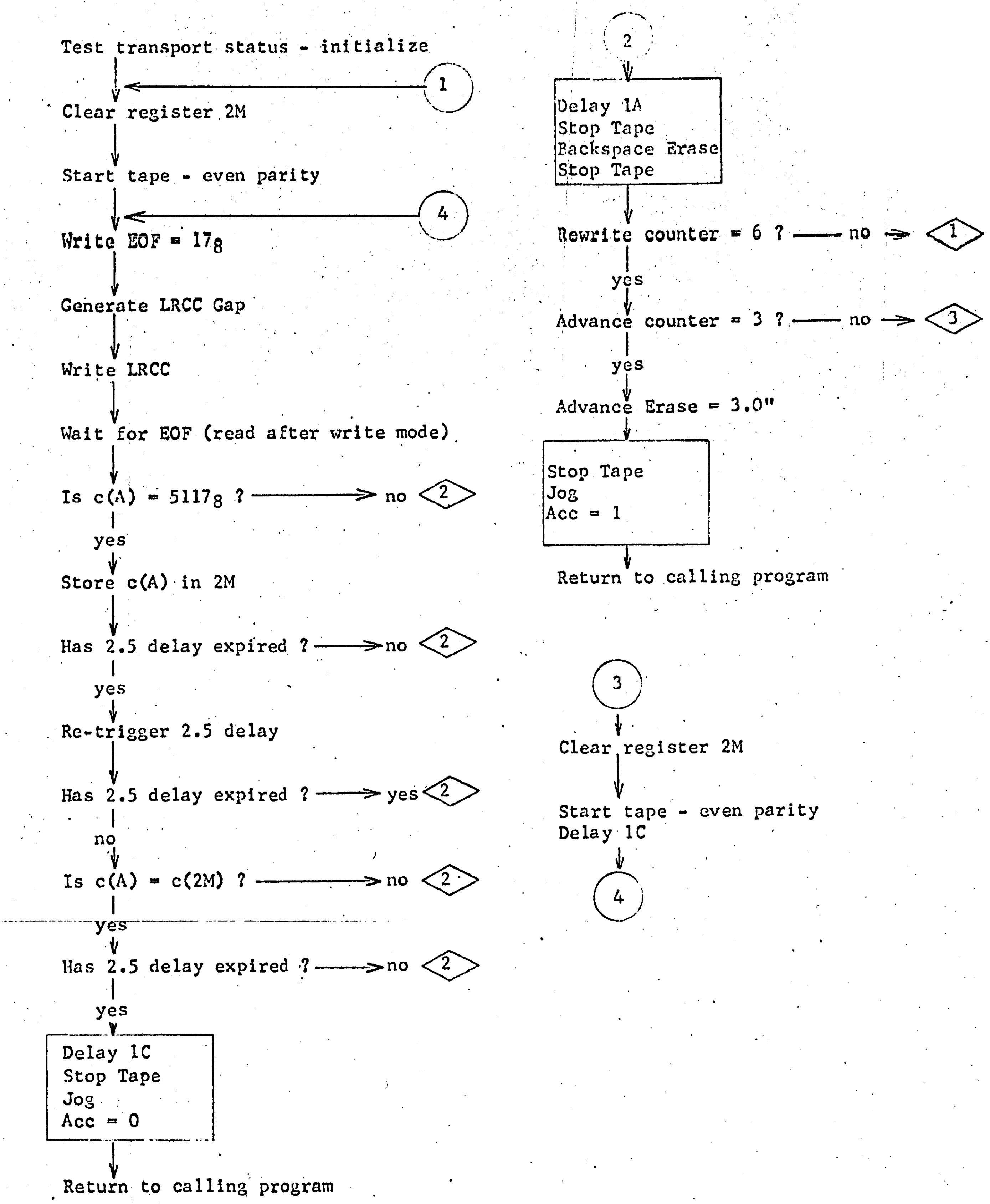
Flow chart #1

WRITE RECORD (2000_8 char.) : Initial Tag 2B

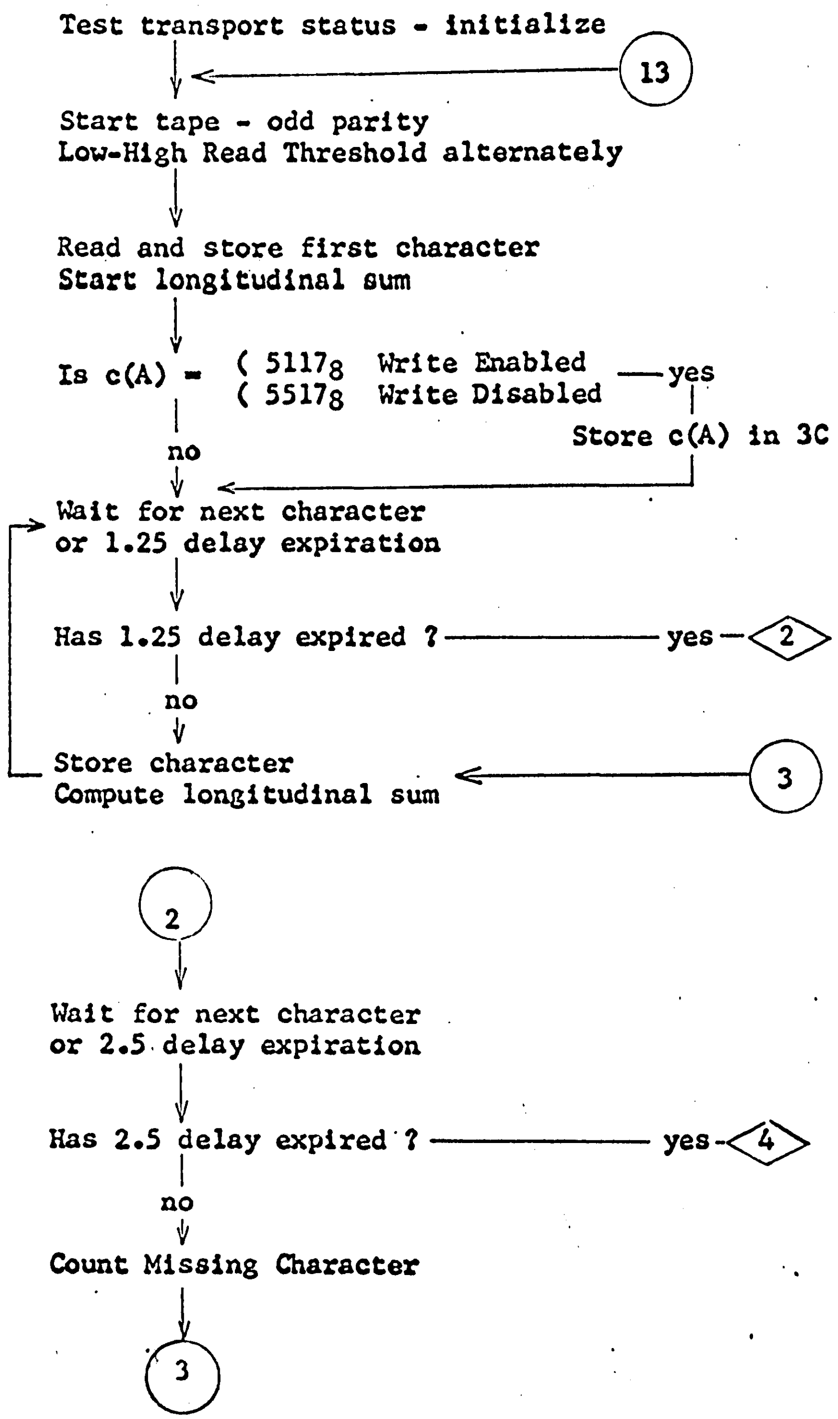


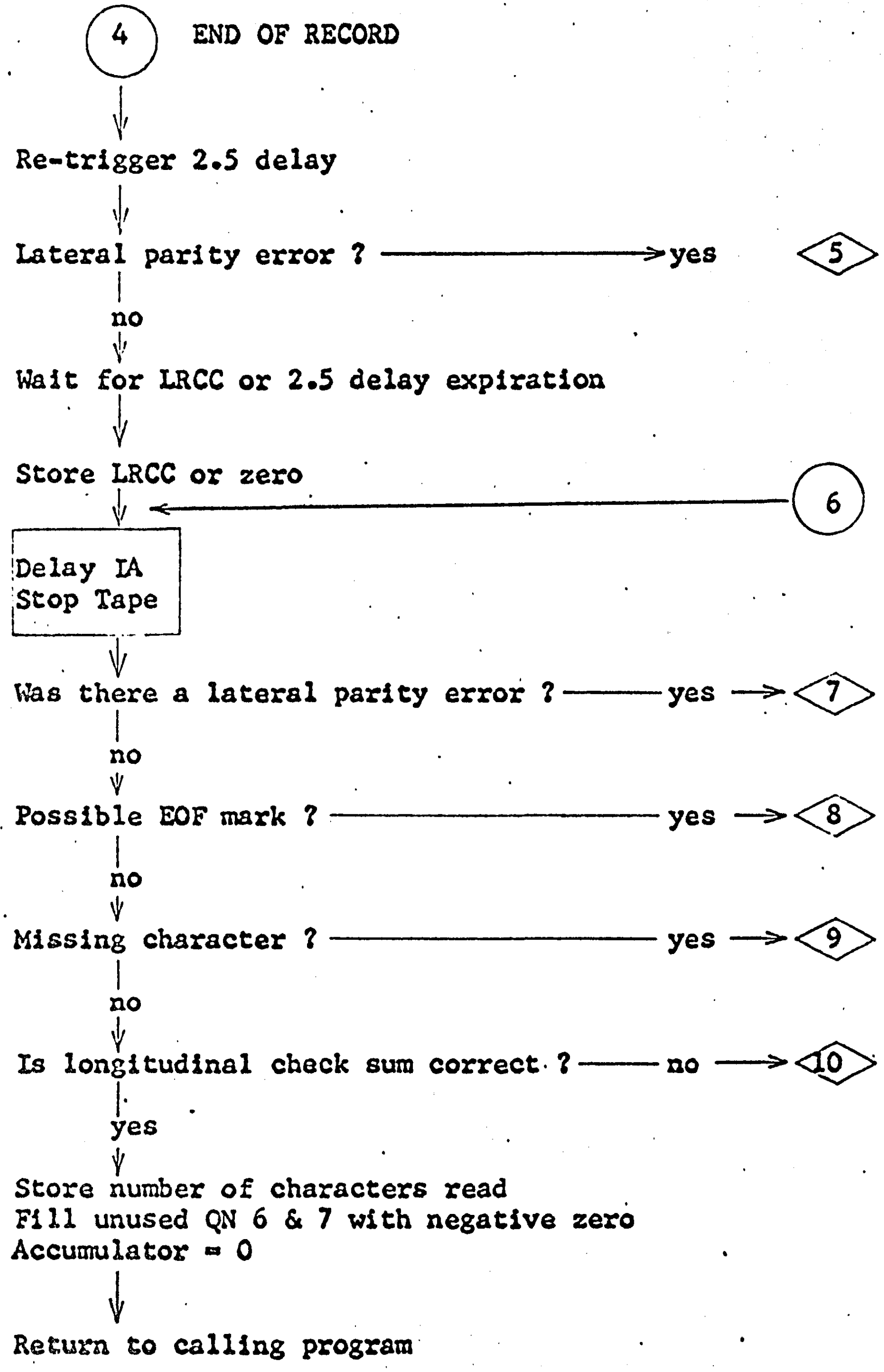
Flow chart #2

WRITE END OF FILE MARK (EOF) : Initial Tag 2K

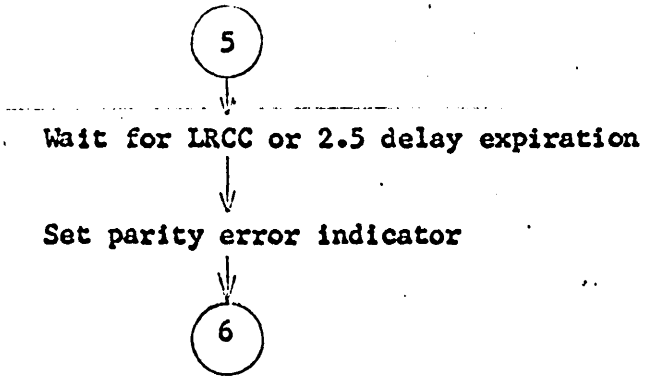


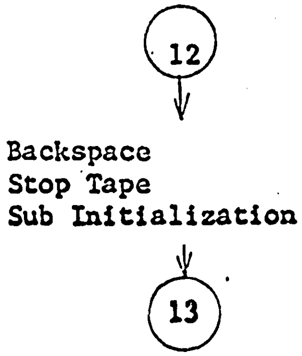
READ FORWARD : Initial Tag 3A

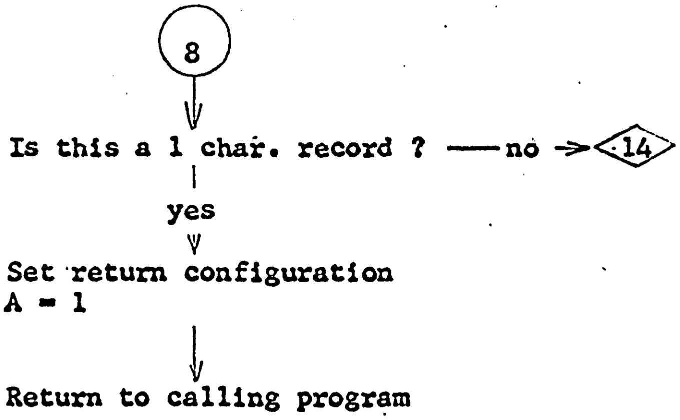


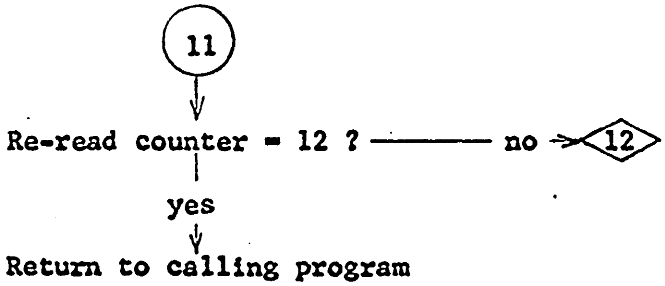

4
END OF RECORD
Re-trigger 2.5 delay
Lateral parity error ?
yes
5
no
Wait for LRCC or 2.5 delay expiration
Store LRCC or zero
6
Delay IA
Stop Tape
Was there a lateral parity error ?
yes
7
no
Possible EOF mark ?
yes
8
no
Missing character ?
yes
9
no
Is longitudinal check sum correct ?
no
10
yes
Store number of characters read
Fill unused QN 6 & 7 with negative zero
Accumulator = 0
Return to calling program

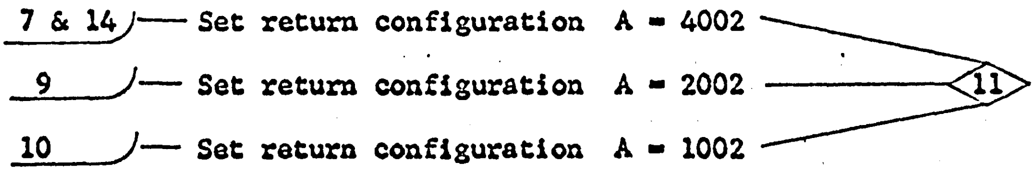
READ FORWARD - continued





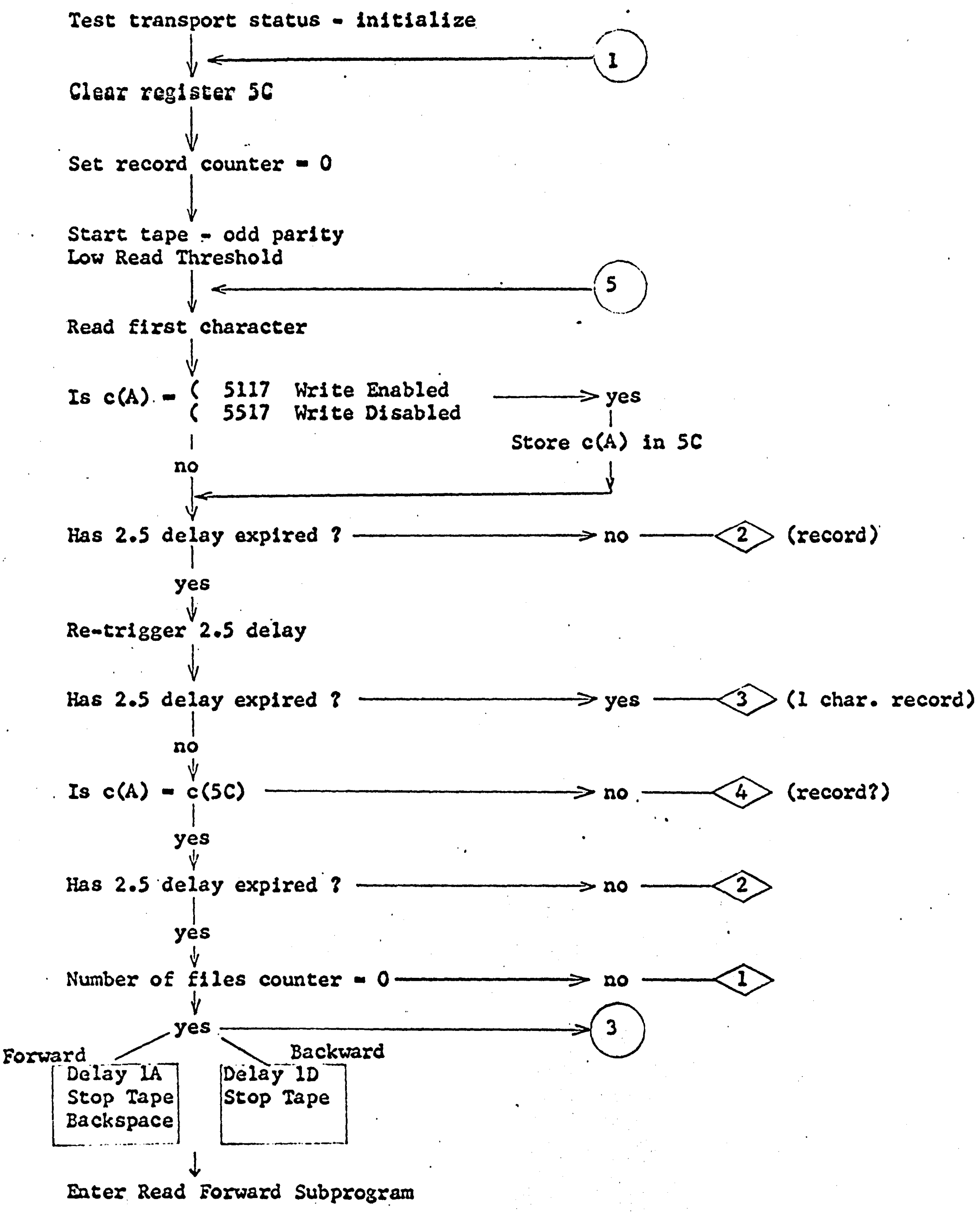


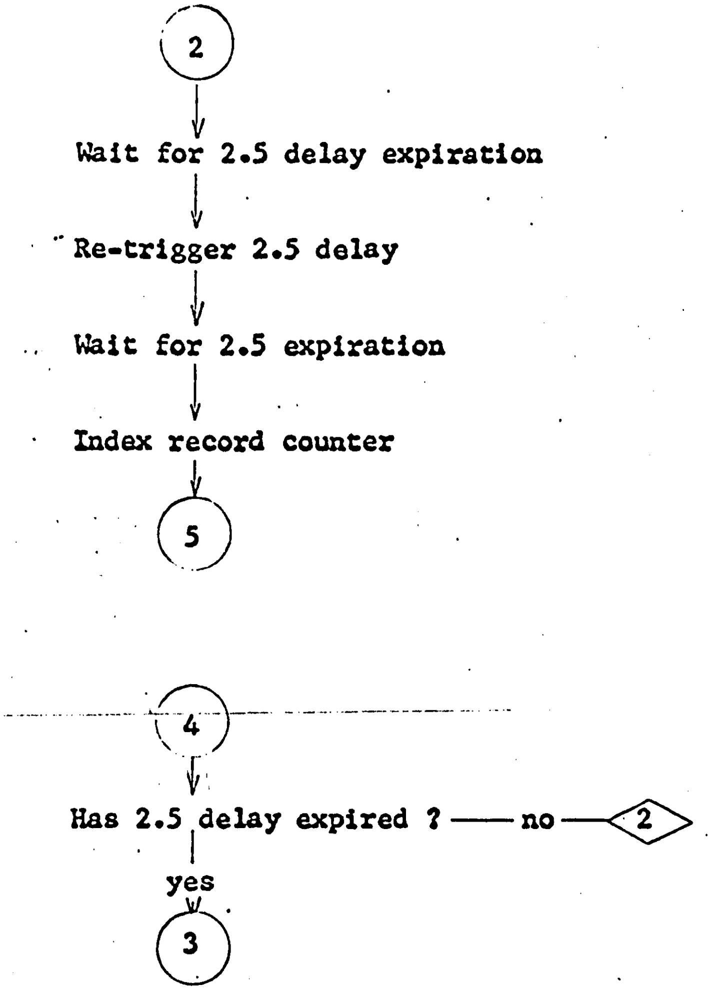


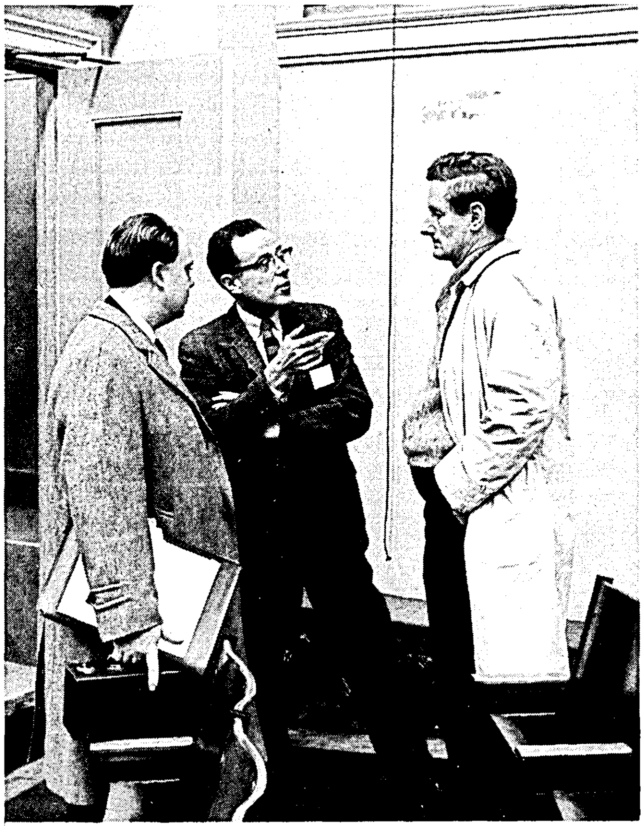


SKIP EOF = (FORWARD : Initial Tag 3L
(BACKWARD : Initial Tag 3K

Enter subprogram with number of EOF counter set




2
Wait for 2.5 delay expiration
Re-trigger 2.5 delay
Wait for 2.5 expiration
Index record counter
5
4
Has 2.5 delay expired ?
no
2
yes
3



LINC Evaluation Report

Submitted by:

Bernard Weiss

The Johns Hopkins University School of Medicine

7-1

Past and Present Research

In this laboratory, the LINC has been used to conduct and analyze experiments on operant behavior. A general purpose digital computer offers a number of features which make it ideal for such experiments because of the ease with which one can arrange complex schedules of reinforcement without patching up new circuits for each new modification. But this is a matter of convenience. It is much more important that the digital computer offers new dimensions in behavior research; it allows one to devise experiments unhampered by the present limits of instrumentation and also permits a detailed examination of the fine structure of behavior. For example, computers permit one to synthesize schedules with both long and short-term serial dependencies and with stochastic processes imposed.

<u>On-Line Programs</u>:

The experiments so far in which the LINC has been used on-line to generate reinforcement schedules have mostly involved monkeys. Several kinds of programs have been written and used.

1) <u>Autoregressive schedule</u>. On the basic schedule, the delivery of reinforcement (the reinforcer is a small amount of fruit juice) depends on the consistency of the Interresponse Times (IRTs). An IRT is defined as the waiting time preceding a response; it is measured from the previous

response. The LINC program evaluates consistency by comparing two adjacent IRTs. In the program, IRT_{i+1} is divided by IRT_i. The closer these two values are to one another -- i.e., the closer the ratio is to 1.0 -- the higher the probability of reinforcement. This probability falls off exponentially as the ratio deviates from unity.

The behavior resulting from this schedule is comprised of two components. One is a tendency to respond in a near-periodic fashion. The other is a significant serial dependency in IRT duration imposed on this tendency. Thus, this program generates both a low overall variability and a low serial variability.

Other modifications of this basic program have also been studied. In one, reinforcement does not occur on each occasion when it is due. Instead it occurs every _n_th occasion. With _n_=4 or 8, we have found that the overall variability is even more sharply bounded but that the serial dependency decreases. In another variation, we have compared, not successive IRTs, but the current IRT vs. the mean of the last _j_ IRTs. This procedure also reduced variability.

2) Stochastic Reinforcement of Waiting. One way in which IRTs can be differentially reinforced is to set a lower limit. Every IRT longer than the minimum duration is reinforced; those that are too short only reset the time counter. We have used the LINC to devise a stochastic analog of such a schedule. Instead of using a sharp cut-off, we have arranged it so

that the probability of reinforcement increases linearly with the duration of the IRT.

The behavior typically generated by this schedule is characterized by a bimodal distribution of IRTs. One peak lies at about 0.5 secs. The other peak, which is much less in amplitude, lies close to about 20 seconds. A more unimodal distribution can be achieved by evaluating, not single IRTs, but the mean of the last j IRTs. Such a procedure punishes long trains of very short IRTs and they gradually fall in frequency. In another variation, the function relating probability of reinforcement to IRT duration is Gaussian, rather than linear.

3) Differential Reinforcement of Low Rate. We have also studied the traditional schedule described earlier, namely, the one in which only IRTs longer than a specified minimum are reinforced. Our original plan was to provide such training before shifting to the stochastic schedule just described, but certain features of the behavior on the traditional schedule became apparent after analysis by the LINC and we continued to study it. We found, particularly after a moderate amount of training, that quite strong serial dependencies were present in IRT duration which often could easily be observed even on a simple display of the durations of successive IRTs.

4) Other Programs. In addition to the programs outlined above, we have also written an on-line program that we have started to use in studies of spontaneous activity in grouped mice given various drugs. We

hope to obtain some information on the variables underlying strain differences in the response to grouping. We are also preparing a program to study the behavior generated by a schedule involving various transition probabilities for specified responses.

As an example of the programming and hardware involved in the execution of one of these on-line programs, consider the autoregressive schedule.

Peripheral equipment: Three monkeys are generally studied at once. Each monkey, in a separate chamber, has access to a microswitch lever. These are connected to DEC 4410 pulse generators in the Data Terminal Box (DTB). Each depression and release of a lever produces a pulse from the appropriate pulse generator. This pulse is used to set a flip-flop whose 1 output is connected to one of the external sense lines. The reset lines come from appropriate OPR terminals in the DTB. Reinforcements are delivered by means of the LINC relays, which actuate 24 v. mercury-wetted relays that close a circuit to a solenoid valve which controls the flow of fruit drink used as the reinforcer.

The DTB also contains a DEC 4401 clock used to time the IRTs. For the ARG schedule, we used a clock rate of 100 pps. At the end of an experiment, an OPR pulse actuates a relay that turns off equipment such as stimulus lights, noise generators, and graphic recorders.

Program: The program is so structured that several monkeys can be studied at once with each IRT identified by monkey.

1. Set up indexing for recording; store session time; store address of first block to be used on data reel; store fixed ratio (KBD); set up random number generator.

2. Examine external levels from clock and 3 monkeys in turn. If clock pulse noted, add 1 to each monkey's IRT counter and to session time, and determine whether session is ended. If no monkey has responded, search external lines again.

3. If a particular monkey has responded, store his identification number for future reference; store the current IRT. Divide the current IRT by the previous IRT or the reverse, putting the larger value in the numerator (octal division); use quotient to locate probability value table (probability is equal to entry divided by largest possible random number value); store table value; generate quasi-random number; if table value exceeds random number, go to reinforcement routine and see if ratio has been met; if smaller, merely record.

4. To reinforce, actuate appropriate relay, the duration of closure being set by a program loop; label IRT as reinforced by placing a 1 in the sign bit. Add 1 to reinforcement counter.

5. Add 1 to response counter; store IRT; store monkey identification number in adjacent word; if memory is filled with data, write tape; if not, return to external level examination.

Because data are being gathered from 3 monkeys simultaneously, part of the program is aimed at keeping their identities clear. The entire program requires about 600_8 instructions.

Analytical Programs:

1. General display: This is a program built up from one originally provided us by Doctors Hance and Killam. It allows us to scan and transform the IRT data in various ways. For example: display data as a bar graph; transform with 3-term moving average; transform with 7-term moving average; plot deviations from moving average; plot successive IRT differences; differentiate; integrate; plot mean; perform peak clipping; reshuffle serial order randomly; expand or contract display; etc. Transformed data can be saved and studied.

2. Histogram display: This plots an IRT histogram of specified data. Options include; variable size; conditional probability transformation; distribution of IRTs following IRTs above or below the median. Resolution and time scale are variable. Axes are also displayed.

3. Joint interval histogram: Plots IRT_i vs. IRT_{i+L}. L is variable.

4. Sequence display. Plot , as a bar graph, sequences of 4, 3, and 2 IRTs, according to whether or not they lie above or below the median.

5. Expectation density: Plot this function, which is a type of averaging, with variable resolution.

6. Miscellaneous: Include runs test, 2^n histograms, histograms of reinforced and post-reinforcement IRTs, mean square successive difference, etc.

Utility Programs:

1. Variable character display: For making graphs.

2. <u>T.T. plot out</u>: Uses teletype to make graphs.

3. <u>Random number generators:</u> Various kinds.

4. <u>Floating point conversion</u>: These use FLP routines supplied by L. Hundley to enter and transfor data from various kinds of off-line experiments as well as experiments performed with the LINC.

Future Research

While we have made a beginning that we believe demonstrates some of the great potential of computers in operant behavior research, a vast range of unexplored applications still remain. In this section of our report, we shall discuss some of these and outline our future plans for the LINC. These are to be taken, of course, as an indication of our wish to retain the LINC for our laboratory.

Introduction. It is clear, of course, that digital computers are indispensable for performing a microanalysis of operant behavior. Even with traditional reinforcement schedules, such as the DRL schedule (Differential Reinforcement of Low Rate), a detailed analysis of the time-series structure of the behavior has revealed properties not observed before. It is less clear, to many investigators, that digital computers are also virtually indispensable for generating reinforcement schedules with properties akin to those found in natural environments. These are: the stochastic character of the reinforcement contingencies and the prevalence of short and long-term serial dependencies. A digital computer can easily simulate such properties.

One of the reasons that digital computers and operant behavior seem so well-matched is that operant behavior can be easily translated into digital terms. One method of doing so is to shape behavior according to the waiting times between successive discrete responses. Such interresponse times (IRTs) can be modified by the process of differential reinforcement in exactly the same way as spatially-defined responses. Their coding makes them ideal for computers.

Methods. In addition to the work performed so far, which we plan to extend (see section on Previous Work), we plan to use the LINC to investigate several other kinds of behavioral phenomena.

a. Schedule with periodic components. On certain reinforcement schedules, the reinforcement is programmed to appear a constant length of time following the previous reinforcement. This is called a Fixed-Interval (FI) schedule. If the intervals are not uniform in length, it is called a Variable-Interval (VI) schedule. Typically, FI schedules are used to study behavior controlled by periodic events and VI schedules are used to produce relatively constant rates of responding. These two schedules become most logically related when they are described in terms of their component waveforms--when they are described, that is, as an engineer might describe them.

Once such an analysis is made, we can then plan to study behavior maintained by interval schedules from a more uniform point of view. Thus, any interval schedule would then be described as a waveform with known component frequencies, and we could then analyze the resultant behavior from the standpoint of how these component frequencies are reflected in the temporal structure of the behavior.

Using the LINC, we can then not only perform the proper analyses, but also generate new variants of such schedules. For instance, we could generate a schedule on which the probability of reinforcement rose and fell as a sine function. We could then impose more and more randomness on the function to determine the degree to which behavior reflected the underlying pattern in the presence of "noise." Interesting questions spring from such an approach in relation to drug effects and the stimulus control of behavior (Weiss, B. and Laties, V.G. *Fed. Proc.*, *23*:801-807, 1964).

b. *Schedules with state changes*. Changes in the sequential patterns of behavior may occur, not only with IRT measures, but with discrete responses, and may be shaped accordingly. For example, suppose that in a chamber containing three levers, A, B, and C, one arranged the following transition matrix:

		Response$_{i+1}$		
		A	B	C
Response$_i$	A	.01	.02	.04
	B	.02	.04	.01
	C	.04	.01	.02

That is, given that response$_i$ was made on lever A, the probability of a second response on lever A producing reinforcement is .01; the probability of the succeeding response producing reinforcement is .04 if lever B is selected. Humans find sequential dependencies greater than simple Markoff chains extremely difficult to learn. We plan to study such chains in the free-operant situation and then determine whether it is possible to go on to higher-order chains. Perhaps, by employing ratio schedules, it will be possible to find evidence of control by higher-order dependencies by looking at the distribution of responses on the various levers.

c. Adaptive schedules. Schedules exist on which a function or value is varied, in accordance with some rule, by what the subject does. These have been called by various names: adjusting, titration, and conjugate schedules.

In our laboratory, such a schedule has been used to study pain and analgesia. The subject, via electrodes, receives an electrical signal whose intensity rises in steps, an increment occurring every few seconds. Each response, e.g. a lever press, drives the current down by a step. In this manner, one can train monkeys to trace out an aversive threshold that displays remarkably little variability.

This is a psychophysical method which has proved relatively sensitive to analgesic agents. We are interested in pursuing it further with the aid of the LINC. One may think of the system as a proportional controller; the probability of a decremental response is a function of shock intensity. Would it be possible to produce a more stable threshold by adding, for example, derivative control? This would be accomplished by making the size of the step that decreased the shock a function of the rate of response; the greater the rate, the greater the size. By viewing the monkey as a sort of psychophysical control system we might learn something about the aversive control of behavior as well as make the technique more sensitive.

d. Traditional schedules. Our understanding of how the more traditional types of reinforcement schedules exert their effects is still elementary. We therefore plan to examine some of these with the LINC as we have done with the DRL schedule of reinforcement.

In addition to the research discussed above, we plan to continue our studies with drugs. So far, we have completed only one drug experiment, mainly because we felt it important to enhance our skills in the use of computers and to perform purely behavioral studies first. Now, however, we are ready to pursue our interests in behavioral pharmacology.

Facilities. Beginning June 1, 1965, both the principal investigator and the co-investigator, Victor Laties, will be on the faculty of the University of Rochester School of Medicine, with primary appointments in the Department of Radiation Biology. In addition, the principal investigator will have a joint appointment in the Center for Brain Research and the co-investigator will have one in the Department of Pharmacology.

Radiation Biology is the largest department in the Medical School at Rochester, and we will have access to all of its many facilities. These will include a computer center partly devoted to on-line work, so that we hope to be able to expand the scope of our experiments by connecting lines to their computers. The great variety of disciplines represented in Radiation Biology, coupled with our joint appointments, will undoubtedly extend the range of behavioral problems studied with the LINC. We are now in the process of designing interface logic to be used with Dataphone transmission between laboratories in order that we may conduct on-line experiments from our laboratory in the Department of Medicine. At Rochester, this system will enable us to conduct collaborative work with the Center for Brain Research.

Because of the move to Rochester, we request that the renewal of the LINC grant, if approved, be made through the University of Rochester.

7-14

Training Program

The training program that we underwent at M.I.T. was excellent in most respects. At the time, I felt that too much emphasis was being placed on the maintenance and design of the LINC and not enough on the programming. Since, then, I have changed my mind. Programming skills, I found, once a computer was available to use as a teaching machine, were more readily acquired than I anticipated, although I could have used guidance when I began to try to write complex programs.

The material on design and maintenance proved valuable for two reasons. In the first place, a considerable amount of trouble-shooting remained to be done even after the machine was delivered to Baltimore. Although I had a great deal of on-the-spot assistance from the C.D.O. staff, the information I was able to give by telephone often sufficed to provide a diagnosis.

In the second place, the design of reliable interface circuitry for our particular needs proved to be no simple matter, and the education I had received during the training program plus my trouble-shooting experience was an immense help in evolving an appropriate system. Discussion of interface problems during the course would have been extremely useful.

Naturally, a less hurried program would have been easier to cope with. Perhaps more important, if we had had more access to a LINC during the course, and could have spent considerable time learning its features by actual operation, many of the points made would have been easier to grasp. The LINC itself is an excellent teacher and it is unfortunate that its potential could not have been exploited because of the circumstances.

Computer Performance

a. Maintenance. A number of early difficulties arose from the prototype nature of our computer. As can be seen from our log entries, some of our early problems resided in the power supply. These were solved eventually after the redesigned circuits were incorporated. Other problems were found to be due to circuit cards that had been hand-soldered, and that contained poor connections. Still others were simply the result of what turned out to be faulty components, such as transistors. During the first three months of operation, therefore, a large percentage of time was spent on maintenance, although the machine ran well for periods sufficiently long to give me practice in writing programs.

During the next few months, my major difficulties were due to interfacing logic. Perhaps this is one area in which the training program could have made a greater contribution, at least for those of us who were not experts. Our problems were mainly those of electrical noise produced by relays, solenoids, and motors. After making numerous changes in our designs and former methods, and consulting with the C.D.O. staff, we finally developed a system that has worked rather well. For about a year, now, the LINC and its interface equipment have operated with very little time lost to dysfunction. The largest problems arose from an oscilloscope malfunction and from a nicked cable. The upper half of memory was installed by C. Molnar on 10/3/64 and has been operating satisfactorily since.

b. LINC design. The LINC is extremely well-suited for operant behavior research. Its speed makes it easy to devise very complex programs without worry about time resolution. Its interface flexibility allows a wide variety of peripheral equipment arrangements. I will comment on some features that might make it more convenient for behavioral work in my laboratory.

(1) Memory size: For data analysis, the present 2048 word memory is generally sufficient. The LINC would be a more efficient machine for on-line use if the upper 1024 words were programmable, so that more experiments could be conducted simultaneously. If four different programs could run at once (each involving, say, three monkeys) the cost of the LINC would be competitive with the cost of the conventional equipment required to conduct this many experiments. (I am estimating $3000 per monkey.) The flexibility of the LINC and its analytical power would then be a bonus.

(2) Tape transfers: When 3 monkeys are all responding at a high rate, and fine resolution (10 msecs, say) of the IRTs is needed, tape transfers can introduce a small error because of the time required for tape motion. If the transfer were autonomous, so that block searching might go on at the same time as the central program, part of this problem would be avoided. Perhaps the buffering required could be incorporated into the DTB.

(3) Compared to the present D.E.C. relay packages (e.g., no.1803), the LINC relays are rather slow. Also, I would prefer to have the outputs which drive the relays available independently for other purposes should I

need them. I would be in favor of changing these to -3 v. level outputs to be used as the experimenter wishes, which can be done now if the relays are not used. This feature would also mean more room in the Data Terminal Box.

(4) The Data Terminal Box is a good design for those experimenters who concentrate on one kind of experiment. Those who, like myself, conduct several different kinds of studies, might find it more convenient, I think, to have the DTB output available *via* a plugboard to which can also be connected a wide variety of logic elements as well as lights, switches, etc. We are experimenting with such a system now, using the Mac Panel plugboard. We shall inform the Computer Research Laboratory of our progress, if any.

Bibliography

A. *Published papers and papers in press*.

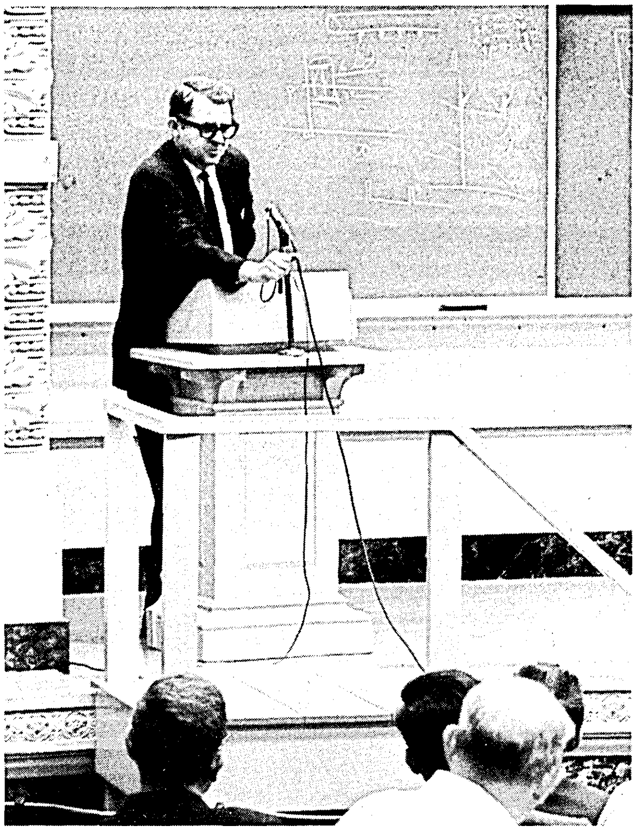
1. Weiss B. and Laties, V.G., Drug effects on the temporal patterning of behavior. *Fed. Proc.*, *23*:801-807, 1964.
2. Weiss, B. and Laties, V. G., A reinforcement schedule generated by an on-line digital computer. *Science* (in press).
3. Weiss, B., Laties, V.G., and Siegel, L. A computer analysis of serial interactions in spaced responding. *J. exper. anal. Behav*. (submitted for publication).
4. Siegel, L. A method for converting voltage levels. *J. exper.anal. Behav*. (in press).

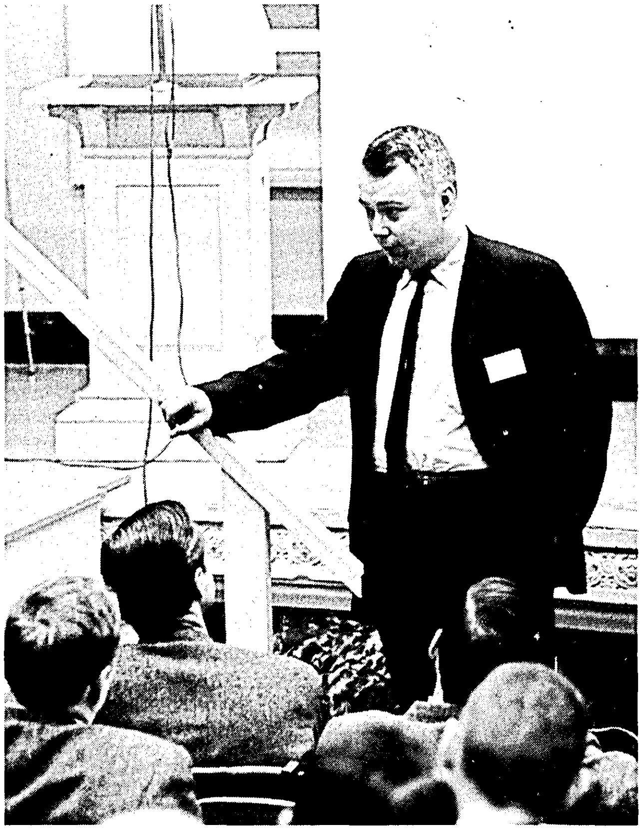
B. *Symposia*

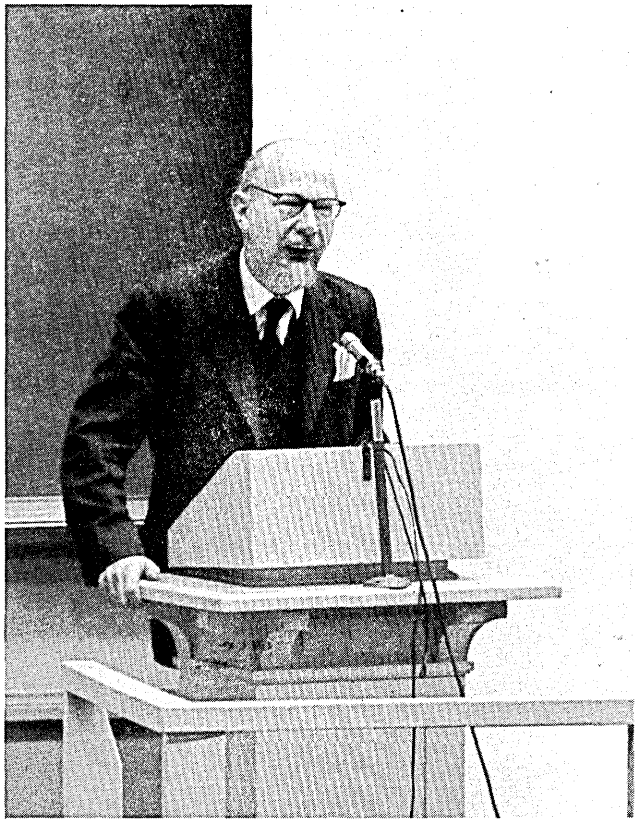
1. Weiss, B. and Laties, V.G. Programming reinforcement schedules with an on-line digital computer. American Psychological Association meetings, September 6, 1964 (Symposium on the Experimental Analysis of Behavior).

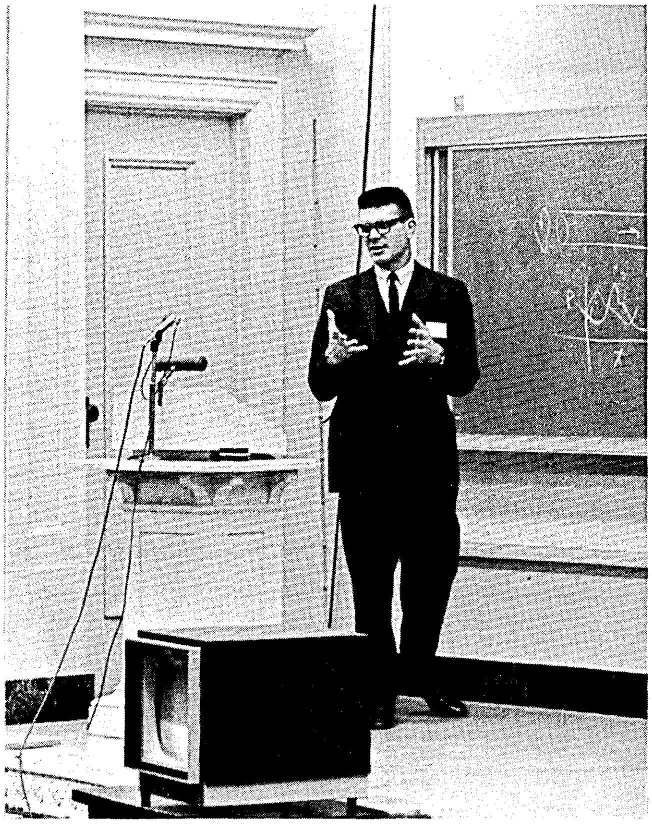
C. *Talks*

1. Johns Hopkins University School of Medicine.
2. Institute for Behavioral Research.
3. Albert Einstein College of Medicine.
4. University of Rochester.









LINC Computer Evaluation

Grant FR 00146-01

June 1, 1963 - March 31, 1965

Principle Investigator: Dr. Fred S. Grodins, Professor of Physiology

Co-principle Investigator: Dr. James E. Randall, Professor of Physiology

Northwestern University Medical School
303 E. Chicago Avenue
Chicago, Illinois 60611

FINAL REPORT
March 1, 1965

8-1

1. THE NORTHWESTERN UNIVERSITY GROUP WISHES TO KEEP THEIR MACHINE

The LINC has become a way of life in this laboratory and its loss would be considered a near-catastrophe. Two graduate students have Ph.D. dissertation problems directly related to the machine. One faculty member has a grant-supported research problem which directly depends upon the LINC for data analysis. Other faculty members are working in general areas which are almost certain to generate problems which can utilize the LINC's capabilities in the future.

We therefore request that we be allowed to retain this LINC in our laboratory. This is a reflection of the most important conclusion of our evaluation program, i.e., we have found the LINC to be a very practical, very useful, in fact an essential laboratory instrument and we want to keep it!

2. PAST AND PRESENT USES

A. Classes of Programs

The following kinds of programs have been written for use in the research problems described later.

1. Teletype subroutines -- This general-purpose package, taking up one memory quarter, includes typing the contents of the accumulator in decimal. This has greatly facilitated the interpretation of output data. Other operations include typing a stored table of teletype characters and returning to the mainline routine when the character for * (252) is read. These teletype characters can be stored as two 6-bit numbers per address since those ending between 00 and 37 are preceeded by a 3 and those between 40 and 77, by a 2. Line feed, return, and space are separate instructions in the subroutine (JMP 2F, JMP 2R, and JMP 2S, respectively).
2. Double-precision subroutines -- These permit addition, multiplication and scale right operations for 23-bit numbers stored in two consecutive addresses.
3. Autocorrelation and variance spectrum -- $4{,}000_8$ 6-bit samples with 100_8 lags. These are displayed on the scope. If these are of interest, a block number is typed on the keyboard and the correlation and spectral values are saved on tape in that block. This data may be recalled and typed as graphs on the teletype. The mathematical expressions are given in APPENDIX III.
4. Cross-correlation and cross-spectra -- Cross-correlation of $2{,}000_8$ data pairs, with 100_8 lags. Program also computes and types out graphs of in-phase and quadrature covariance spectra. These will later be expanded to compute the transfer function (gain and

phase as a function of frequency) between the two statistical inputs. The mathematical expressions are given in APPENDIX III.

5. Miscellaneous -- Other more specialized routines, to be described elsewhere, included: random pulse generator, synchronization of analog tape with LINC, reading movie images projected upon photocells, and reading amplitude histograms.

B. Finger Tremor Variance Spectra

In terms of data collection, the major laboratory use of the LINC has been for computing the variance (power) spectra of finger and hand tremors. Robert N. Stiles is evaluating some of the mechanical factors which determine the spectral peaks of the finger tremor as part of his doctorate research problem. The speed of data reduction which the LINC makes possible has allowed us to determine what factors must be controlled to get reproducible spectra and to appreciate that this is a distributed mechanical system with many vibration modes. Programming for cross-spectral analysis between tremors at two locations is just about completed.

When the position of a finger is being maintained there is a statistical fluctuation about a mean position. This tremor has been widely studied in descriptive terms because of the interest in its origin and purpose and because it becomes exaggerated in certain neurological diseases. Observations have traditionally used many different measures of the frequency content and of the amplitude of the tremors. Quantitative tests of specific hypotheses concerning the source of tremor have been limited by lack of a unifying scheme which is mathematically precise but which will allow for the random nature of the fluctuations. Our present purpose is to put some order and meaning into the measurement and quantitation of this fluctuation by treating it as a statistical time series. Variance is used as a measure of tremor magnitude and variance spectra indicate the frequency information averaged over a sampling period. Cross-spectral analysis between tremor and other physiological signals may give clues regarding the relative contributions of several possible factors to the total variance.

Tremor is sensed by a 2.5 gram variable reluctance accelerometer taped to the finger of a subject. The output of the accelerometer amplifier system is a ± 1 volt signal which can be applied to the analog input jack on the LINC. Sampling rates of about 80/second are used and the samples stored in memory. Data has been analyzed in batches corresponding to a few seconds of the tremor. Longer samples are needed to be certain that the statistical parameters have reached some sort of a steady state. Methods of doing this are described elsewhere in this report.

Our attention has been focused on the center frequencies of the spectral peaks of acceleration tremor. We have preliminary data which are presented here only as illustrations of how the LINC has been used, and not as evidence for or against any particular hypothesis of tremor mechanism. One spectral peak, near 9 to 10 c/sec, is particularly consistent. Since this peak is relatively sharp, often with a bandwidth of about 1 c/sec, the 10 c/sec component is the one most obvious in a time tracing. Much of the literature on tremor has been focused on this particular frequency band. However, by sensing acceleration (which is more sensitive to higher frequencies than is a measurement of displacement) and using a kind of spectral analysis which is sensitive to random, non-periodic fluctuations, we have revealed additional frequency components which are significantly greater than the background noise. These other components are more variable and often have broad spectra which would not be revealed by many of the methods of analysis used in the past. The center frequencies of these bands appear to be particularly associated with mechanical factors associated with the hand and finger.

When all of the hand is supported except for the forefinger, the broad spectral peak occurs at about 25 c/sec., as in Figure 1.1. Adding weights to the finger lowers the center frequency of this band, as in Figure 1.2 where a 10 gram weight had been taped to the finger. As more weights are added the center frequency of this band falls, the relationship approximating the smooth curve to be expected for the resonant frequency of a linear mechanical system with lumped elasticity and mass. This spectral band also corresponds to the undamped natural frequency of the vibration observed when the finger is externally forced with a mechanical tap (approximating an impulse).

When the arm is supported, leaving the hand and fingers free to move, the broad spectral band centers near 15 c/sec (Figure 1.3). This frequency is lowered when masses are added to the hand, and corresponds to the frequency of the damped vibrations which an externally forced hand undergoes. On some measurements where only the finger was free to move there was evidence of the 15 c/sec component associated with the hand.

One of the more interesting aspects occurs when sufficient mass is added to bring the high-frequency spectral peaks down to below 9 - 10 c/sec. In Figure 1.4 the hand was supporting a 450 gram mass and the two spectral peaks (9 c/sec and 15 c/sec) appear to have merged and moved to about 6 c/sec. But upon increasing the mass added to the whole hand the two spectral bands separate again into one at 10 c/sec and one at 5 c/sec. This suggests some interaction between a specific load added to the hand and the origin of the 9 - 10 c/sec component.

C. Tremor Cross-spectra

The programs for cross-correlation analysis and in-phase and quadrature cross-spectra are just being completed. These will be used to rest for

```
BLOCK 0122        TOTAL POWER =  0225              LSW  0004
FREQ                    POWER/BANDWIDTH     X  0016
      ..... 0014
      ........ 0026
      ....... 0023
      ..... 0014
      ..... 0014
      ...... 0018
      ....... 0021
0005  ....... 0023
      ........ 0027
      .......... 0034
      .......... 0034
      .......... 0033
      ................. 0061
      ........................ 0089
      ..................... 0078
0010  ............... 0053
      ........... 0039
      ............ 0041
      ............ 0043
      ......... 0030
      ......... 0028
      ........... 0038
      ........... 0037
0015  ......... 0031
      ......... 0030
      .......... 0034
      ............ 0041
      .............. 0048
      ............... 0055
      ................. 0061
      .................. 0067
0020  ...................... 0081
      ........................... 0100
      .................................. 0131
      ..................................... 0143
      ..................................... 0142
      .............................................. 0176
      .............................................. 0179
      ................................... 0132
0025  ............................ 0104
      .......................... 0098
      ............................. 0110
      ........................... 0102
      .................... 0075
      ................. 0063
      ................ 0059
      .................. 0064
0030  ................. 0060
      ........... 0038
      ........ 0025
      ......... 0028
      ......... 0029
      ......... 0031
      ......... 0030
      ......... 0028
0035  ......... 0029
      ........ 0026
      ....... 0020
      ...... 0017
      ...... 0016
      ...... 0018
      ....... 0021
```

Figure 1.1 TREMOR VARIANCE SPECTRUM
Forefinger
$s^2 = 2 \times 10^{-4}\ g^2$
BW = 0.6 c/sec

7

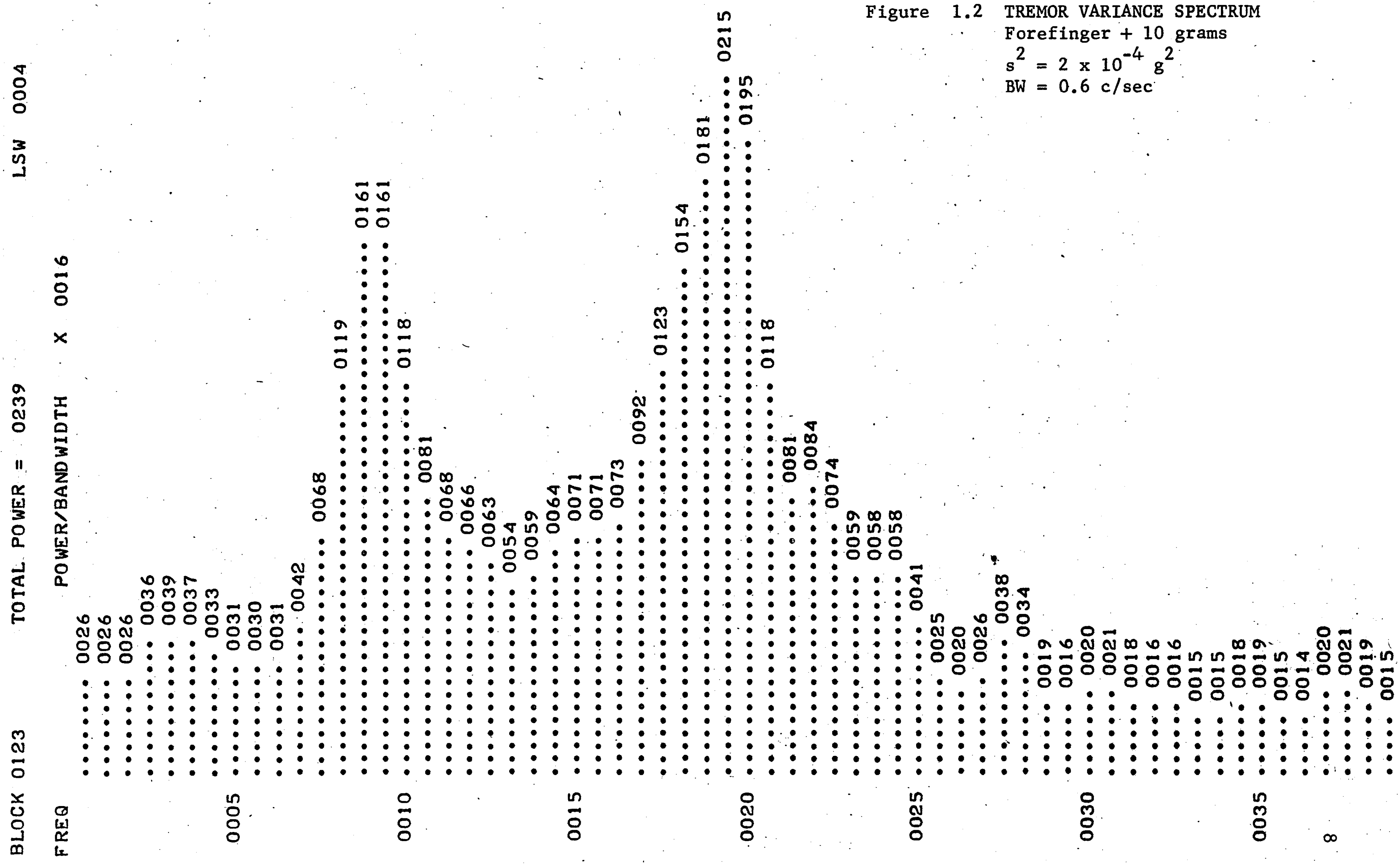


Figure 1.2 TREMOR VARIANCE SPECTRUM
Forefinger + 10 grams
$s^2 = 2 \times 10^{-4}\ g^2$
BW = 0.6 c/sec

```
BLOCK 0205         TOTAL POWER =  0045                LSW  0004

FREQ                     POWER/BANDWIDTH    X  0016
     •• 0001
     •• 0002
     •• 0001
     •• 0002
     •• 0001
     •• 0002
     •• 0002
0005 ••• 0004
     •• 0003
     ••• 0005
     ••• 0006
     •••• 0010
     •••••••••••• 0042
     •••••••••••••••••••••••••••••••• 0123
     •••••••••••••••••••••••••••••••••••••• 0144
0010 ••••••••••••••••••••• 0079
     •••••••••• 0033
     •••••• 0019
     •••• 0011
     •••• 0008
     •••• 0009
     ••••• 0013
     •••••• 0017
0015 ••••••• 0020
     ••••••• 0023
     ••••••• 0023
     •••••• 0016
     •••• 0011
     •••• 0008
     ••• 0006
     ••• 0005
0020 •• 0001
     •• 0001
     ••-0000
     •• 0001
     •• 0000
     •• 0000
     •• 0001
     •• 0001
0025 •• 0001
     ••-0000
     ••-0000
     •• 0001
     •• 0001
     •• 0000
     •• 0000
     •• 0000
0030 •• 0000
     ••-0000
     ••-0000
     ••-0000
     ••-0000
     ••-0000
     ••-0000
     ••-0000
0035 •• 0001
     •• 0001
     ••-0000
     ••-0000
     •• 0000
     •• 0000
     •• 0000
```

Figure 1.3 TREMOR VARIANCE
Hand

```
BLOCK 0266          TOTAL POWER =  0231              LSW  0002

FREQ                   POWER/BANDWIDTH    X  0004
      ... 0004
      ... 0005
      ... 0007
      ... 0005
      ... 0007
      ... 0007
      .... 0010
 0005 ....... 0023
      ............. 0046
      ........................ 0091
      .......................................... 0162
      ........................................... 0164
      ........................... 0101
      .............. 0051
      ......... 0030
 0010 ........ 0025
      ...... 0017
      .... 0010
      .... 0010
      ... 0007
      ... 0005
      ... 0006
      ... 0004
 0015 .. 0001
      .. 0002
      .. 0002
      .. 0001
      .. 0001
      .. 0001
      .. 0002
      .. 0002
 0020 ... 0004
      ... 0004
      ... 0004
      ... 0005
      .. 0003
      .. 0001
      .. 0001
      .. 0000
 0025 .. 0000
      .. 0000
      .. 0000
      .. 0000
      .. 0001
      .. 0000
      .. 0000
      .. 0000
 0030 .. 0001
      ..-0000
      ..-0000
      .. 0000
      .. 0000
      .. 0000
      .. 0000
      .. 0000
 0035 .. 0000
      .. 0000
      .. 0000
      .. 0000
      .. 0000
      .. 0000
      .. 0000
```

Figure 1.4 TREMOR VARIANCE

Hand + 450 grams

```
BLOCK 0222          TOTAL POWER =  0066              LSW  0004

FREQ                   POWER/BANDWIDTH    X  0016
       •• 0003
       •• 0002
       •• 0003
       •• 0003
       ••• 0006
       ••••• 0015
       •••••••••• 0032
 0005  ••••••••••••••••••••••••••••••• 0118
       •••••••••••••••••••••••••••••••••••••••••••••••••••••••• 0216
       ••••••••••••••••••••••••••••••••••••••••••••• 0175
       •••••••••••••••••••••• 0083
       •••••••••••• 0042
       ••••••••• 0031
       •••••••• 0027
       ••••••••••• 0037
 0010  ••••••••••••••••••• 0070
       •••••••••••••••••••• 0074
       •••••••••• 0034
       •••• 0008
       •• 0002
       •• 0001
       ••-0000
       ••-0000
 0015  •• 0001
       •• 0001
       ••-0000
       •• 0001
       •• 0000
       ••-0000
       ••-0000
       •• 0001
 0020  ••-0000
       ••-0000
       ••-0000
       ••-0000
       ••-0000
       ••-0000
       ••-0000
       •• 0000
 0025  •• 0000
       •• 0000
       •• 0000
       •• 0000
       ••-0000
       ••-0000
       ••-0000
       ••-0000
 0030  ••-0000
       ••-0000
       •• 0000
       •• 0000
       ••-0000
       ••-0001
       ••-0000
       ••-0000
 0035  ••-0000
       ••-0000
       ••-0000
 II    ••-0000
       ••-0000
       ••-0000
       ••-0000
```

Figure 1.5 TREMOR VARIANCE
Hand + 550 grams

correlations between tremor and other physiological fluctuations as clues of the sources of the tremor spectral peaks. A few preliminary observations are given here as illustrations of the LINC performance even though we are not in a position to make any physiological interpretations as yet.

The cross-spectra can be illustrated for pairs of signals of known phasic relations. The cross-correlation function for a signal compared to itself is the same as its autocorrelation function. For this case there are no quadrature spectral values and the in-phase spectrum is the variance spectrum of the signal. This is illustrated in Figures 2.1 and 2.2 which are the in-phase and quadrature spectra for a 10 c/sec sine wave cross-correlated with itself. Figures 2.3 and 2.4 show that for the correlation between a sine wave and a cosine wave there is only a quadrature spectrum.

One accessible signal to be compared with the finger tremor is the electromyogram, the complex summation of the action potentials arising from the many motor units which fire during muscular activity. Figure 2.5 shows the variance spectrum of the electromyogram measured at the muscle which appears to be contributing to finger tremor. The variance is spread out over a broad spectrum peaking near 75 c/sec and showing very little activity at the frequencies of the tremor spectral peaks. If the electromyogram is rectified and passed through a low-pass filter the resulting waveform is related to the envelope of the electromyogram and is often considered an index of muscular activity. The power spectrum of this envelope, as shown in Figure 2.6, has peaks at frequencies corresponding to those in the tremor recording. This then raises the questions whether there is a consistent phasic relation between electromyogram envelope and tremor. The cross-spectral analyses will eventually measure the consistency of any relationship and perhaps the dynamic nature of it.

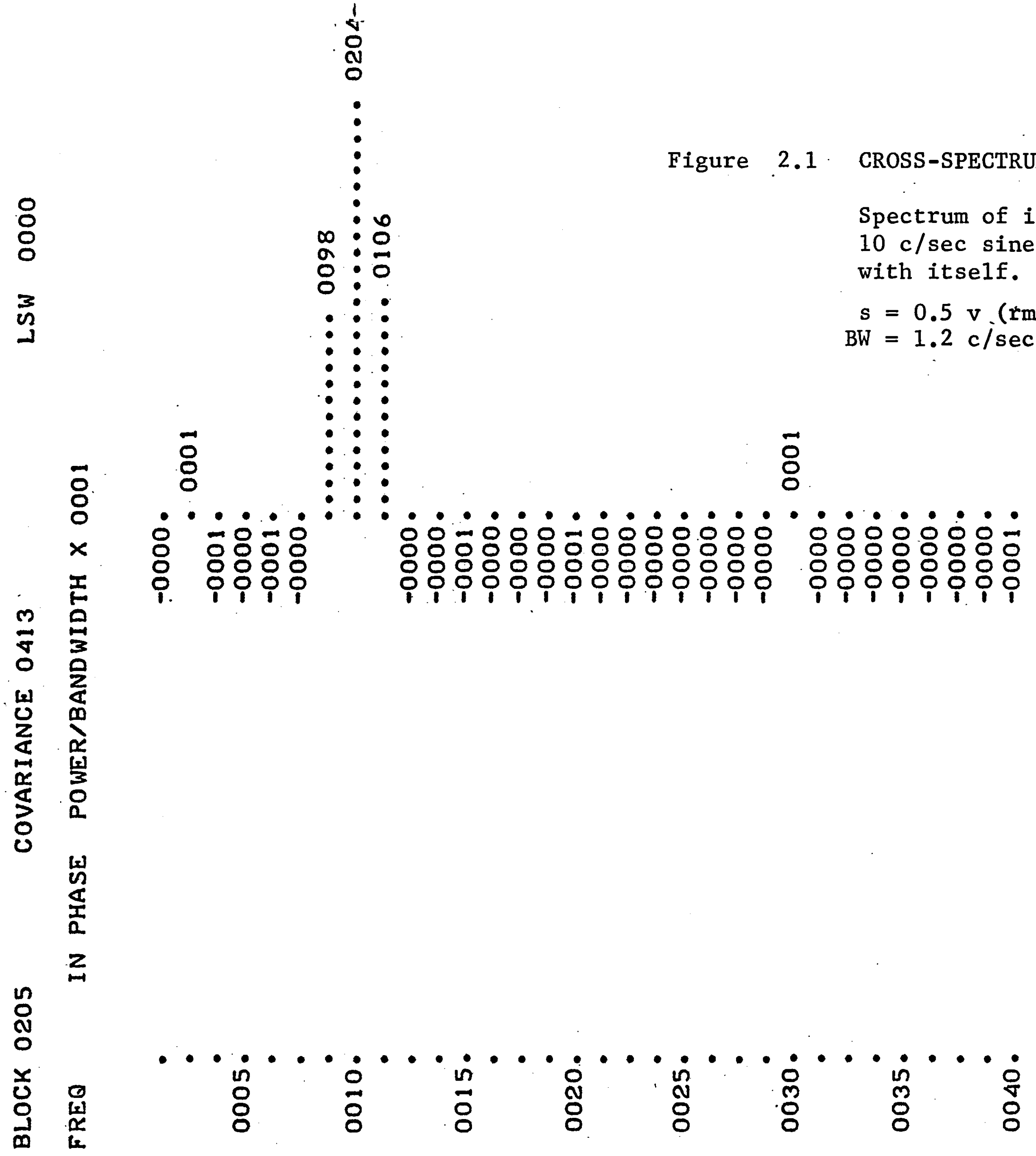


Figure 2.1 CROSS-SPECTRUM OF SINE WAVE

Spectrum of in-phase components of 10 c/sec sine wave cross-correlated with itself.

s = 0.5 v (rms)
BW = 1.2 c/sec

8-14

```
BLOCK 0205        COVARIANCE 0413              LSW  0000

FREQ      QUADRATURE POWER/BANDWIDTH X 0001

    .                        -0000.
    .                        -0000.
    .                        -0000.
0005.                        -0000.
    .                        -0000.
    .                        -0000.
    .                        -0002.
0010.                        -0000.
    .                             . 0002
    .                        -0000.
    .                        -0000.
0015.                        -0000.
    .                        -0000.
    .                        -0000.
    .                        -0000.
0020.                        -0000.
    .                        -0000.
    .                        -0000.
    .                        -0000.
0025.                        -0000.
    .                        -0000.
    .                        -0000.
    .                        -0000.
0030.                        -0000.
    .                        -0000.
    .                        -0000.
    .                        -0000.
0035.                             . 0000
    .                             . 0000
    .                        -0000.
    .                        -0000.
0040.                        -0000.
```

Figure 2.2 CROSS-SPECTRUM OF SINE WAVE

Spectrum of quadrature components of 10 c/sec sine wave cross-correlated with itself.

```
BLOCK 0170         COVARIANCE 0007             LSW  0000

FREQ      IN PHASE  POWER/BANDWIDTH X 0001

        .                           -0000.
        .                           -0000.
        .                           -0000.
    0005.                           -0000.
        .                           -0000.
        .                           -0000.
        .                           -0000.
    0010.                           -0000.
        .                           -0000.
        .                           -0000.
        .                           -0000.
    0015.                           -0000.
        .                           -0000.
        .                           -0000.
        .                                . 0001
    0020.                                . 0003
        .                                . 0003
        .                                . 0001
        .                           -0000.
    0025.                           -0000.
        .                           -0000.
        .                           -0000.
        .                           -0000.
    0030.                           -0000.
        .                           -0000.
        .                           -0000.
        .                           -0000.
    0035.                           -0000.
        .                           -0000.
        .                           -0000.
        .                           -0000.
    0040.                           -0000.
```

Figure 2.3 CROSS-SPECTRUM OF SINE WAVE

Spectrum of in-phase components of sine wave cross-correlated with cosine wave of same frequency.

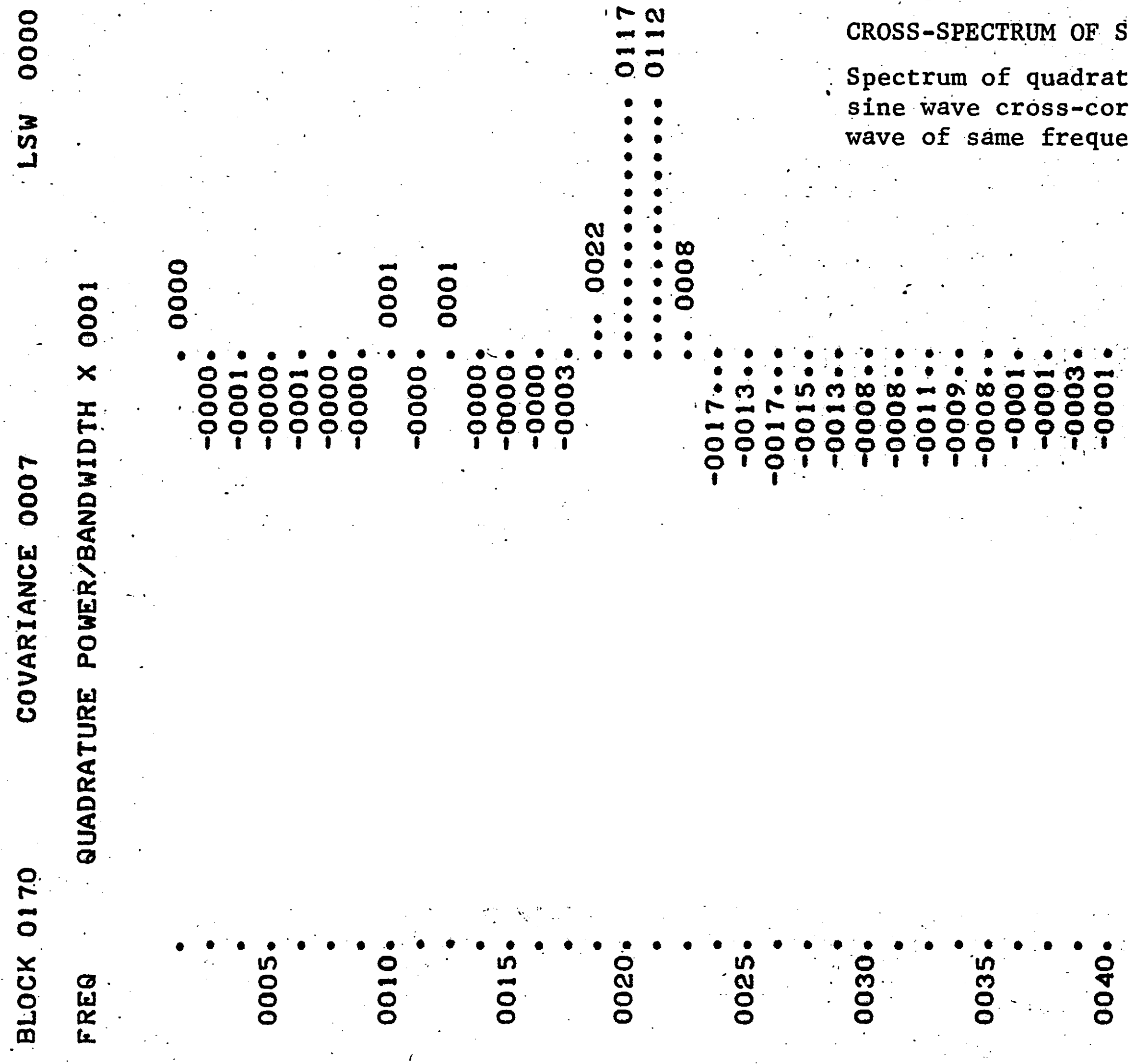


Figure 2.4

CROSS-SPECTRUM OF SINE WAVE

Spectrum of quadrature components of sine wave cross-correlated with cosine wave of same frequency.

8-16

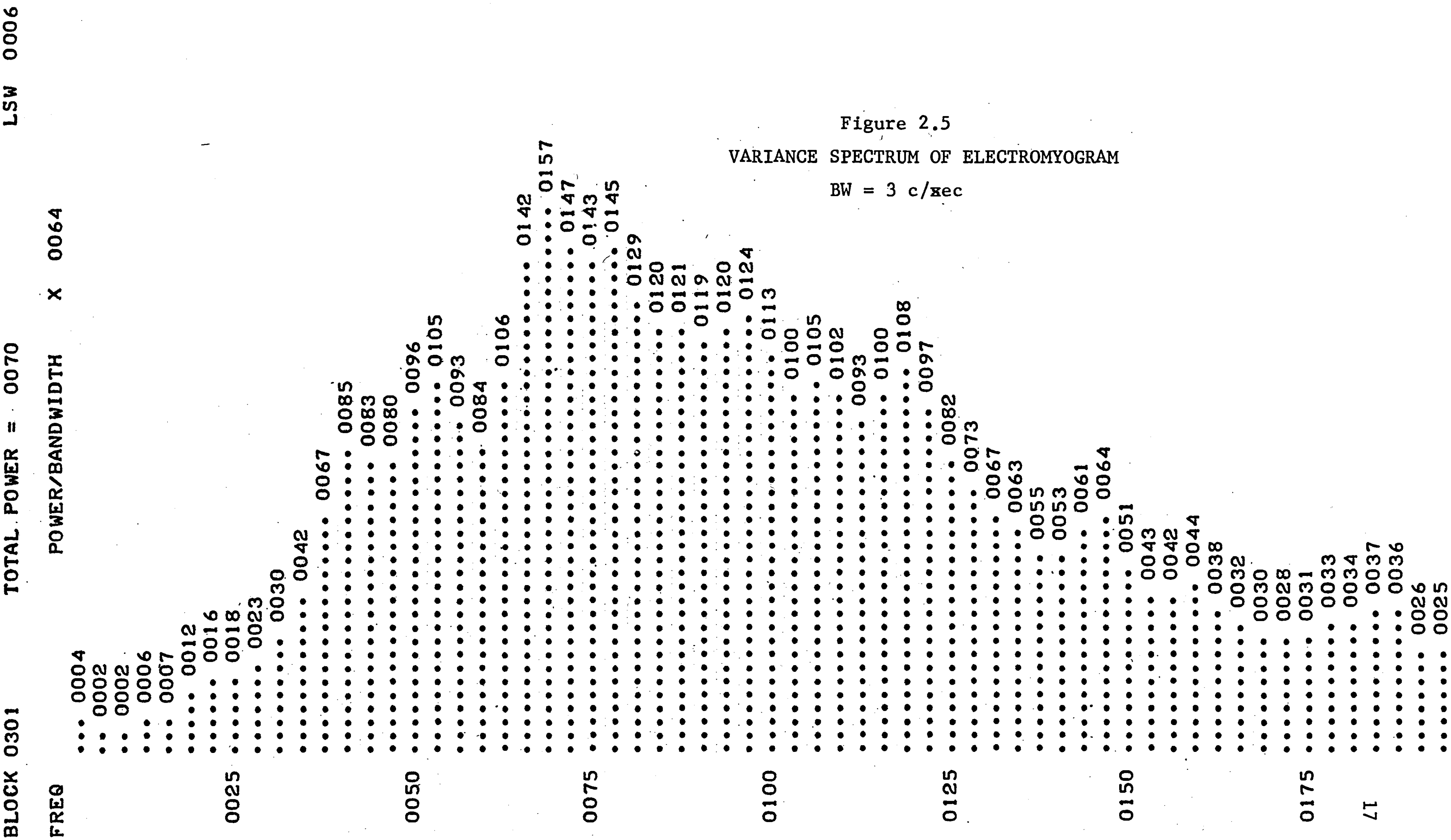


Figure 2.5

VARIANCE SPECTRUM OF ELECTROMYOGRAM

BW = 3 c/sec

COVARIANCE = -0007 (X 16 = -0112 INPUT BITS SQUARED)

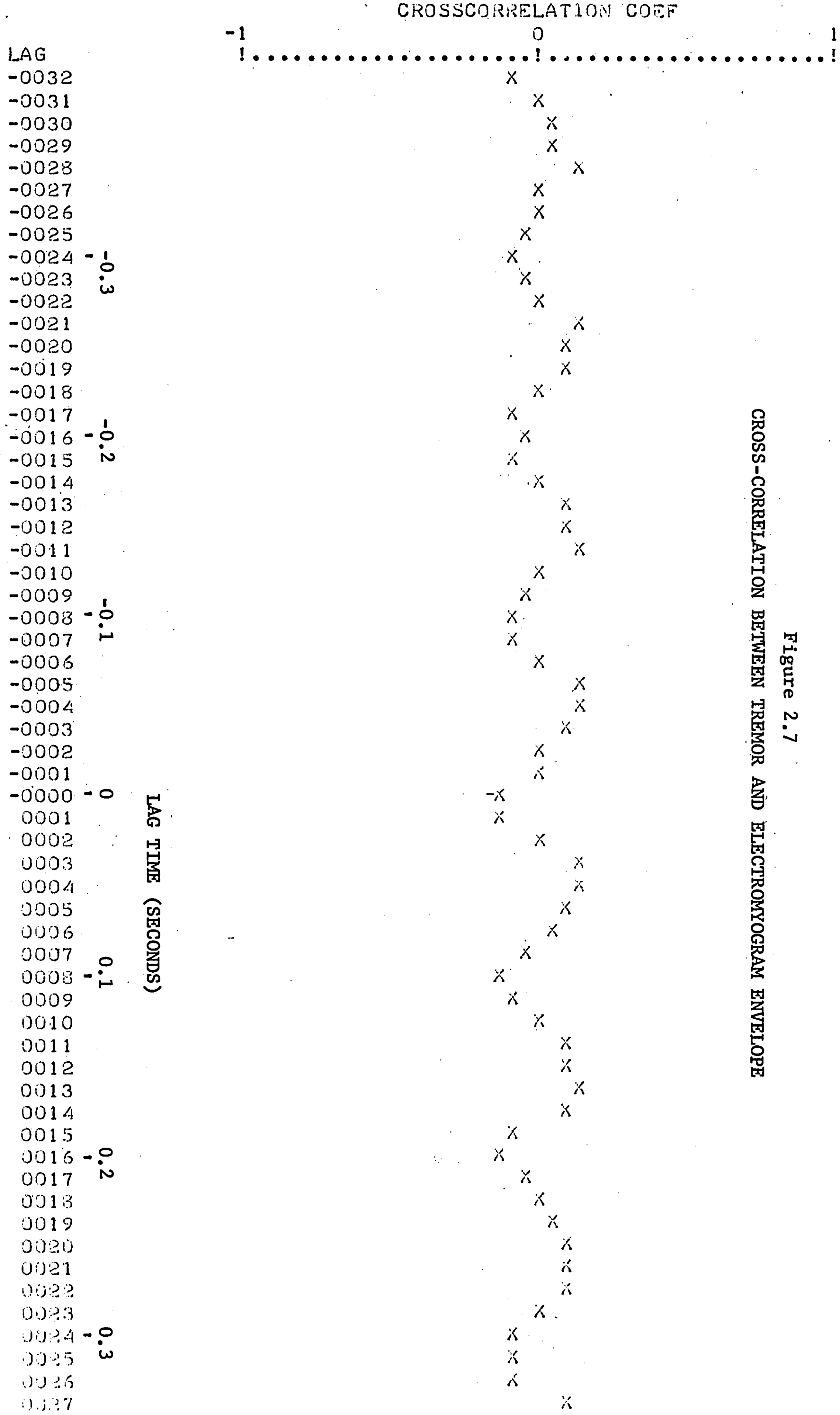


Figure 2.7

CROSS-CORRELATION BETWEEN TREMOR AND ELECTROMYOGRAM ENVELOPE

8-20

```
BLOCK 0200         COVARIANCE-0007                   LSW  0006

FREQ      IN PHASE   POWER/BANDWIDTH X 0064

    •                                     • 0000
    •                                     • 0002
    •                                     • 0003
0005•                                -0003•
    •                                -0007•
    •                               -0014••
    •                  -0113•••••••••••••••
0010•          -0183•••••••••••••••••••••••
    •                     -0089••••••••••••
    •                               -0010••
    •                                -0000•
0015•                                -0000•
    •                                -0001•
    •                                     • 0003
    •                                -0004•
0020•                               -0012••
    •                                -0007•
    •                                     • 0007
    •                                     •• 0009
0025•                                -0005•
    •                               -0014••
    •                               -0010••
    •                                -0001•
0030•                                -0002•
    •                               -0011••
    •                              -0016•••
    •                               -0011••
0035•                                -0002•
    •                                -0002•
    •                                -0006•
    •                               -0012••
0040•                              -0017•••
```

Figure 2.8

CROSS-SPECTRUM FOR TREMOR-ELECTROMYOGRAM

Spectrum of in-phase components obtained from cross-correlating tremor with electromyogram envelope.

8-21

```
BLOCK 0200        COVARIANCE-0007              LSW  0006

FREQ      QUADRATURE POWER/BANDWIDTH X 0064

    .                           -0003.
    .                           -0003.
    .                                . 0001
0005.                                . 0002
    .                           -0003.
    .                           -0006.
    .                                . 0005
0010.                                ... 0019
    .                                .. 0014
    .                                . 0004
    .                           -0004.
0015.                           -0007.
    .                                . 0006
    .                                . 0002
    .                           -0004.
0020.                                .. 0008
    .                                .. 0015
    .                                .. 0010
    .                           -0004.
0025.                          -0011..
    .                           -0003.
    .                           -0001.
    .                           -0007.
0030.                           -0003.
    .                                . 0007
    .                                . 0001
    .                           -0007.
0035.                           -0000.
    .                           -0000.
    .                           -0001.
    .                           -0004.
0040.                           -0006.
```

Figure 2.9 CROSS-SPECTRUM FOR TREMOR-ELECTROMYOGRAM

Spectrum of quadrature components obtained from cross-correlating tremor with electromyogram envelope.

A preliminary cross-correlation function obtained between tremor and the electromyogram envelope is given in Figure 2.7. This shows evidence of a correlation between these two quantities with a period of about 0.1 sec. When this is subjected to a cross spectral analysis the frequencies at which correlation exists can be seen. Figures 2.8 and 2.9 show that for this particular case most of the correlation was at the 10 c/sec band. It is these kinds of records we wish to collect for a variety of experimental conditions.

D. Quantitation of Microcirculation

An important factor in determining the supply of nutrients to the cells in a vascular bed is the resistance to blood flow for that bed. Flow and pressure are usually measured in large arteries so that the estimates of resistance are macroscopic. Actually the total resistance is a complex summation of a large number of elemental resistances, those of the arterioles and capillaries. Microscopic observations of this microcirculation indicate that the flow through different vessels may be independent, the individual vessels opening and closing apparently at random. Also the flow through any one path does not appear to be continuously graded, but more nearly of an all-or-none nature. This suggests that the total resistance may be graded (to some extent) by changes in the number of paths open at any instant and by the duration of the open periods for any one micro-vessel. Thus a description of the mechanism of change of the total vascular bed resistance may amount to a shift of a statistical distribution in such a direction as to change the number of paths open at any one instant without necessarily saying what will happen in any one path. There is no quantitative experimental evidence by which one can evaluate the statistical nature of the microcirculation.

For many years the microcirculation has been viewed and filmed, particularly in the bat wing. Movies show the individual red blood cells streaming or stationary. However, because of poor contrast, observation of individual film frames does not distinguish between a cell and its background. The movement of cell with respect to background is what the eye can detect. For this reason there have been no manual measurements to determine whether the flow is graded or whether the flow, no-flow states of different vessels are independent or correlated.

The question of how to measure the statistical nature of the microcirculation had been discussed in our laboratory group for a number of years. As a direct result of becoming familiar with the LINC and seeing those seven analog input jacks, one graduate student in physiology, Patrick D. Harris, had an inspiration about how such measurements might be done automatically. He is working on the technique as part of his doctoral research problem.

As the technique stands now, (with its unresolved imperfections) movies are taken of the microcirculation in the bat wing using a microscope with a power of 645X at film speeds of 16 frames/second. After development, this film is projected on a translucent screen giving an overall optical magnification in which an 8 micron red blood cell projects as about 8 millimeters. High-speed photoconductive cells are placed on the projected image of a vessel and on a background (non-vascular) area. Differences in transmitted light intensity are sensed by the photocells and fed to the analog inputs of the LINC. The differences in transmitted light with and without a red blood cell are only slightly greater than the fluctuations in the nonvascular background. The major task at the moment is to obtain positive evidence that the differences in photocell voltages sampled by the LINC are the results of a passing red blood cell.

Once the sensing of the presence of a red blood cell by one photocell can be perfected, the measurements of several photocells can be combined along with computer logic to sense

1) spacing between individual cells
2) velocity of individual cells
3) the duration of flow, no-flow states
4) probability of flow state for any particular vessel
5) joint probabilities for the states of a vessel versus states of other vessels.

If these latter two probabilities are identical, it is evidence that the flow states in the different vessels are independent.

Since casual observations have placed the durations of capillary states in the range of 1 to 30 seconds with a majority falling into a 6 to 10 second range, it is felt that 400 to 500 feet of film of a particular area would be needed in order to have confidence in the probability determinations. For 40 frames per foot taken at 16 frames/second, a minimum of 100 state changes would be recorded for 400 feet of film. The computer program has been designed to take one photocell sample for each projector flash. The fact that there are three flashes for each film frame provides some redundancy for evaluating the variability of all instrument components beyond the film, e.g., projector vibration, projector light intensity changes, and photocell noise. For the present program, which stores a photocell sample for each flash (3 per frame), a conventional reel of LINC tape will store the information for 500 feet of film.

3. FUTURE PLANS

Thesis problems for two graduate students and one grant-supported research problem are dependent upon the continuing use of the LINC for the next one to three years. In addition, a former member of the LINC group at Stanford has recently joined the Northwestern Medical School faculty and wishes to make some use of the LINC here. His plans are given in detail in this section.

A. Student Predoctoral Research Problems

Robert Stiles will be using the LINC to obtain variance spectra and cross-spectra in the next year as part of his study of the mechanism of finger tremor. During this period particular attention will be placed upon cross-correlation of finger tremor and other fluctuations and their in-phase and quadrature spectra.

Patrick Harris will be using the LINC to sense movement of red blood cells as projected on photocells. After development of the technique, statistical information about the flow and no-flow states of the microcirculation will be collected, as described in a previous section.

Other students anticipated for future years, will very probably choose research problems which can take advantage of this computer facility.

B. Computation of Transfer Functions Between Statistical Signals

James Randall has plans to use the LINC as the primary means of data analysis for his research grant "Stochastic Properties of Physiological Time Series", USPHS Grant HE 08516-02, which will run through 1967. The purpose of this study is to develop techniques of obtaining the transfer function between two time-varying quantities using naturally occurring variations as the forcing and the response. The tremor studies constitute one specific application of these general techniques. A discussion of the mathematical relations is given in Appendix III.

Measures of coherence (phase consistency) between two signals may be considered as a correlation coefficient which is a function of frequency. A conventional correlation coefficient may not sense the high correlation between two signals within a narrow frequency band because of the presence of large uncorrelated fluctuations at other frequencies. Coherency is an index of the consistency of the phase between a given frequency component in two signals. When the coherence is high it may be possible to represent the relationship between the two signals by a transfer function with gain and phase as a function of frequency. The cross-spectral techniques which are being developed on the LINC will permit this to be obtained even in the presence of large amounts of incoherent noise in the signals.

C. Neuropharmacology Studies

Dr. Charles Berry, Assistant Professor of Pharmacology, has had previous training on the LINC at the Stanford University Medical School. He is interested in using the Northwestern University LINC to reduce and analyze electroencephalographic and evoked potentials obtained in his studies of drugs and neural system interactions.

His present effort in neuropharmacology is directed toward measurement and evaluation of "recovery curves" derived from paired responses evoked from the cortex of unanesthetized cats. If pairs of identical stimuli having interstimulus intervals of 16-400 msec are presented to the cortical surface, characteristic alterations in the waveform envelope of the second local cortical response occur when compared with the first response. When the ratio of "second response"/"first response" is plotted against increasing interstimulus time intervals, recovery patterns emerge which are interpreted as a measure of the excitable responsitivity of the local cortical area. Prior experiments have shown that visual, auditory, somatosensory, association, and motor cortexes each exhibit distinct area-specific recovery patterns.

Until the present time the data upon which these results are based has been recorded photographically from a cathode ray oscilloscope and subsequently projected from the film for measurement purposes. It has been proposed that use of a magnetic tape recorder in conjunction with LINC might materially reduce this data reduction bottleneck and for this purpose an Ampex SP-300 recorder has been acquired. By utilizing magnetic tape facilities and proper programming it is anticipated that LINC could be used for the following purposes:

1. Constructs of averaged response data could be generated for display and computational purposes. In the experiments described above only amplitude measurements have been utilized as indicators of neuronal recovery functions. However, other measures such as latency changes, the area within the response envelope, or alterations in the waveform pattern may be more appropriate. Analytical comparisons of this nature are very tedious without programmed computational assistance.
2. The construction of recovery curves would be based upon the appropriate data generated in 1) above.

Additional anticipated uses of LINC not directly related to the research plan outlined above would include auto- and crosscorrelation analyses of electroencephalographic data recorded from focal discharing epileptogenic lesions and "mirror foci" prior to and following pharmacological manipulation. It is also planned to utilize the flexible capabilities of LINC in attempting to correlate characteristic waveform responses generated in the EEG with the stages of behavior which develop in cats during the successive stages of learning a problem in conditioned behavior

4. TRAINING PROGRAM

The strongest aspects of the program are the intangible ones which can not be measured in terms of numbers of routines or the length of the bibliography. The single outstanding feature was the one month spent at M. I. T. The intensiveness of this period, without outside distractions, and the dedication of the C. D. O. staff to the task combined to make this a unique experience. The first-hand contact with computer logic and machine language broke down a barrier which assembly and compiler programs have always erected. Having the computer in the laboratory has stimulated our own quantitative thinking, and that of our students, to a greater extent than any amount of computer center service could ever do.

Looking back on the one-month training period we can realize that many of the methods served purposes that were not then evident. The holding off of the LAP 3 until the last week forced the group to become familiar with machine language to such an extent that this has helped in debugging programs. On the other hand, the hours and hours of reading timing diagrams for computer logic operations has not been productive. Since this information was so detailed and was not put to use, it soon slipped away. If it had been needed to keep the computer running this past 18 months no one would have had the time to master these intricacies. One can not do good programming without some knowledge of machine logic, but there is a limit to how much detail will be useful.

The digital input-output capabilities of the machine are still not fully appreciated.

There were many programming tricks and formats that have become individualized which complicates the interchange of programs between LINC users.

5. COMPUTER PERFORMANCE

During the 18 months that we have had the LINC there has been only one period of 3 to 4 weeks when time was spent on servicing the machine rather than on normal operation. The problem then (June 4, 1964 through June 27, 1964) was eventually minimized, but never completely eliminated.

In retrospect it is now appreciated that the difficulty was marginal operation of the tape heads. This was manifested as a failure to get check sums to match data sums on the WRC instruction. At times one could not use the LAP 3 because of a continual rocking on one block. The single operation of cleaning the heads at the time of marking a new tape proved to make the most difference. Other factors as head alignment, grounding, deburring the tape guide shoes, etc. contributed marginal improvement in the tape signals. At the present time tapes are always marked on the unit which gives the greatest signal level. Even with these precautions there are many times when the tape must make several passes to get a correct check sum on WRC. After filling the tape with 0's with WRI instructions, the checking is frequently hung up. Once this difficulty was understood it has not been a serious limitation.

The original input preamplifiers in the data terminal box were non-linear, but this was corrected by a modification suggested by Bill Simon. This and the above tape troubles have been the only difficulties that we have experienced. In fact, since installing the second part of memory in November of 1964 the dust covers have not been off the rack nor have the power supply voltages been adjusted (or even checked). During this period the machine has been used for a total of over 450 hours. This suggests that the design is "physiologist-proof".

We are not confident of the analog-to-digital ladder adjustments and fear that they may introduce systematic errors in our data collection. We would be more confident if there were some objective, simple method of testing and adjusting this.

The potentially high sampling rate fills the memory in a short time, limiting computation to batches. Our method of overcoming this, described in an appendix, requires that the signal be recorded on analog tape. Perhaps there is some other solution.

Since purchasing the teletype machine the camera has not been used. The camera is awkward to use, the film is expensive, and the manuscripts too small to edit. On the other hand, the teletyped manuscripts are very popular since they can be written upon and also give an explicit memory location for debugging programs. Similarly, graphs of data which are teletyped are easy to file and read. So far we have had no mechanical difficulties with the teletype as an output device.

We have had no difficulties with the keyboard reading in wrong characters as some have had. The keyboard has always been sluggish when the machine is first turned on, but this has not gotten worse over the past 18 months so it is not considered serious. The teletype is not reliable as an input device and is too slow for convenient entry of LAP 3 manuscripts.

At this writing the light behind the exam push button is intermittent, but this button functions normally. Also there is an occasional momentary intense whiteness to the 6-bit light of the accumulator. This is most often seen when entering the LAP 3 from the left switches and occurs just before the accumulator reads the block number 300.

Only one or two of the input potentiometers are ever used at a time in our applications. It is possible that the rest of them could find use as additional analog inputs, though this has not been a limitation for us.

Because of the nature of our computations, and the number of products being summed, 11-bits has not been sufficiently large for handling temporary sums. Is there any possibility of automating some of the double precision arithmetic steps?

The input preamplifiers in the data terminal box have an adequate input sensitivity. However, since laboratory sources have wide ranges of output voltages, it would be advantageous to have more flexibility in gain adjustment at the computer. At the present time this range is from about 4.5X to 5.5X.

There should be a service organization backing up the LINC. We question whether it would be efficient use of our time in trying to trace down a subtle malfunction. Sooner or later something is going to wear out, and it frightens us to be so highly dependent upon the machine and to be cut off from service help if we need it.

In our experience, which has been fortunate, there would have been no difficulty in handling computer operating costs from existing research grants. There is some question whether one could justify the LINC's initial expense as part of a single-investigator's operation.

6. BIBLIOGRAPHY

There have been no publications.

APPENDIX I SYNCHRONIZING ANALOG TAPE WITH LINC SAMPLING

The combination of LINC sampling rate and memory size means that analysis may be limited to batches of a few seconds of data. To be sure that the batches were long enough, i.e., that the data statistical variability would not be much different for larger batches, a scheme was developed for transferring data on analog tape into the LINC memory and then onto LINC tape by making two passes of the analog tape. A total of 1.3 x 10^5 8-bit samples, corresponding to 20 minutes of sampling of a single channel at 100 samples/second, can be transferred to one spool of LINC tape. In principle this could go on indefinitely by switching from one tape unit to the other after all blocks are filled.

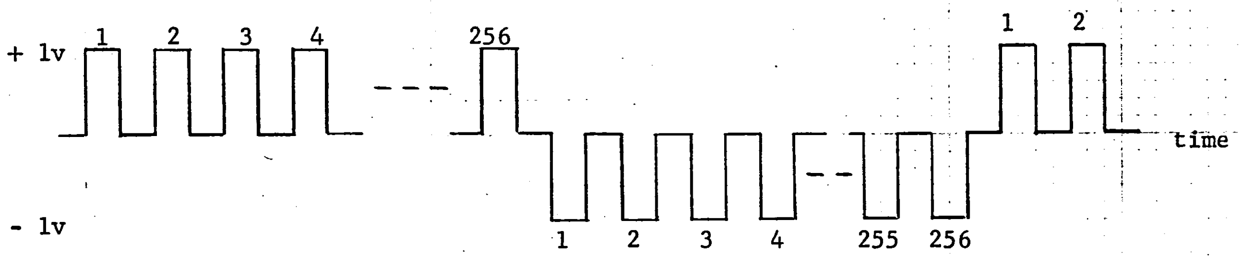
Our scheme consists of putting marking pulses on one channel of the analog tape at the time that the data is recorded on that tape. These pulses are supplied by a coder-decoder designed and constructed by the Northwestern University Biomedical Instrumentation Laboratory. It is mounted on the mobile Ampex analog tape rack. Output of the coder consists of alternating sequences of 256n pulses of +1.5 v and 256n pulses of -1.5 v, n corresponding to 1, 2, or 4 at the option of the operator. The pulse rate can be selected at 50, 100, or 200 samples per second or can be driven by an external oscillator. The maximum pulse rate was limited by the analog tape speed. These pulses determine when the LINC will take and store data samples from the analog tape channels.

During playback the analog tape channel containing marking pulses is fed through the decoder which stores them to clean pulses of +1 v and -1 v with equal off and on intervals. These pulses and the analog data channels are fed into the LINC analog input jacks. A program was written which stores data values in the LINC memory each time a mark pulse goes from

0 to +1. After 256n of these samples, the contents of n quarters of memory are transferred to n successive blocks on the LINC tape. The program then ignores the negative marking pulses and takes its next sample on the next positive pulse, the first of a new train of 256n. This continues until one-half of the LINC tape is filled with samples taken for each positive marking pulse. The analog tape is then rewound to the same starting place, and the LINC program changed to take samples on the change from 0 to -1 v in the mark channel. The samples are transferred to tape during the positive pulses.

The LINC program was designed to resolve certain possible ambiguities which might occur by starting the analog tape up at some arbitrary point. For example, it has to ignore starting the sampling in the middle of a train of 256n pulses of the appropriate polarity and must start the sampling during the second pass in phase with that of the first pass. The test to see whether data samples have been missed has been to display two successive blocks of tape on the scope and look for discontinuities between the blocks.

Figure 3.0 Marking pulses recorded on 1 channel of analog tape, data on other channels



FIRST PASS OF ANALOG TAPE (SNS SW DOWN)

LINC ignores → | ← LINC samples data channels at start of each pulse, samples stored in memory → | ← LINC transfers memory to LINC tape then waits → | ← LINC samples and stores, etc.

SECOND PASS OF ANALOG TAPE (SNS SW UP)

LINC ignores → | ← LINC ignores → | ← LINC SAMPLES and stores data → | ← LINC transfers memory to LINC tape, waits

APPENDIX II SIMULATION OF FORCING A DAMPED SYSTEM WITH STATISTICAL PULSES

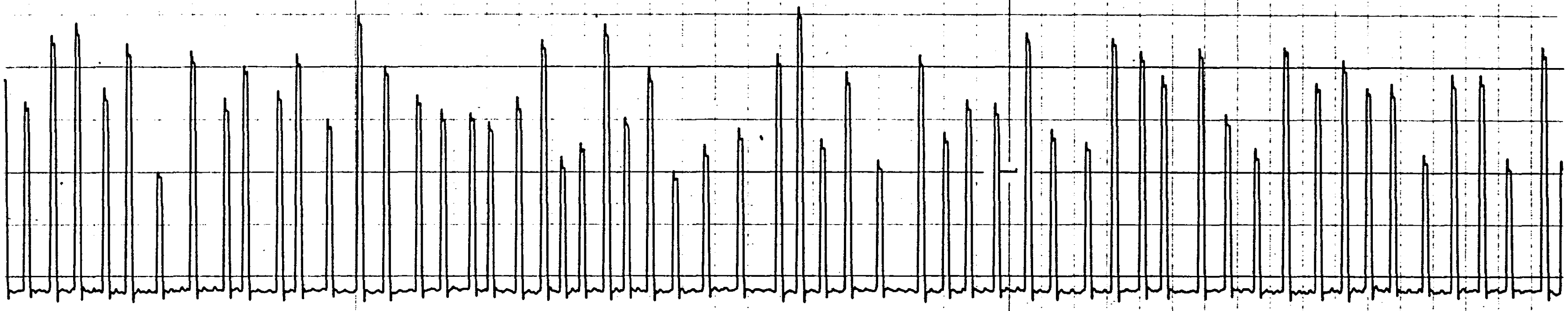
One explanation for the 10 c/sec component of the tremor spectrum has been that it corresponds to the resonant frequency of a damped mechanical system which is being forced by the statistical manner in which muscle units may fire. It was not intuitively evident how changes in the statistical distributions of the forcing would be reflected in the response from such a mechanical system. The LINC was used to generate a statistical forcing which was applied to an analog computer model of a second-order system with a natural frequency of 10 c/sec and damping coefficient of 0.15 critical. This is not presented here as a serious model for the tremor, but as an illustration of the flexibility of the LINC.

The LINC generated a sequence of pulses of 20 milliseconds duration of amplitudes which followed some statistical distribution and at pulse-to-pulse intervals which followed some statistical distribution. Two different distributions were used, a Gaussian and a pseudo-random one with a flat distribution. The numbers determining pulse amplitude and pulse-to-pulse interval according to a Gaussian distribution were obtained by sampling the input noise of two Tektronix 122 preamplifiers connected in cascade. The output of the second amplifier was about ± 1 volt, the amplitude was Gaussian distributed, and the variance spectrum was flat over the amplifier pass band. Figure 3.1 shows the forcing pulses with such a distribution and the simulated output of the mechanical system.

The numbers determining pulse amplitude and pulse-to-pulse interval as distributed according to a flat distribution were obtained from a pseudo-random number generator supplied by William Simon. This program was tested and found to give a flat distribution. Figure 3.2 gives the appearance of the forcing pulses with these statistical characteristics and the resulting output of the simulated mechanical system.

Figures 3.3 and 3.4 show variance spectra of the outputs of a simulated 10 c/sec damped system driven by statistical pulses with flat and Gaussian distributions. Note the difference in the center frequencies of the spectral peaks. This illustrated to us the possible significance of the forcing on the output spectrum. This was further illustrated by shifting the resonant frequency of the damped system to a lower frequency and measuring the output spectra for the same two forcings. In Figure 3.5, in which forcing was by pulses with a Gaussian distribution, the spectral peak at about 12 c/sec is a manifestation of the mode of the pulse-to-pulse intervals which was near 1/12 sec. The other spectral peak reflects the damped system. However, for the flatly distributed pulse-to-pulse intervals, in which there is no mode, the only peak is at the resonance of the simulated mechanical system (Figure 3.6). Once again, the point is not that these are simulated tremor spectra, but that the LINC allowed us to program a rather special pulse generator and to experiment mathematically.

Figure 3.1 GAUSSIAN DISTRIBUTION OF AMPLITUDES AND INTERVALS



OUTPUT OF SECOND-ORDER SYSTEM
(Natural freq. = 10 c/sec, damping coef = 0.15)

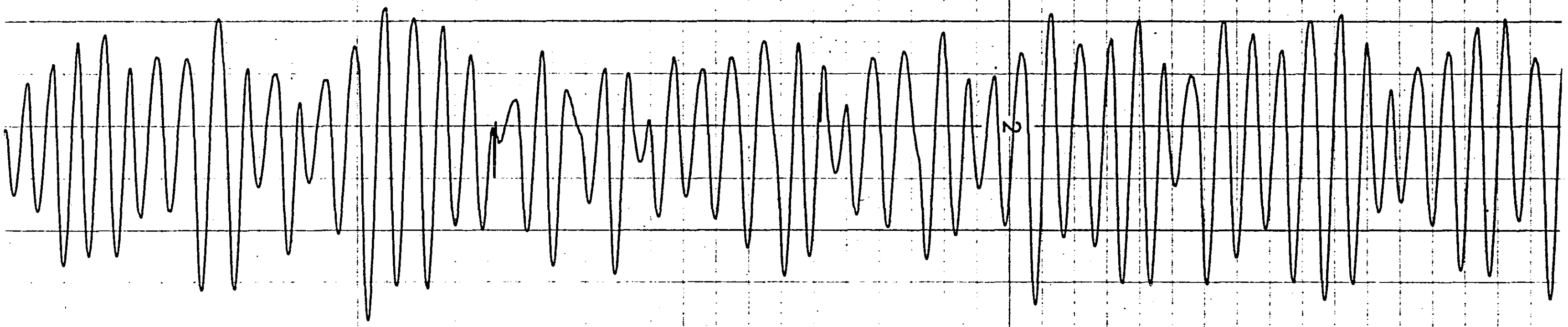
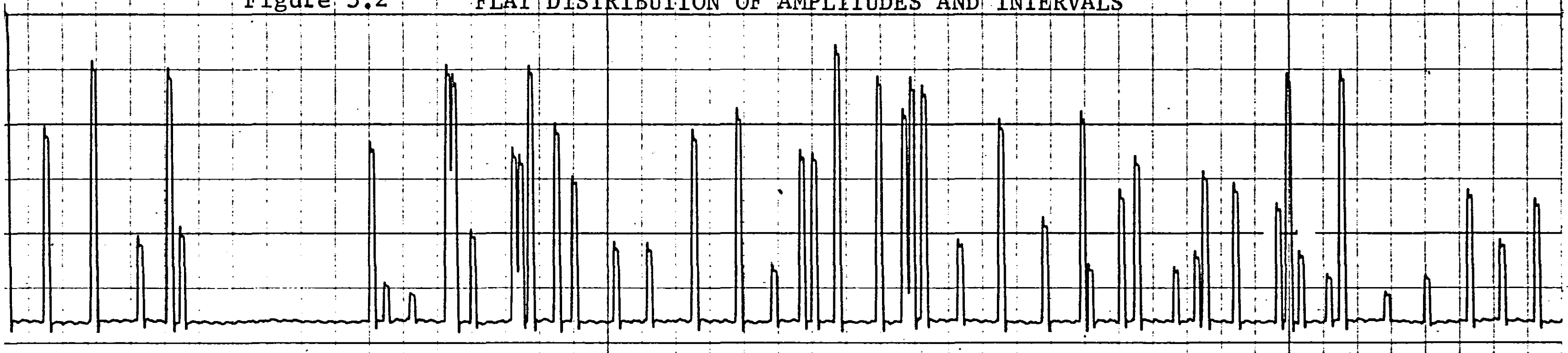
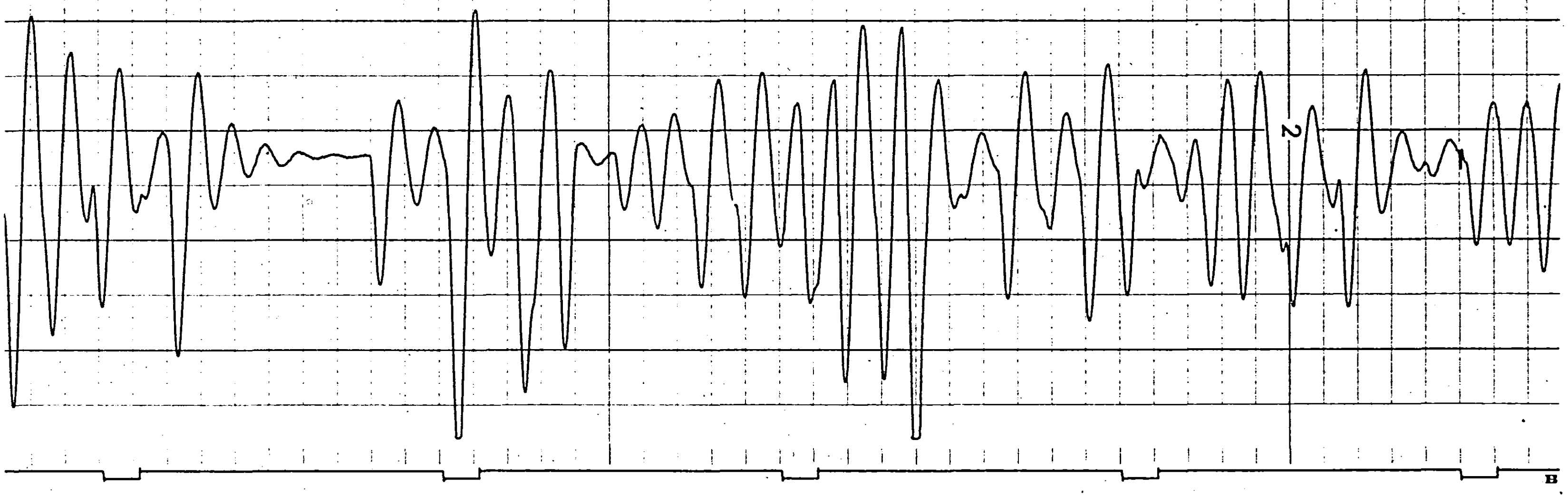


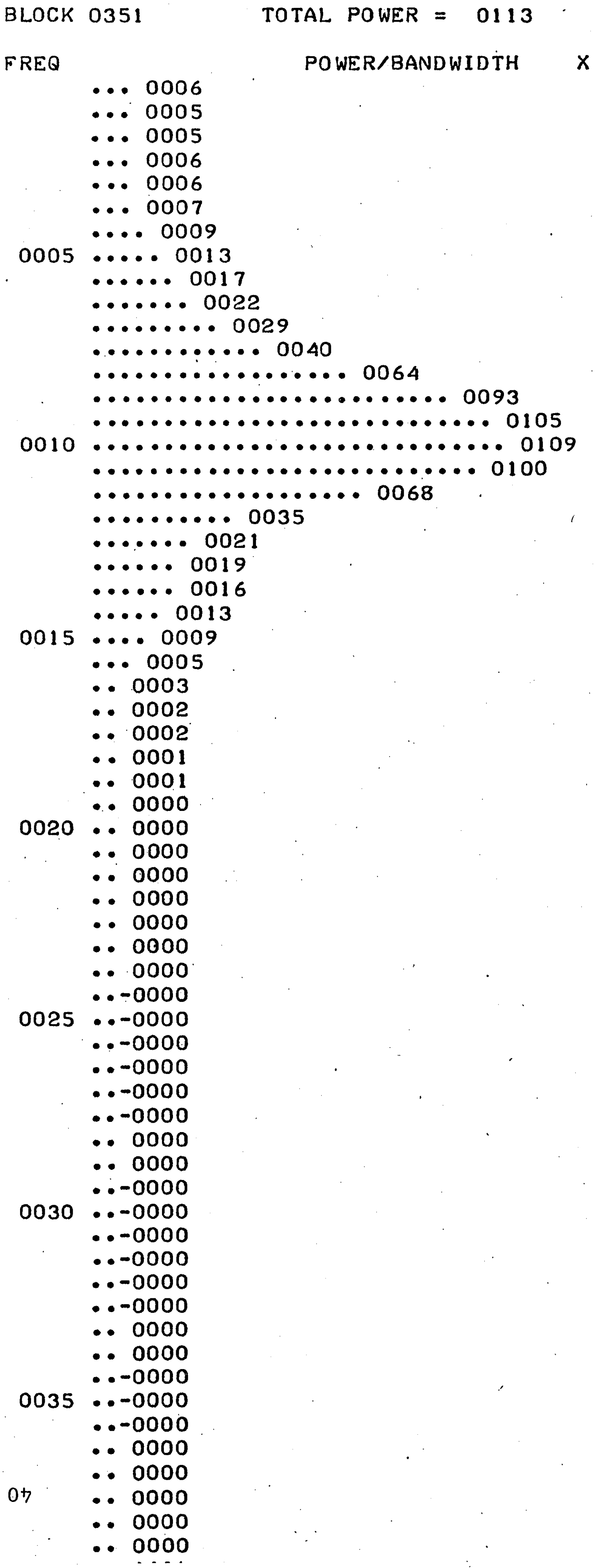
Figure 3.2 FLAT DISTRIBUTION OF AMPLITUDES AND INTERVALS



OUTPUT OF SECOND-ORDER SYSTEM
(Natural freq. = 10 c/sec, damping coef. = 0.15)



4-8



```
BLOCK 0351        TOTAL POWER =  0113              LSW  0003
FREQ                  POWER/BANDWIDTH    X  0008
      ••• 0006
      ••• 0005
      ••• 0005
      ••• 0006
      ••• 0006
      ••• 0007
      •••• 0009
 0005 ••••• 0013
      •••••• 0017
      ••••••• 0022
      ••••••••• 0029
      •••••••••••• 0040
      •••••••••••••••••• 0064
      ••••••••••••••••••••••••• 0093
      •••••••••••••••••••••••••••• 0105
 0010 •••••••••••••••••••••••••••• 0109
      ••••••••••••••••••••••••••• 0100
      ••••••••••••••••••• 0068
      •••••••••• 0035
      ••••••• 0021
      •••••• 0019
      •••••• 0016
      ••••• 0013
 0015 •••• 0009
      ••• 0005
      •• 0003
      •• 0002
      •• 0002
      •• 0001
      •• 0001
      •• 0000
 0020 •• 0000
      •• 0000
      •• 0000
      •• 0000
      •• 0000
      •• 0000
      •• 0000
      ••-0000
 0025 ••-0000
      ••-0000
      ••-0000
      ••-0000
      ••-0000
      •• 0000
      •• 0000
      ••-0000
 0030 ••-0000
      ••-0000
      ••-0000
      ••-0000
      ••-0000
      •• 0000
      •• 0000
      ••-0000
 0035 ••-0000
      ••-0000
      •• 0000
      •• 0000
 40   •• 0000
      •• 0000
      •• 0000
```

Figure 3.3 VARIANCE SPECTRUM OF OUTPUT OF 10 C/SEC DAMPED SYSTEM

(Flat distribution of forcing pulses)

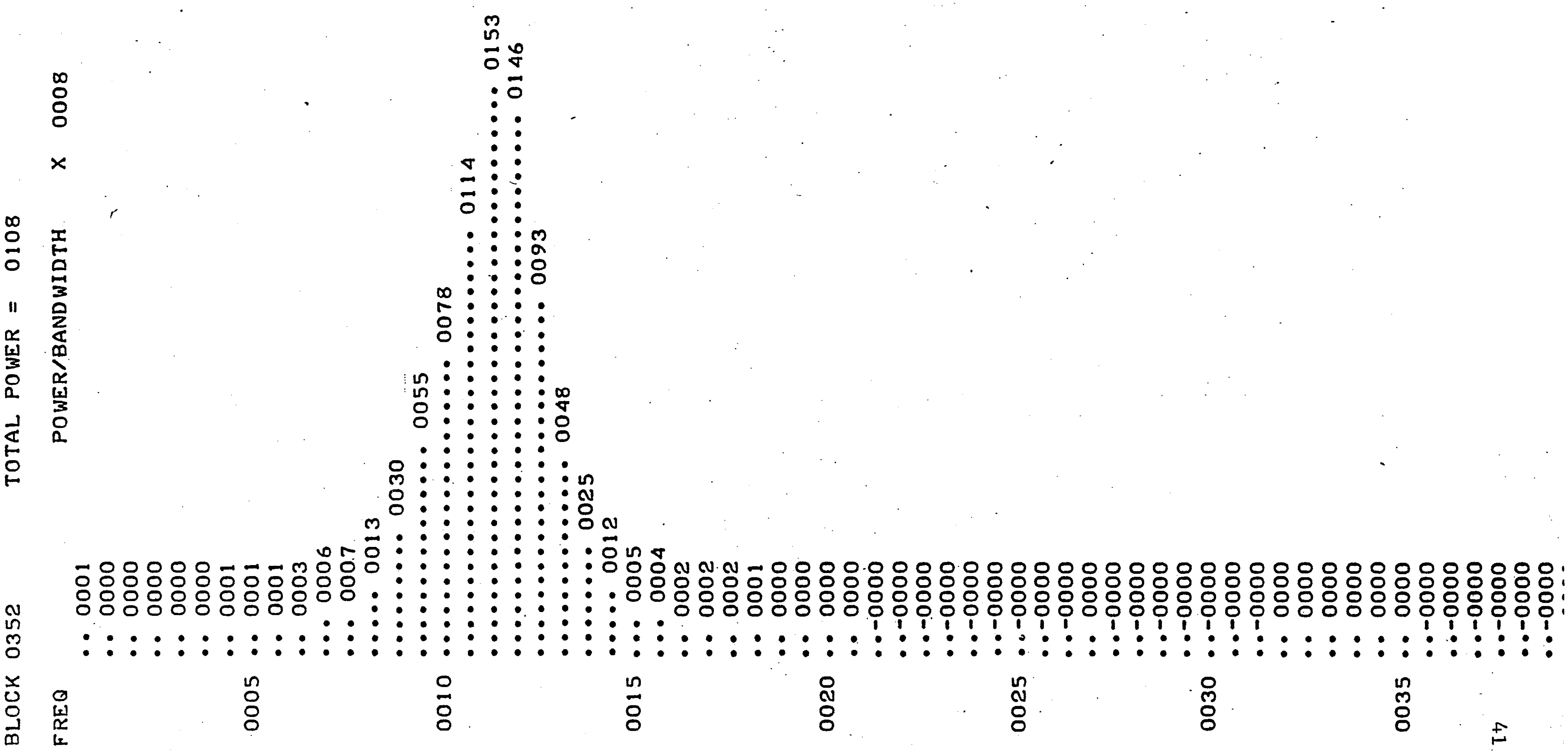


Figure 3.4 VARIANCE SPECTRUM OF OUTPUT OF 10 C/SEC DAMPED SYSTEM

(Gaussian distribution of forcing pulses)

14-8

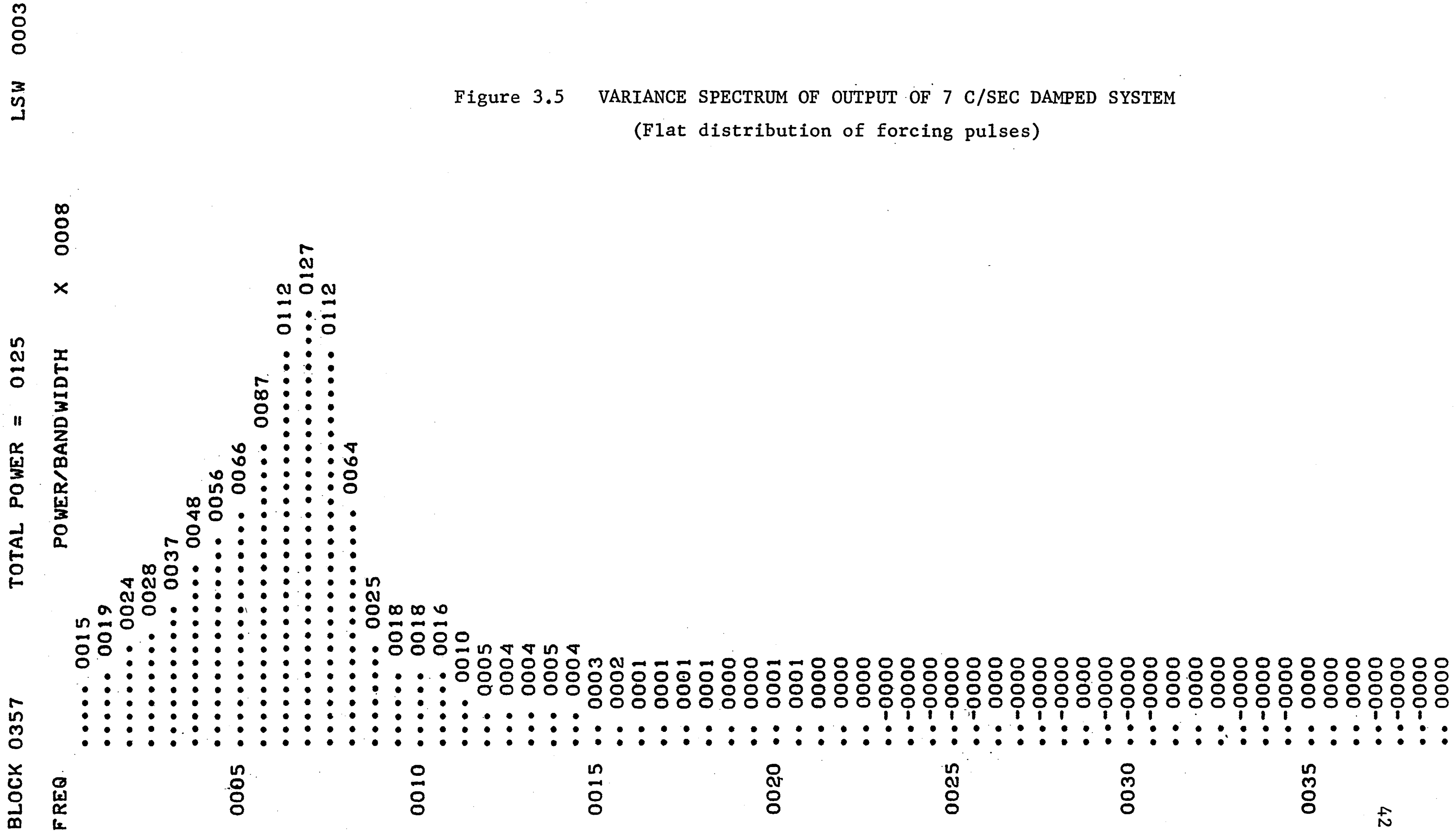


Figure 3.5 VARIANCE SPECTRUM OF OUTPUT OF 7 C/SEC DAMPED SYSTEM
(Flat distribution of forcing pulses)

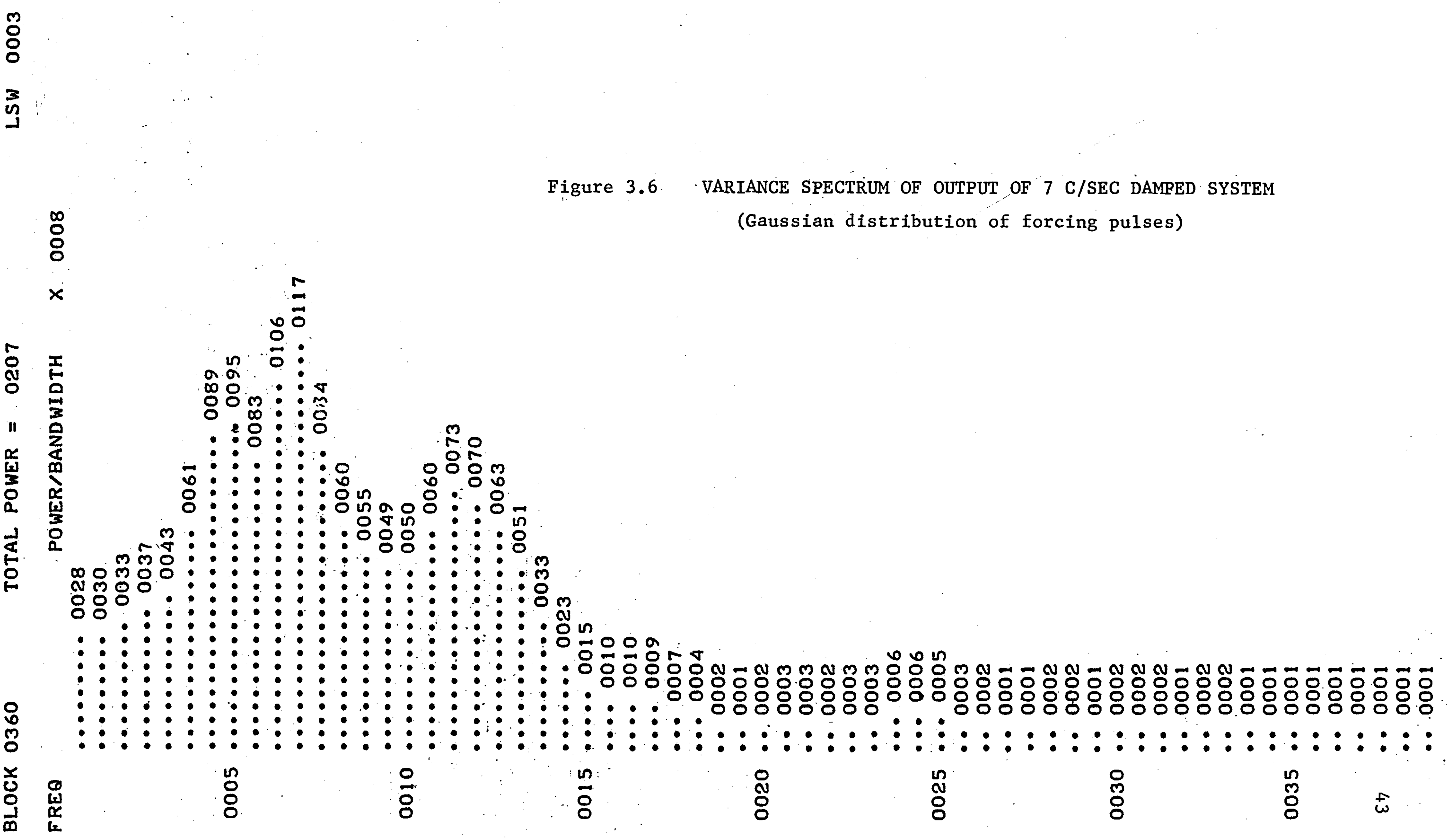


Figure 3.6 VARIANCE SPECTRUM OF OUTPUT OF 7 C/SEC DAMPED SYSTEM
(Gaussian distribution of forcing pulses)

8-43

APPENDIX III MATHEMATICAL EXPRESSIONS FOR THE CORRELATION AND SPECTRAL FUNCTIONS

A. Autocorrelation Function and Autocorrelation Coefficient

The autocorrelation function and the variance spectrum are mathematical transformations of one another, but they each emphasize different aspects. The autocorrelation function, having lag time as the independent variable, emphasizes the correlation between a time series and the same time series an interval of time later. This function was computed by us as an intermediate step to finding its Fourier transform, the variance spectrum. The autocorrelation function for the time series having discrete values of x_i was computed as

$$R_{xx}(\tau) = (1024)^{-1} \Sigma (x_i - \bar{x}) (x_{i+\tau} - \bar{x})$$

for i = 1, 2, . . 1,024 and $\tau = 0, 1, . . 64$. In practice 1,088 6-bit samples were taken and stored at time intervals of Δt sec. The autocorrelation function was calculated for lag times of $(\tau\Delta t)$ sec. The value of the mean $\bar{x}$ was computed from only the last 1,024 samples for ease in division. The value of R(0) is the estimate of the variance, or the square of the estimated standard deviation, $R(0) = s^2$.

The normalized autocorrelation function, the autocorrelation coefficient, was used for graphing and display purposes. This is given as

$$r_{xx}(\tau) = R_{xx}(\tau)/R_{xx}(0)$$

It is unity for lag = 0 since there is perfect correlation between the data samples x_i and x_{i+0}.

B. Variance Spectrum

The variance of a finite segment of a continuous times series can be considered as an estimate of a statistical parameter of that time series. Since the variable is a time series, the variance can be analyzed into a frequency spectrum, the integral of which is the variance. Just as the variance of different finite samples of the time series will differ, so will the estimates of the amount of the variance in a given band width. However, these variance estimates will tend to be statistically distributed so that confidence limits for the "true" variance can be set. Thus spectra from different observations can be compared to find the probability that they are from the same or different time series.

We have computed the variance spectrum from the autocorrelation function by the relation

$$G(N) = (32)^{-1}[\, R(0)/2 + \Sigma\, R(\tau) \cos \tau N(360/128) + R(64)/2 \cos N\, 180^{\circ}]$$

for $N = 0, 1 \ldots 64$ and $\tau = 1, 2 \ldots 63$. $G(N)$ is the raw spectral estimate of variance within a band $(128\ \Delta t)^{-1}$ c/sec wide and centered at a frequency of $N\ (128\ \Delta t)^{-1}$ c/sec. This variance per bandwidth will sum to the total variance or

$$R_{xx}(0) = \Sigma\, G(N)$$

where $N = 0, 1 \ldots 64$. This is the variance within the frequency range of 0 to $(2\ \Delta t)^{-1}$ c/sec. Tests were made for aliasing.

Because most natural phenenomena show very irregular spectra, the usual practice is to compute a smoothed spectra by finding a running average according to

$$g(N) = 0.25\ G(N-1) + 0.50\ G(N) + 0.25\ G(N+1)$$

for $N = 0, 1 \ldots 64$ and letting $G(-1) = G(+1)$ and $G(65) = G(63)$.

C. Cross-correlation Function

The statistical relationship between two time series x and y can be measured by computing the simple correlation function coefficients between paired samples from the two time series. However, tests of significance for such correlation coefficients are based on the hypothesis that different observations of x (and of y) are independent of one another. The presence of autocorrelation values other than 0 means that such observations are not independent. It is possible to test the significance of the relation between x and y by cross-spectral analysis which allows for the autocorrelation in each. Changes in x may not be accompanied by simultaneous changes in y, but may be accompanied by corresponding changes in y some time later. The cross-correlation coefficient, the correlation between x and y as a function of a lag time, would then have high correlation at the corresponding lag time. Caution must be exercised in attaching significance to large correlation coefficients.

We have used the cross-correlation function as an intermediate step in computing the cross spectra. This was computed as

$$R_{xy}(\tau) = (1024)^{-1} \quad \Sigma\,(x_i-\bar{x})(y_{i+\tau}-\bar{y})$$

for i = 1, 2 . . 1,024 and $\tau = -32, -31, . . 31, 32$. In practice 1,088 pairs of 6-bit samples of x_i and y_i were taken at a time increment of Δt sec. The means of $\bar{x}$ and $\bar{y}$ were computed for only the last 1,024 samples. The standard deviations of x and y were also estimated as

$$s_x = [\Sigma\,(x_i-\bar{x})^2/1024]^{1/2}$$

$$s_y = [\Sigma\,(y_i-\bar{y})^2/1024]^{1/2}$$

For plotting and display, the cross-correlation coefficient was used,

$$r_{xy}(\tau) = R_{xy}(\tau)/s_x s_y$$

This could have maximal values of +1 or -1 at lag times corresponding to the time it would take for a change in x_i to be manifested as a change in $y_{i+\tau}$. For $\tau = 0$ this becomes a correlation coefficient between x_i and y_i.

D. The Cross-spectra

Since the autocorrelation function was an even function in which $R(\tau) = R(-\tau)$, the Fourier transform contained only the cosine coefficients. However, the cross-correlation function is not symmetrical about $\tau = 0$ and its Fourier transform will, in general, contain both cosine and sine spectra. For ease in computation we resolved the cross-correlation function into even and odd component functions and then found the respective cosine and sine spectra.

The even cross-correlation coefficient was computed from

$$C_{xy}(\tau) = 0.5[R_{xy}(\tau) + R_{xy}(-\tau)]$$

and the odd cross-correlation function was computed from

$$Q_{xy}(\tau) = 0.5[R_{xy}(\tau) - R_{xy}(-\tau)]$$

where $\tau = 0, 1 \ldots 32$.

The cospectrum, or the cosine transformation of the even cross-correlation coefficient was then computed from

$$C(N) = (16)^{-1}[C(0)/2 + \Sigma C(\tau) \cos \tau N(360^\circ/64) + C(32)/2 \cos N 180^\circ]$$

for $N = 0, 1, \ldots 32$ and $\tau = 1, 2, \ldots 31$. If x_i were voltage and y_i were current, than C(N) would be the in-phase power being dissipated within a frequency band $(64\,\Delta t)^{-1}$ c/sec wide and centered at $N\,(64\,\Delta t)^{-1}$ c/sec. This cross-covariance (or power) per bandwidth will sum to the total cross-covariance

$$R_{xy}(0) = C_{xy}(0) = \Sigma C(N)$$

where $N = 0, 1, \ldots 32$. The raw spectral estimates were smoothed with weights of 0.25, 0.50 and 0.25.

The cospectrum allows only for the simultaneous relations between the time series. To obtain information about lag relations between x_i and y_i, the quadrature spectrum must computed from the odd cross-correlation function as

$$Q(N) = (16)^{-1}[Q(0)/2 + \Sigma Q(\tau) \sin \tau N (360^\circ/64) + Q(32)/2 \sin N 180^\circ]$$

for $N = 0, 1 \ldots 32$ and $\tau = 1, 2, \ldots 31$. If x were voltage and y were current then Q(N) would be the (reactive) power from quadrature components of x and y within a band $(64\ \Delta t)^{-1}$ c/sec wide and centered at a frequency of N $(64\ \Delta t)^{-1}$ c/sec.

The utility of the quadrature spectrum is that the ratio of in-phase to quadrature spectral values is a measure of the phase between frequency components of x and y. The phase angle θ is given in

$$Q(N)/C(N) = \text{Tan}\ \theta\ (N)$$

For example, if x_i and y_i were of the same frequency and 90° apart, all of the cross-products between x_i and $y_{i+\tau}$ would result in only a quadrature spectrum.

If there is a broad spectra to x_i and y_i, it is possible to compute the phase between x and y as a function of frequency.

E. Coherency

The combination of variance and covariance spectra can be used to measure how good the relationship is between x and y at any specific frequency. This combination, expressed as the coherence is given by

$$CH(N) = [\ C^2(N) + Q^2(N)]/[G_x(N)\ G_y(N)]$$

where $G_x(N)$ and $G_y(N)$ are the spectral estimates of x and y at frequency N and $C^2(N)$ and $Q^2(N)$ are the squares of the in-phase and quadrature spectral estimates. The coherence can vary from 0 to 1 and is similar to the square of the correlation coefficient, except that the coherency can vary with frequency.

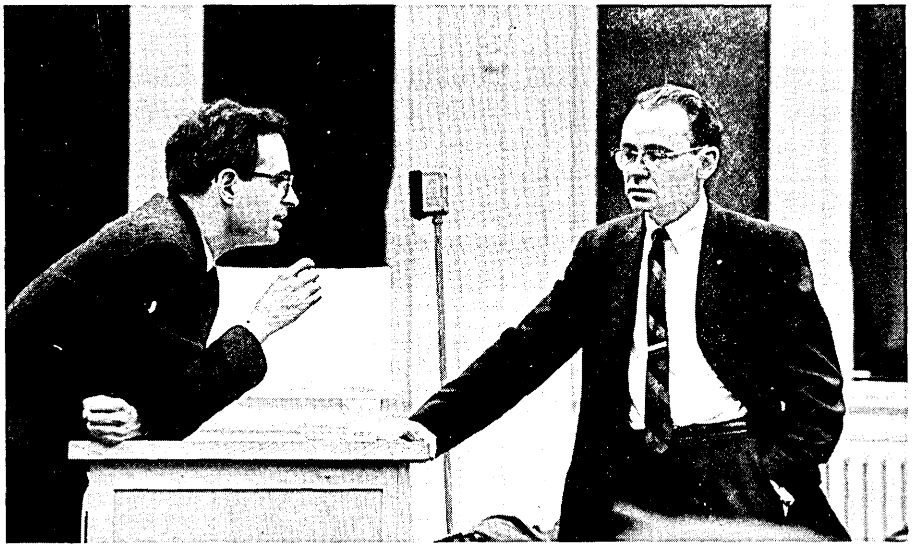
For example, if x and y were 10 c/sec signals with constant phase difference, the sum of the squares of their in-phase and quadrature spectra would be equal to the product of the variances of x and y and the coherency would be 1. If there were power at 10 c/sec in x and y, but they were not of consistent phase, the numerator would be low and the coherency low.

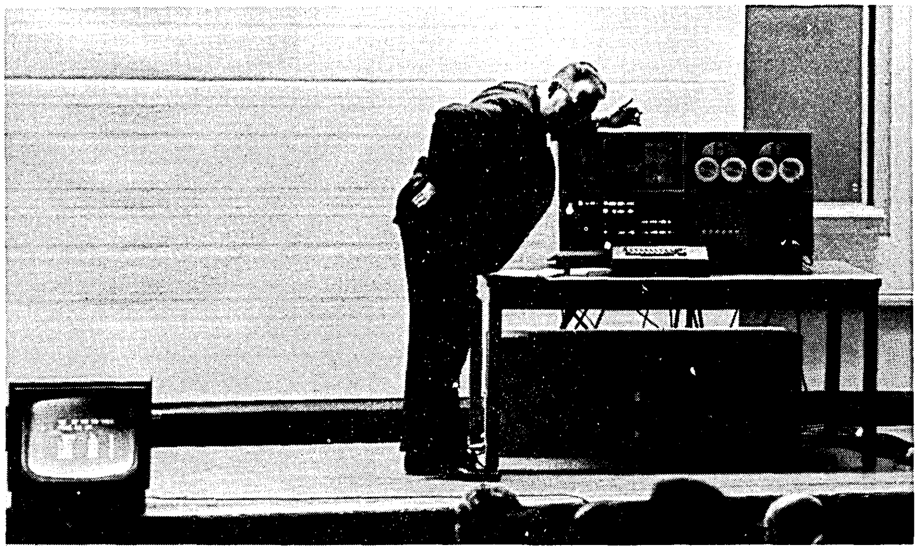
The ratio of y to x, at frequency N can be calculated from

$$[G_y(N)/G_x(N)]^{1/2}$$

This is the absolute value of the transfer function gain between x and y.

The transfer function gain and phase between x and y can only be used at frequencies where the coherency is high.





LINC EVALUATION PROGRAM

FINAL REPORT

E. O. Attinger and A. Anne

Research Institute, Presbyterian
Hospital, 51 North 39th Street
Philadelphia, Pennsylvania

NIH Grant Number FR00148-01

Table of Contents

I. Introduction and specific aims

II. Methods of Procedure

    1. Experimental technique

    2. Analysis of data

III. Facilities used

IV. Past and Present research

V. Future Research

VI. Training program

VII. Computer performance

VIII. Conclusion

Appendix:

1. Copy of log book

2. Copy of pertinent program

3. Bibliography

I. Introduction and Specific Aims

The LINC evaluation program was devised to:

A. Evaluate the feasibility of designing a general purpose digital computer for use as a routine instrument in a biological laboratory.

B. Evaluate the performance of such a computer and its impact on an actual research program over a period of 18 months.

Our task was to carry out these two objectives within the framework of a cardiovascular research facility. The basic problems we are interested in involve the relations between pressure and flow, the propagation of the pressure pulse, the stress-strain relations in the vascular wall and the relations between cardiac work and the physical properties of the vascular system. Because of the pulsatile nature of blood flow and the complex geometry of the vascular tree, a large number of variables have to be measured as a function of time. In order to subject the experimental data to a meaningful analysis these various time functions have to be expressed mathematically. If such analyses are to be carried out within reasonable time periods (thus allowing a fair number of replications of a given experiment) a digital computer and analog-to-digital conversion facilities within the laboratory are essential.

Our evaluation deals, therefore, primarily with the following aspects of a laboratory instrument computer:

1. A-D conversion (speed and accuracy)
2. Ability to process a large number of digitized data rapidly and reliably.

3. Output of the reduced data in suitable form.

4. Reliability of long term performance (breakdowns).

II. Methods of Procedure

A. Experimental technique

At the present stage of our research program we are primarily interested with the following two projects:

1. Evaluation of velocity profiles and viscous losses in the vessel wall during pulsatile flow. These experiments are carried out entirely in models built with viscoelastic tubes. The following parameters are measured:

Pressures at various sites along the tube are sensed by means of Statham strain gauges.

Volume flow is estimated by electromagnetic flowmeters.

Radial displacement of the wall is measured by magnetic coils and wall vibrations by piezo electric crystals.

Velocity profiles are obtained by estimating the radial velocity gradient at various points across the tube using a birefringence technique.

Flow is generated by an air-driven pump with variable frequency and stroke volume, the perfusion liquid is a suspension of Hector bentonite in water. (Details of the technique are described in #2 of the appended bibliography.)

In these experiments we measure simultaneously 13 parameters, record them on magnetic tape, marking the cycles to be analyzed by signals generated from the computer on the fourteenth channel of the tape recorder during the actual experiments. Each flow pulse generates a trigger, and the computer measures the cycle length during this initial pass of data. The data are then played back for analysis (Fig. 1).

2. Evaluation of pressure-flow relations and propagation of pressure pulses in the arterial tree.

These experiments are carried out on anesthetized dogs. Up to eight pressures and four volume flows are measured simultaneously at various points of the vascular bed. Again the cycles to be analyzed are marked by a signal generated by the computer. In this instance the QRS wave of the electrocardiogram is used as a trigger to determine cycle length. The data are then played back for analysis.

In order to obtain pressure gradients, we feed the pertinent pressures into an analog computer (TR 48), where they are subtracted and the output of the TR 48 is then treated as a regular analog input into the LINC.

B. Analysis of Data

1. Procedure

In order to represent the various time functions in mathematical terms, we are using Fourier series, treating the pulsatile flow as a steady state and hence periodic phenomenon:

$$f(t) = a_o + \sum_{n=1}^{\infty} a_n \cos n\omega t + \sum_{n=1}^{\infty} b_n \sin n\omega t$$

$$= a_o + \sum_{n=1}^{\infty} \cos (n\omega t - \Theta_n)$$

where $a_o = \frac{1}{T} \int_0^T f(t)\ dt$

$$a_n = \frac{2}{T} \int_0^T f(t) \cos n\omega t\ dt$$

$$b_n = \frac{2}{T} \int_0^T f(t) \sin n\omega t\ dt$$

$$c_n = (a_n^2 + b_n^2)^{1/2}$$

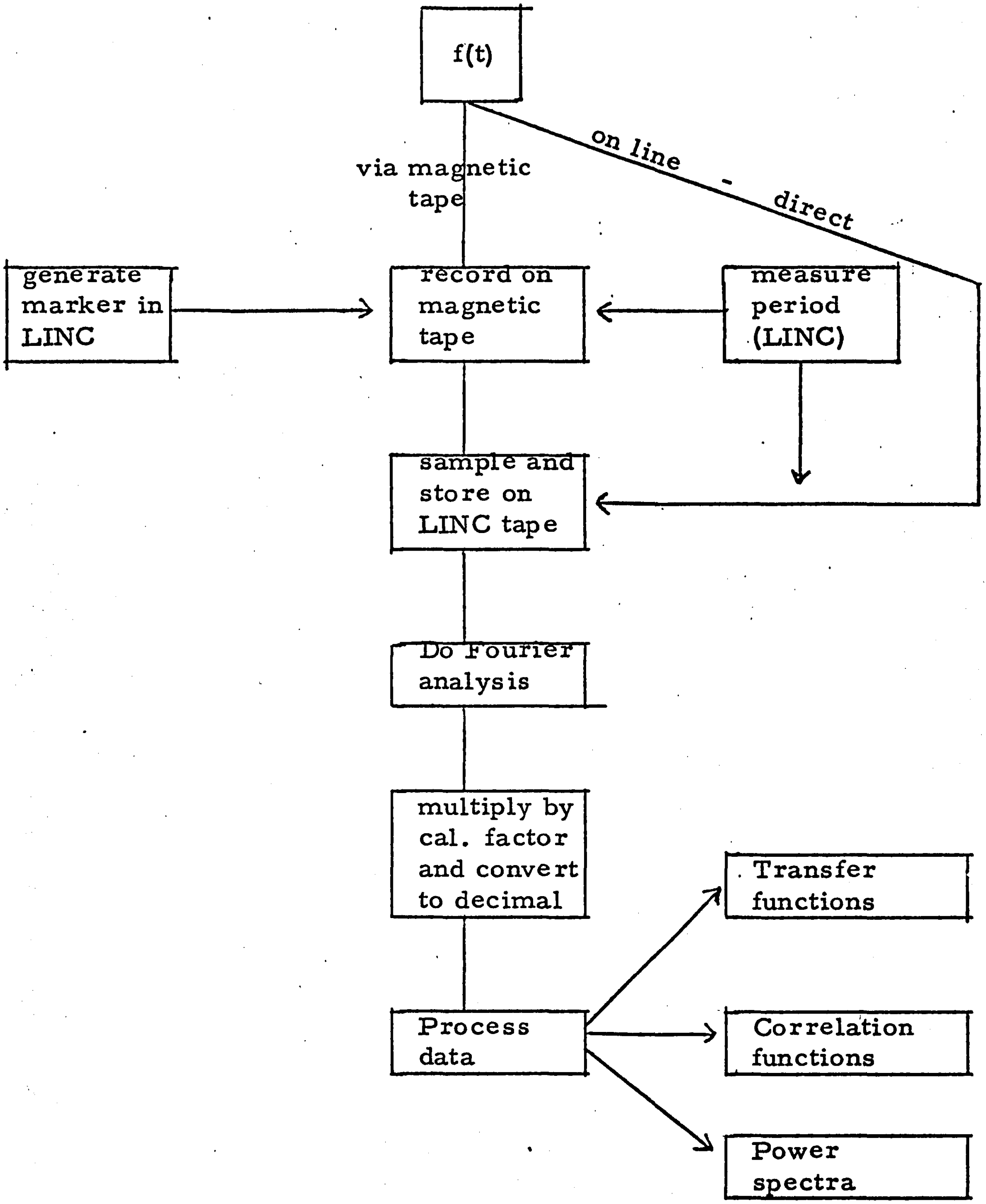
$$\Theta_n = \tan^{-1} \frac{b_n}{a_n}$$

where T = period of a cycle

$\omega = 2\pi f$ = angular velocity

All the transducers are calibrated before the experiment in terms of the transducer sensed quantity per LINC unit, which yields the appropriate calibration factor for a given transducer output applied to a given A-D converter input channel (Variation in gain between the various A-D converter channels is in the order of five per cent.).

The flow chart for the whole analytical procedure is as follows:


f(t)
via magnetic tape
on line - direct
generate marker in LINC
record on magnetic tape
measure period (LINC)
sample and store on LINC tape
Do Fourier analysis
multiply by cal. factor and convert to decimal
Process data
Transfer functions
Correlation functions
Power spectra

Flow Chart

Depending upon the circumstances of the experiment, the Fourier coefficients are then subjected to further analysis, namely the calculation of wave velocity, longitudinal and transverse (wall) impedance, correlation functions and power spectra.

For the correlation functions and power spectra we make use of the periodic properties; thereby reducing the number of calculations considerably. Under these conditions the correlation functions become:

Autocorrelation:

$$\phi_{xx}(\tau) = \int_{-\infty}^{\infty} x(t) \cdot x(t - \tau) \ dt$$

$$= a_o^2 + \frac{1}{2} \sum_{n=1}^{\infty} (a_n^2 + b_n^2) \cos n\omega\tau$$

Cross correlation:

$$\phi_{xy}(\tau) = \int_{-\infty}^{\infty} x(t) \cdot y(t - \tau) \ dt$$

$$= a_o A_o + \frac{1}{2} \sum_{n=1}^{\infty} (a_n A_n + b_n B_n) \cos n\omega\tau$$

$$+ \frac{1}{2} \sum_{n=1}^{\infty} (a_n B_n - b_n A_n) \sin n\omega\tau$$

and the corresponding power spectra:

$$G_{xx}(\omega) = \int_{-\infty}^{\infty} \phi_{xx}(\tau) \ e^{-j\omega\tau} \ d\tau$$

$$G_{xy}(\omega) = \int_{-\infty}^{\infty} \phi_{xy}(\tau) \ e^{-j\omega\tau} \ d\tau$$

The final data are either printed out or displayed in graphical form on the oscilloscope and photographed (Fig. 2).

The amount of samples required per cycle depends, of course, on the frequency spectrum to be investigated (0 to 30 cps for pressure, flow, displacement; 0 to 100 cps for wall vibrations; 0 to 500 cps for birefringence). Programs are available which use sine and cosine tables or calculate the trigonometric functions, depending on the circumstances. All programs are written in double precision arithmetic as well as in floating binary (see Appendix). On line analysis for 100 samples and 12 harmonics takes about two minutes per cycle, if no playback is required, i.e. we can perform a complete run with seven variables every 15 minutes, having the results of the previous run already available. In a typical animal experiment we use between eight and 12 variables and perform between 40 and 60 individual runs. Time for complete analysis is about two days. In birefringence experiments, where we use 13 variables and 20 to 40 runs, analysis time is between five and eight days, including all auto- and cross-correlations as well as power spectra.

Accuracy of analysis:

Maximum static and dynamic accuracy of the transducers is ± 1% over a frequency range of 0 to 30 cps. The range of the A-D converter is ± 1 volt with an accuracy of 1/256 if the whole range can be used. The Fourier components of our time functions are characterized by a ratio of the first six harmonics in the order of 1:3/4:1/3:1/4:1/8:1/10. Hence it is obvious that the accuracy of the conversion will decrease as the number of harmonics increases even if we amplify the complex wave form, representing the input to the A-D converter to a magnitude of ± 1 volt. Furthermore, because of unavoidable drifts, both in the biological and electrical systems, we can only use about 2/3 of the actual converter range. Thence the expected minimal errors will be in the order of

1.5%	for the fundamental
2.0%	for the second harmonic
3.0%	for the third harmonic
4.0%	for the fourth harmonic
6.0%	for the fifth harmonic
7.0%	for the sixth harmonic

and progressively larger for higher harmonics. In terms of the calculated angles the error will be largest if $\tan \Theta \sim 1$ or $\tan^{-1} \Theta \sim 45^{\circ}$. Here the maximum error is in the order of 36°! This is one of the main reasons why we feel that the word length of the A-D converter should be increased. (see VII.) Particular care has to be taken to avoid clipping. We use zero suppression circuits to obtain maximum amplitudes independent of mean values.

Errors due to sampling are negligible at the frequencies considered (0 to 30 cps). Maximum error in the determination of the sampling interval is 4 µsec (1/2 the cycle length of the LINC). For 100 samples, this amounts to an error of 400 µsec between measured and sampled period. However, if we are looking for higher frequency components (100 to 500 cps) this error may amount to 5 msec. Fig. 3 illustrates the magnitude of errors involved if the actual period and the sampling period are not equal. On the abscissa the ratio between these two periods is plotted and the ordinate shows the amount of harmonic content generated at this ratio for a pure sine wave input. For example, for a five per cent error in period measurement, the computer will calculate a spurious harmonic content of 10% second, 5% third etc. (percentages in terms of the fundamental). As the error in period measurement increases this distortion becomes intolerable. Hence, for this kind of experiment, accurate determination of sampling interval and period is of primary importance. Aliasing errors are less significant. Furthermore, in order to decrease the conversion error, a larger voltage range of the converter as well as a larger work size as compared to the present LINC would be preferable.

Otherwise, we have been well pleased with the performance of the converter, although we plan to convert some of the "knob inputs" into regular inputs for analog data, in order to increase the number of channels which can be converted simultaneously. The limited memory (2000 words) has been no serious problem, although it necessitated judicious programming and considerable tape-shuffling with associated time losses. Output format, reliability and performance have been superb (see Appendix: Log). As a matter of fact, troubles during experiments have never been due to the LINC but always to the related equipment, particularly the tape recorder (Ampex CP 1000). The performance of the latter has, at times, been so annoying that I hesitate to recommend it.

III. Facilities Used

The LINC was set up in the Research Institute at Presbyterian Hospital in Philadelphia in September, 1963. No problems were encountered in its operation and the computer was demonstrated to the scientific community of Philadelphia in November 1963, using our first on line Fourier analysis program on various time functions.

As mentioned above we have used exclusively analog inputs, originating from : strain gauges, electromagnetic flowmeters, magnetic coils, phototubes, piezo crystals, and displacement transducers. In a number of instances the analog data were first processed in an analog computer and then fed into the LINC. A most disturbing result in our evaluation of the analog and digital computers was the following: Two pressure measurements were fed simultaneously into a Statham differential pressure transducer. P_1 and P_2 were fed simultaneously into the analog computer and into the LINC and the differential pressures obtained compared with those measured directly from the gauge. The differences in the three methods of obtaining a pressure gradient were disturbingly large and at present we do not have a satisfactory explanation for it. The sampling procedures employed included both on line methods and play back from magnetic tape and used various triggers generated by the experimental preparation for initiation of the sampling.

IV. Past and Present Research

The general aspects of this problem have been discussed in Section II. The programs used are discussed in more detail in the Appendix.

Results:

A. Studies on velocity profiles:

The birefringence experiments indicate that disturbances are superimposed upon the theoretically expected Bessel-type velocity profile. These disturbances develop first in the central fluid core, spreading toward the periphery as velocity and acceleration are increased. Although absolute values on power losses due to these disturbances are difficult to estimate they represent probably less than 5% of the total viscous losses in the fluid (See Fig. 2). The frequency spectrum of these disturbances during physiological flow rates extends from 0 to 150 cps, that of the wall vibrations from 0 to 60 cps.

B. Comparison of calculated and measured flow rate:

Accurate measurement of blood flow is of primary importance for the evaluation of the circulatory system in health and disease. Since at present it is not possible to obtain such measurements without surgical intervention, it would be highly desirable if the actual flow rate could be predicted from pressure measurements, which are easier to perform. We have compared measured flow and flow calculated from the pressure gradient, using Womersley's solution in various experimental models. Our results indicate that the error in the prediction is in the order of 10 to 20% and random, and probably related primarily to the difficulty of measuring a small pressure gradient over a short distance accurately. These experiments are now being extended to the living system.

C. Propagation of the pressure pulse in the arterial tree:

We have mapped out the propagation of the pressure pulse from the facial artery to the foot of the dog in intervals of 2.5 cm. From these data we obtained estimates with respect to damping, site of reflection and fluid impedance. These experiments are now being extended to include flow measurements at various sites in the arterial tree.

D. Role of the LINC in these experiments:

The performance of these experiments is contingent on the availability of a digital computer within the laboratory. The validity of the experimental data depends to a large extent upon the reliability and accuracy of the various transformers and we have used the LINC extensively for the dynamic calibration of our equipment. Without the LINC we would have been able to perform less than 10% of the above experiments even if the staff had been increased. We feel that the cost of a LINC is recovered within one year of operation if the computer is properly used.

V. Future Research

None of the projects discussed under IV have been completed yet. This work will continue and incorporate some of the second order effects, such as tapering, anisotropy, nonlinearities, and ellipsoid cross sections.

In addition we plan to embark this summer on a program project dealing with engineering analysis of the circulatory and respiratory system and their interaction. After all, the function of these three systems is so closely interrelated that a knowledge of the overall function is necessary for the assessment of an individual system and its integration into the living organism. The physical aspects of the circulatory and respiratory systems are quite similar,and we want to extend our investigations into the respiratory and nervous systems, using essentially the same methods (Fourier analysis and correlation techniques). However, since the nonlinearities are more prominent in the respiratory system, the analysis will be more difficult and will involve among others phase plane methods.

We have already begun to investigate the frequency modulation of the arterial pulse produced by the respiratory pressure changes. For this work the availability of the LINC will be essential.

Neuro-otology

Dr. R. Cutt is studying auditory nerve discharge following destruction of the olivocochlear bundle. He plans to use the LINC for averaging of successive repetition of click stimuli, so that the neural response will stand out against the noise level, and for cancellation of microphonic potentials by averaging equal numbers of responses to condensation and rarifaction clicks.

VI. Training Program

We feel that the initial training period at M.I.T. has been invaluable. However, I believe that a month of intensive training is a minimum if a new computer (even if it is relatively simple to operate) is to be used profitably in a biological laboratory. I am not at all convinced that a biologist without either mathematical and computer experience or the assistance of a good engineer will be able to make use of the LINC and its particular features.

We have been particularly fortunate in having Dr. Anne with us, who has, I believe, made maximal use of the special features of the LINC for our purposes. He has given a number of courses on the computer to our staff, so that most of them are able to use the LINC, although they would be somewhat at a loss if they had to write their own programs for their rather complicated analyses. However, the extensive use we have made of the LINC in our Institute has impressed upon each member of the staff the importance of clear and careful formulation of a problem. I firmly believe that any interdisciplinary research project can only be attacked successfully if its director is solidly grounded in <u>all</u> the disciplines involved.

VII. Computer Performance

A. Maintenance:

We are quite satisfied with the mechanical performance of the LINC. After we received the LINC, we had very few breakdowns and little loss of time. The breakdowns were rather minor and we had little trouble correcting them. We noticed recently that some of the keys in the keyboard are giving trouble once in a while. It looks like we may have to replace it in the near future.

B. Technical Performance:

We are, in general, satisfied with the adequacy of the LINC as an all-purpose digital computer. We are happy to say that we could incorporate the LINC into our experimental set-up without any procedural changes. We used the LINC to convert analog signals into digital information and then to perform the required mathematical operations. We have taken advantage of all the special features like relays and external trigger lines. The total performance of the LINC can be described as follows:

<u>Word length</u>: We found that the word length is inadequate to perform most of our calculations using single word arithmetic. Double precision

arithmetic was used in most of our calculations. Even in the floating point binary arithmetic the addition or subtraction of two numbers is restricted to four decimal places. To increase the accuracy of the calculations, it is suggested that the length of the word should be increased. A word length of 18 bits (including sign bit) will almost double the magnitude of the numbers that can be represented by one word. A larger word will reduce the use of double precision arithmetic, thus reducing time needed for calculations.

Memory size: One of the primary functions of the LINC, we believe, is to convert analog signals into digital information. In this respect, we sometimes have to restrict the number of samples below the desired limit, although it is always possible with the present memory to obtain any number of samples by storing the analog information on an analog tape. However, by doubling the present memory size not only will the amount of time required for conversion from analog tape be reduced, but on line analysis of the data will be facilitated. (In our case, particularly for the analysis of steady flow and of various respiratory and circulatory control mechanisms)

A-D Converter: The rate of A-D conversion is very satisfactory. The accuracy of A-D conversion is not sufficient. We believe that the length of the converted number should be increased from seven bits to ten or 11 bits. It is also suggested that the range of the input voltages to the analog channels be increased from the present range of -1 to +1 volts. This will be helpful for people who are using DC amplifiers to amplify the signals from the transducers, thus reducing the problems due to drifting in the amplifiers. (As mentioned before, as a matter of precaution we use only 2/3 of the conversion range in order to avoid clipping). It seems to us that eight knobs on the display oscilloscope are not necessary and at least four of them can be converted to high speed analog channels.

D-A Converters: Two D-A converters available on the LINC are, unfortunately, not useful except for oscilloscope display. The reason is that they are permanently connected to the accumulator and B-register. Availability of digital-to-analog converters, without tying up any of the registers that are needed to do arithmetic operations, is essential if the LINC and an analog computer are used together in hybrid computation. Even though presently many people may not use hybrid computations, the above feature will increase the usefulness of the LINC in the future.

Relays: The relays are useful sometimes and should be retained. We have used them to generate marker pulses to be recorded on the analog tape.

External Trigger Lines: This is an important feature and should be retained. However, we feel that fourteen lines as a standard may not be necessary. Money saved by reducing them to half can be used for other improvements.

Instructions: The structure of the instructions is simple and easy to understand. We really did not find any use for Half-Word instructions. The functions of these instructions can be achieved by already available means. An important addition to the overall usefulness of the LINC would be the introduction of Double-Precision instructions: LOAD, STORE, ADD, COMPLEMENT and MULTIPLY. This feature would reduce the calculating time for double precision arithmetic tremendously.

Input/Output Units: The small keyboard should be kept. Improvement in the quality of the keyboard is necessary. Oscilloscope display is a useful feature and should be retained. However, it has to be complemented by a printer for numerical data, otherwise the operation becomes too expensive. The magnetic tape units are satisfactory. It would be desirable that the time needed to read and write on magnetic tape be reduced. A printing unit should be made standard on the LINC.

In summary, we believe the following changes will increase the usefulness of the LINC.

1. Word length: 18 bits including sign bit
2. Memory size: 4096_{10} words
3. A-D converters: a) input voltage range -2 to +2 volts
   b) converted word = 10 or 11 bits
4. D-A converters: a) at least two; provision to increase if needed
   b) output voltage range: -10 to +10 volts
5. Double-Precision instructions: LOAD, STORE, ADD, COMPLEMENT, and MULTIPLY
   (This can be done at the expense of Half-Word instructions, if necessary.)

We believe with the above modifications and other existing features, LINC will be an ideal computer for any laboratory. In the present LINC we would like to see the changes 3 and 4.

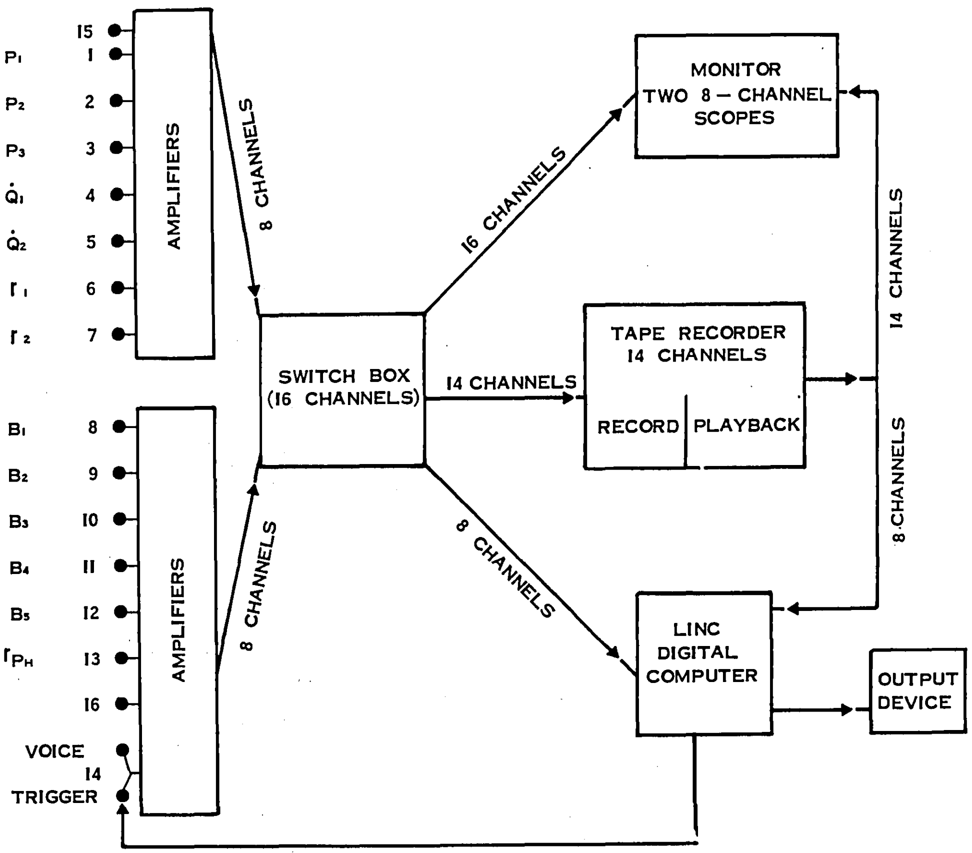
VIII. Conclusion

We have been most satisfied with the performance of the LINC in our laboratory. The computer has practically been in daily use and has permitted us to execute a large amount of experimental work which would not have been possible otherwise. We definitely wish to retain the machine.

For use of the LINC in our work we would like to suggest the following modifications in its design:

1. Increase the voltage of the A-D converter to ±2volts and extend the word length for the converter to 10 or 11 bits.
2. Increase the word length of the LINC to 18 bits.
3. Increase the memory to 4096_{10} words.
4. Incorporate double precision instructions into the logic.

Although we have the impression that the LINC was designed primarily with the needs of neurophysiology and psychology in mind, we respectfully suggest that it is at least as applicable in the field of circulatory and respiratory physiology. We have now been using computers in our experimental work for more than five years, but from the point of the experimenter we have not yet had a computer facility comparable to the LINC. No other digital computer has offered even a comparable ease of communication between the experimental preparation and the investigator.



SYSTEM FOR ON – LINE ANALYSIS OF PULSATILE FLOW

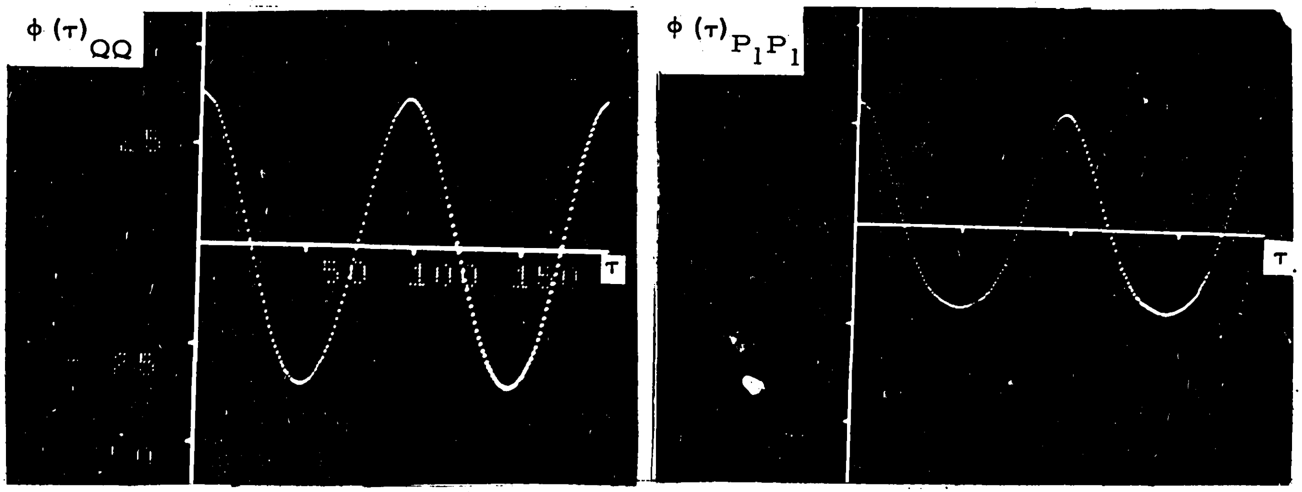
The different transducer outputs (1-13) are fed through amplifiers into a switch box, and are continuously monitored by 2 8-channel oscilloscopes.

For direct on line analysis (maximum 8 channels, 96 samples per cycle), the data are fed directly into the A-D converter of the LINC where the period between two triggers is determined and the sampling interval calculated. Sampling is then initiated by the next trigger (This assumes steady state with constant period.)

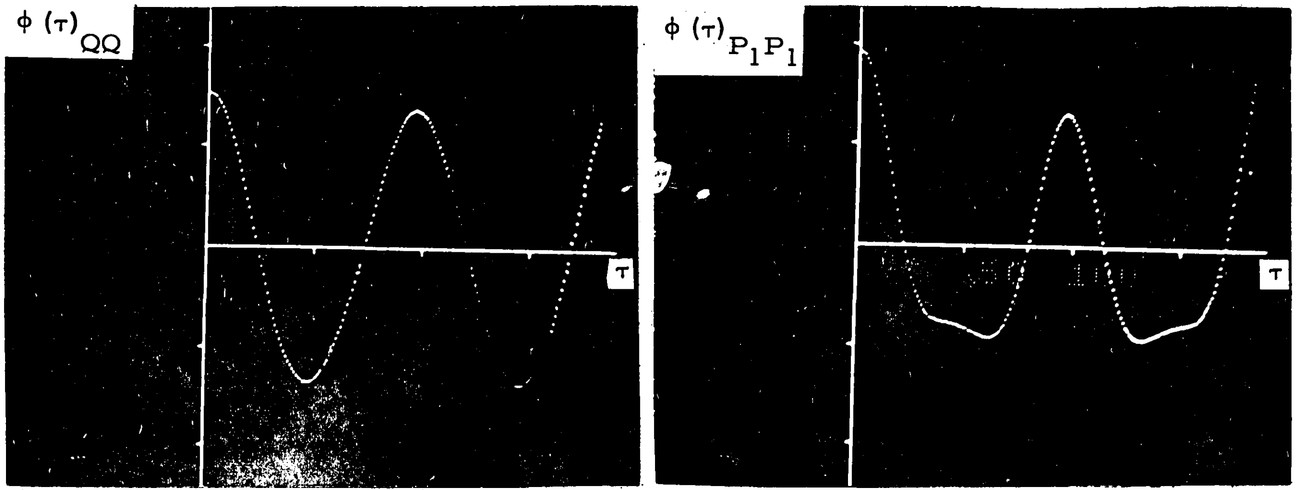
For "indirect" on line analysis, the data are recorded on magnetic tape, the cycles to be analyzed are marked under visual observation by a signal generated in the LINC. At the same time the period for the marked cycle is determined in the LINC. The data are then played back and sampled by the converter for the marked cycles.

Figure 1

AUTOCORRELATION OF FLOW AND PRESSURE



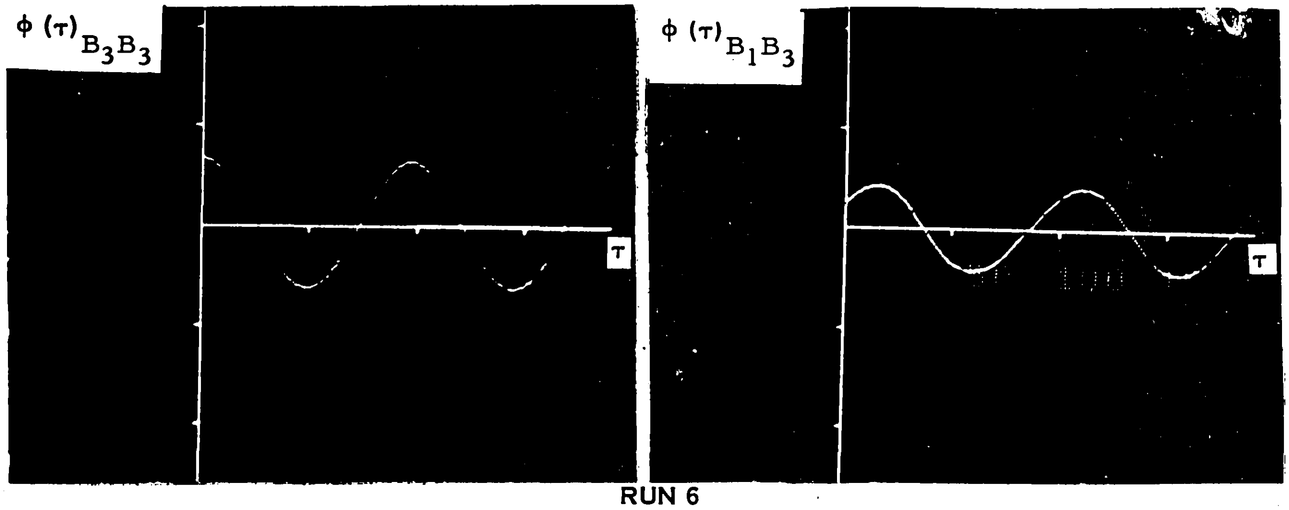
RUN 6
DRIVING PRESSURE 300 cmH$_2$O f = 1.2, RE = 2900



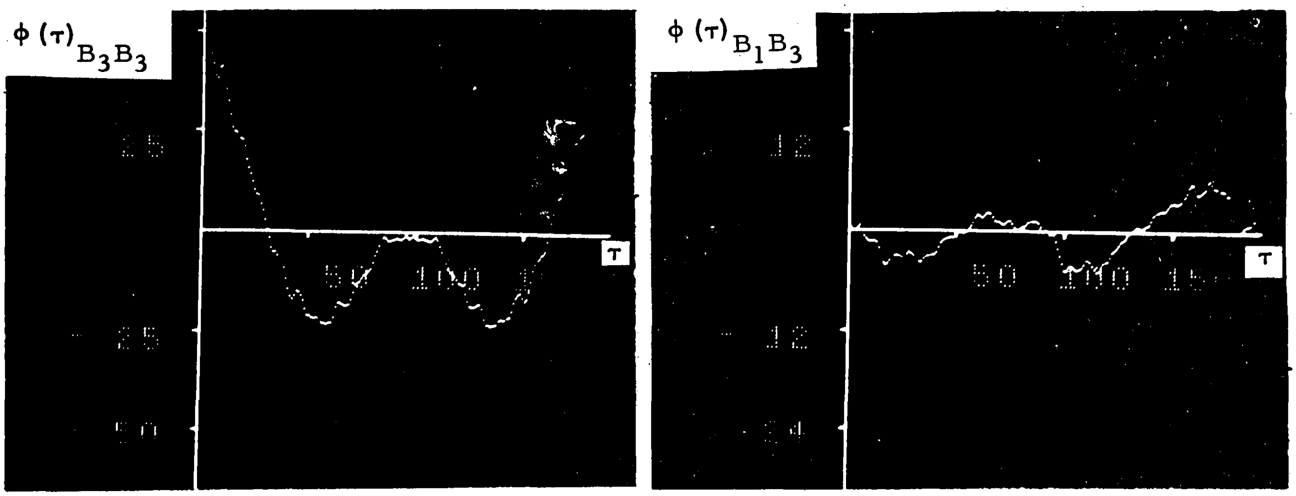
RUN 7
DRIVING PRESSURE 500 cmH$_2$O f = 1.2, RE = 5500

Figure 2 A

AUTOCORRELATION (LEFT) AND CROSS CORRELATION (RIGHT) FOR BIREFRINGENCE

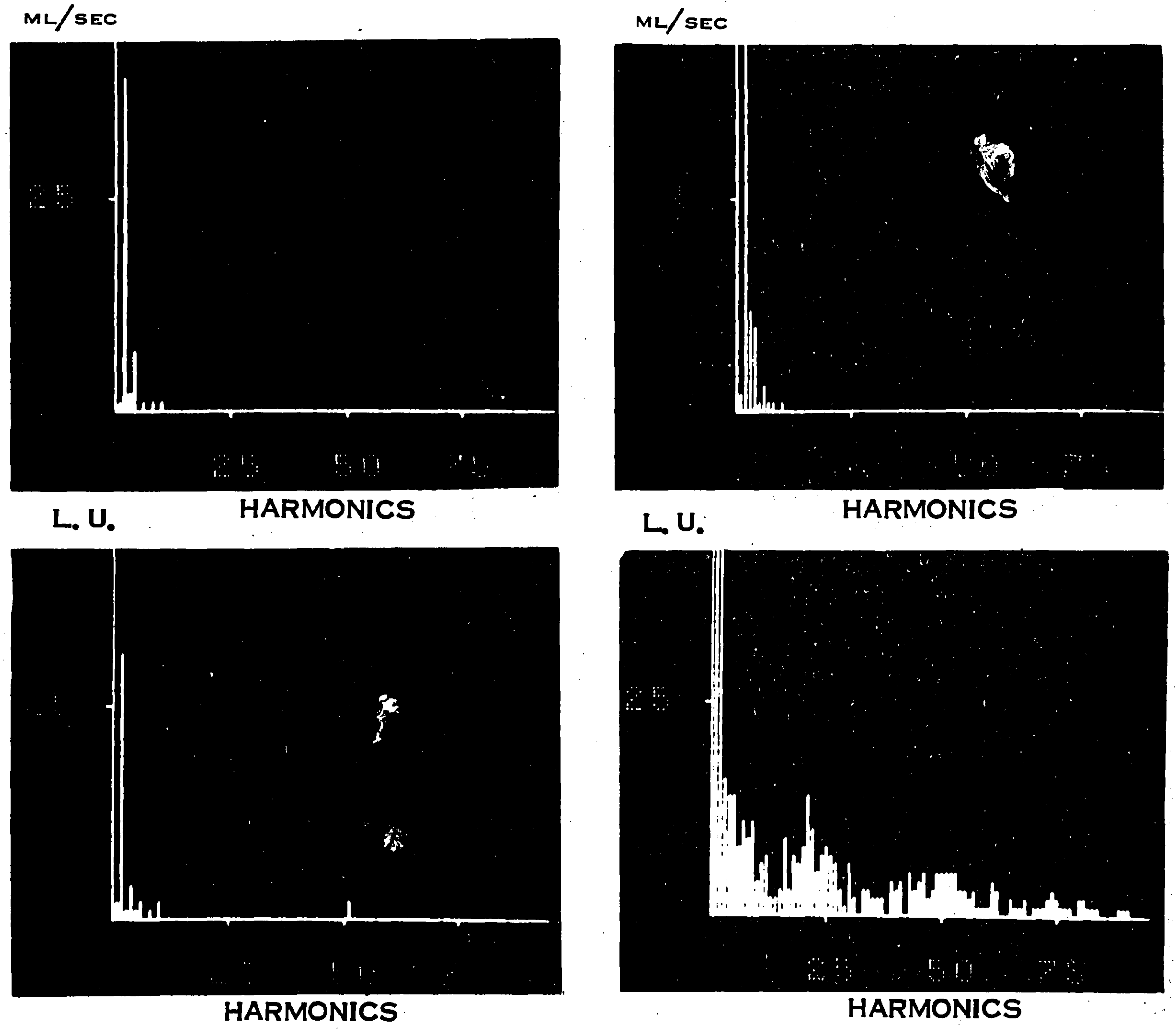


RUN 6
DRIVING PRESSURE 300 cmH_2O f = 1.2, RE = 2900



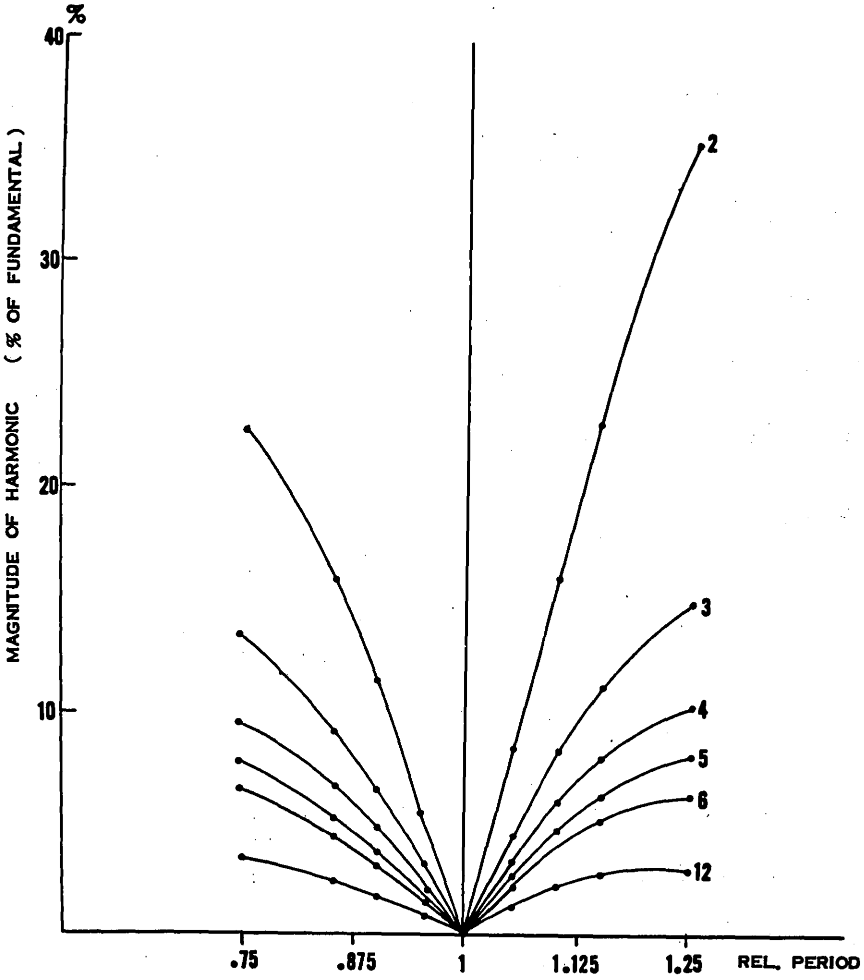
RUN 7
DRIVING PRESSURE 500 cmH_2O f = 1.2, RE = 5500

Figure 2 B



FREQUENCY SPECTRA OF VOLUME FLOW AND BIREFRINGENCE PATTERNS IN PULSATILE FLOW.
[f = 1.2 CPS, DRIVING PRESSURE = 300 CM H_2O (a , b) AND 500 CM H_2O (c , d), MEAN FLOW = 28 ML/SEC (a , b) AND 52 ML/SEC (c , d)]

Figure 2 C



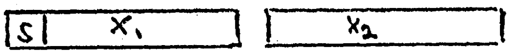
Error introduced by inaccurate determination of sampling period. A pure sine wave was analysed for various ratios of actual over sampling period. Note the increasing amount of harmonic distortion introduced if the period is not accurately determined.

Figure 3

Appendix 2

1. Description of Programs

Fourier analysis can not be handled satisfactorily using only 11 bits in each arithmetic operation. The data is never that accurate but the nature of the operations carried out requires that more bits be kept in the intermediate steps even though they will be discarded at the end. Therefore, for this reason we use double precision arithmetic. Two LINC words are used to represent an integral number. An integral number, x, is represented as x_1 x_2 where x_1



contains the most significant 11 bits in one storage location and x_2 the less significant 12 bits in the next higher numbered location. The sign of the number is denoted by the sign bit (s) in x_1 whereas the sign bit in x_2 is used as one of the bits in the number.

Subroutines were written to perform arithmetic operations in double precision. They are: 1) D.P. add (2) D.P. complement (3) D.P. multiply and 4) D.P. Divide. Subroutines were also written to calculate certain functions of a number. they are : 1) D.P Square Root, 2) D.P. Sine or cosine of an angle in radians and (3) D.P. Arc tangent.

Subroutines were written to do an arithmetic operation on two numbers stored in floating point representation. A floating binary point representation consists of writing all numbers as some fraction between 0.1_2 and 1.0_2, times a power of two. The object of the floating point method is to store the power of two, called exponent , in the memory along with the fractional part called mantissa. The mantissa and the exponent are stored in consecutive core storage locations. In the mantissa the binary point is assumed to lie between bits 10 and 11 of a LINC word. The following subroutines were written for the floating binary point method: (1) Conversion of an integral number into a floating point number (normalization) (2) Fl. add (3) Fl. multiply (4) Fl. divide (5) Conversion of floating binary number into floating decimal number and print on teletype (6) Fl. square root (7) Fl. sine or cosine of angle in radians.

2. Main Programs :

A) Calculation of Fourier Series: This program uses double precision arithmetic. The number of samples can be 24, 48, 96, 192, 384 or 768. The samples have to be stored on a block starting with location zero. The number of harmonics to be calculated can be varied from 1 to 384. A sine table in intervals of 2 /768 from 0^{o} to 90^{o} is contained in the programs. Sines and cosines are multiplied by 1000_8 to maintain the required accuracy. Number of samples, number of harmonics, sample block number and coefficient block number are typed in through the keyboard. When the calculation is finished the coefficients are stored on the tape and the machine is ready for the next calculation.

When Fourier coefficients are calculated for more than one variable, the calculations are performed in groups of variables. In this case the following information is typed:

1) Number of samples

2) Number of harmonics

3) Number of variables in a group

4) Initial sample block number of the first group.

5) Initial coefficient block number of the first group

6) Number of blocks to be advanced on the tape to find the next group.

7) Initial sample block number of the last group.

Using the above information, the machine can run hours without the presence of an operator. This program can be used to calculate Fourier coefficients either for each variable or in groups of variables, depending upon the position of sense switches 2. In either case, sense switch 0 controls the display of Fourier coefficients.

b) Conversion of Fourier coefficients and calculation of phase angles

The program converts the Fourier coefficients which are in LINC units, to actual units of a variable when the appropriate calibration factor is typed in. The phase angles of the different harmonics are calculated using an arc tangent routine. The magnitude and phase angles of the Fourier coefficients are displayed in the decimal system.

c) Sampling Programs

Two programs are available for on line analysis of the data. One uses the zero crossing of the positive slope of a reference signal, for example a sine wave, to determine the period of one cycle. The second program uses the positive or negative peak of a square wave as a reference. This can be used in animal experiments using the QRS complex of the ECG. Both programs take the required number of samples and store them on the tape. At the end of sampling the period is displayed on the scope.

When the data is recorded on an analog tape, the cycles of interest are marked by a marker pulse generated by the computer. A sampling program was written to handle this case.

d) Calculation of Power Spectra:

Programs were written to calculate the auto- and cross-correlation functions using the samples of any periodic signal. Programs were also written to calculate power spectra from either auto-correlation function or Fourier coefficients.

e) Print Programs:

Programs for printing Fourier coefficients and power spectra in the decimal system were written for the teletype printer.

The actual programs follow.

Bibliography

1. Attinger, E.O., A. Anne, H. Sugawara, A. Riccetto and D.A. McDonald: Computer-coupled model for pulsatile flow studies. Fed. Proc. 23: 110, 1964.

2. Attinger, E.O., A. Anne, and H. Sugawara: On-line analysis of hemodynamic parameters, in Modern Instrumentation. W. Murray and P.F. Salisbury, ed. New York, Plenum Press, 1964, p. 195-210.

3. Attinger, E.O.: Interpretation of circulatory parameters in pulsatile blood flow, IV European Congress of Cardiology, Prague, August, 1964. (To be published in Proceedings).

4. McDonald, D.A. and E.O. Attinger: The shape of the traveling pressure pulse, presented at Fall Meetings of American Physiological Society, Providence, R.I., September, 1964.

5. Anne, A., E.O. Attinger, H. Sugawara and A. Navarro: Frequency spectra of arterial flow parameters, presented at 17th Annual Conference in Engineering and Biology, Cleveland, November, 1964.

6. McDonald, D.A. and E.O. Attinger: The characteristics of arterial pressure wave propagation in the dog. Submitted to Circulation Research.

7. McDonald, D.A., A. Navarro, R. Martin and E.O. Attinger: Regional pulse-wave velocities in the arterial tree. Federation American Societies of Experimental Biology, Atlantic City, April, 1965.

In Preparation:

8. Attinger, E.O.: The physics of pulsatile blood flow with particular reference to small vessels. To be presented at Symposium on Vascular Disorders of the Eye, San Francisco, June 4-5, 1965.

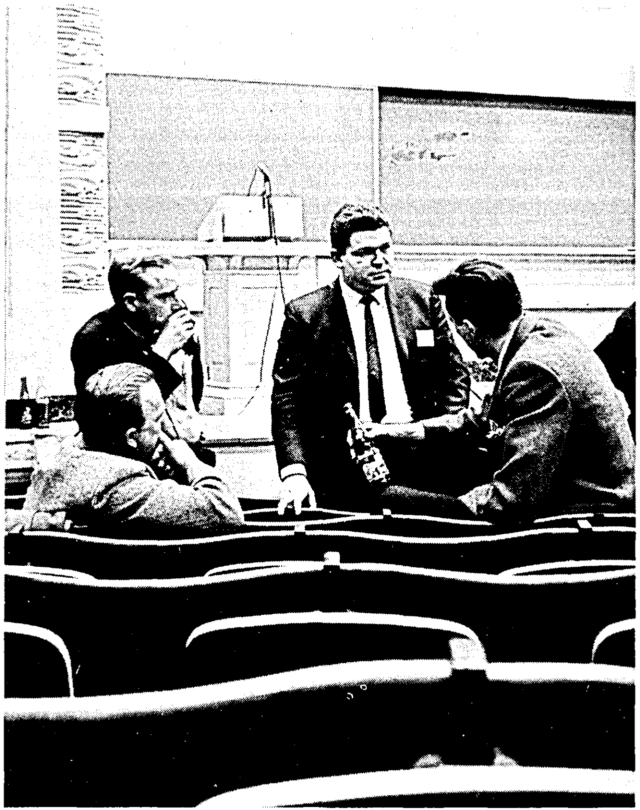
9. Attinger, E.O., A. Anne and D.A. McDonald: Simulation of the Cardiovascular System - to be presented at the Conference on Advances in Biomedical Computer Applications, New York, June 1965. To be published in Ann. N.Y. Acad. Sci.

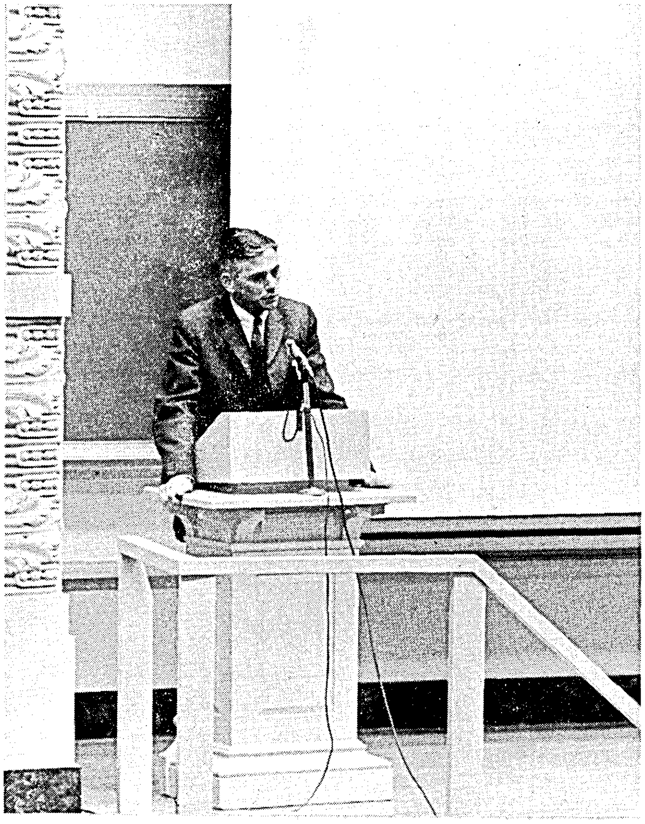
10. Attinger, E.O., D.A. McDonald and A. Anne: Some approaches to the analysis of the circulatory system. To be presented at the Gordon Research Conference, Andover, N.H., July 19-23, 1965.

(Bibliography cont.)

11. Attinger, E.O., A. Anne, H. Sugawara and D.A. McDonald: Some problems in the analysis of pulsatile blood flow. To be presented at the 16th International Conference on Medical Electronics and Biological Engineering, Tokyo, August 22-27, 1965.

12. Attinger, E.O., A. Anne and H. Sugawara: Theoretical and experimental problems in the analysis of pulsatile blood flow. To be presented at the International Congress of Physiological Sciences, Tokyo, September 1-9, 1965.






5

FINAL REPORT

LINC EVALUATION PROGRAM

I. University of Washington
Department of Physiology and Biophysics
Seattle, Washington

II. J. Walter Woodbury
Professor of Physiology

Albert M. Gordon
Instructor in Physiology

III. Laboratory facilities and personnel.

A. Facilities. The laboratory is equipped for doing research on the electrical properties of excitable cells--notably cardiac muscle. The most applicable experimental technique is the use of intracellular or extracellular microelectrodes for measuring transmembrane or extracellular potentials and for applying currents. The requisite instrumentation, e. g. micromanipulators, stereo microscope, dual beam CRO's, wave form generators, calibrators and preamplifiers with electrometer inputs, and stimulators are all available and have been set up to enable data collection and experimental control by the LINC in the laboratory.

B. Personnel. The laboratory is used by the principal investigator and the co-investigator, by Dr. D. R. Firth, a postdoctoral fellow in Biophysics, and by Mr. W. L. Hardy, a graduate student in Biophysics. In addition the LINC is used by remote control from another laboratory by Mr. W. H. Calvin, a graduate student in Physiology.

IV. Present research in which LINC is an integral part

The present research of the principal and co-investigator will be described in detail. The other projects using LINC in which others are principally involved will be listed below and described in detail in a later section.

A. The determination of the electrical equivalent circuit of cardiac muscle during diastole; to investigate cell to cell electrical interconnections. Data acquisition and preliminary analysis. J. W. Woodbury and A. M. Gordon.

B. Measurement of the fluctuations of the interspike interval of the crayfish stretch receptor as a means of investigating the excitability mechanism. D. R. Firth.

C. Analysis of correlations between interspike intervals and synaptic noise in spinal cord synapses. W. H. Calvin.

D. Theoretical calculations on receptive field characteristics. W. H. Calvin.

E. Dependance of conduction velocity in frog myelinated nerve fibers on external sodium concentration. W. L. Hardy.

A. Equivalent circuit of cardiac muscle. A long thin cell such as a nerve or skeletal muscle fiber is equivalent, electrically, to a lossy cable. An intracellularly or extracellularly applied current produces changes in transmembrane voltage (V) which vary both in time and distance. Records of V (x,t) and extracellular potential can be analyzed to give the basic parameters of the cell. However previous research in this laboratory has shown that cardiac muscle cells are electrically interconnected by means of closely approximated patches of low resistance membrane, probably the tight junctions seen in many electron micrographs. Hence the equivalent circuit of a block to cardiac tissue is much more complex than that of skeletal muscle or nerve. A first approximation is that the tissue is a one, two, or three dimensional equivalent (depending on the relative dimensions of the tissue) of a lossy cable. Preliminary analysis based on steady state distribution only has shown that this approximation is a reasonable one fitting the data with some accuracy. A more complete analysis requires the knowledge of both the time and spatial variation of voltage caused by a step of applied current. The LINC is an integral part of this present research. Voltage is measured as a function of time and distance following the application of a pulse of current either inside a cell through an intracellular electrode or outside a cell through a suction electrode. The later method for current application has been used in the present study because of the difficulty in keeping two microelectrodes in cardiac muscle cells near one another and of forcing enough current through high resistance microelectrodes. We are still in the data collection and preliminary analysis phase of this problem. The detailed analysis will be described below in the proposed research. The data collection involves the following:

1. A subthreshold step of current is applied to a bull frog atrial trabecula through a suction electrode. The intracellular or extracellular potential is recorded with a microelectrode at a distance along the trabecula. Records are taken both with the microelectrode outside and inside the cell. The LINC is triggered to sample both the current and voltage at variable time intervals during the applied current pulse. Both records are stored in memory and the pulse repeated. A predetermined number of records are included in the average to get around the problem of noise in some of the low level voltage signals. The usual sampling is 32 microseconds interval (250 points per record) and 96 microseconds for current (63 points per record). The records can then be displayed with the record information and stored on magnetic tape or rejected.

Ancillary information stored along with the records are the resting potential recorded if the microelectrode is inside a cell (both before and after all records included in the average), the injury potential measured by the suction electrode used for passing the current pulse, the time scale for the particular sample rate, the voltage calibration signal which is measured with each record, and the distance from the suction electrode to the microelectrode (can be set on a potentiometer).

The control features of the LINC are utilized in these experiments. The LINC determines when the microelectrode is inside a cell as indicated by a resting potential large enough to indicate little or no damage. The sensing of a sufficiently large resting potential initiates all the sampling procedures. This feature obtains records which would otherwise be lost from cells in which the microelectrode stays for a very brief time (seconds). In addition the relays of the LINC are used to obtain calibration levels or pulses at appropriate times. Since the voltage change resulting from the applied current pulse is much less than the resting potential, the LINC also is used to increase the gain of the voltage measuring system before the current pulse is applied.

2. After the voltage and current records have been stored on magnetic tape, they can be read in at a later date for editing.

3. To obtain the change in the transmembrane potential in response to the applied current pulse, the difference between voltage records taken with the microelectrode inside and just outside the cell membrane must be obtained. The appropriate records can be read back into the LINC memory, the difference obtained, displayed on the LINC oscilloscope, and stored with the proper values extracted from the difference record. This procedure largely cancels capacitative and resistive artifacts due to the current pulse and associated circuitry.

V. Future research in which LINC will be an integral part

A. The determination of the electrical equivalent circuit of cardiac muscle during diastole, steady state and transient analysis.

B. The use of this equivalent circuit to calculate the effects of transmembrane current density on membrane potential from applied current-voltage curves obtained throughout the prolonged cardiac action potential.

A. Equivalent circuit of cardiac muscle. Steady state and transient analysis. We are writing programs to compute the solutions to the steady state cable equation for the one, two, and three dimensional cases (e^{-x}, $H_0^1(jx)$, and $xH_{1/2}^1(jx)$ respectively). This theoretical curve (one of the above) will be displayed on the oscilloscope of the LINC along with the experimental measurements of $V(x,\infty)$. The sum of the squared vertical differences between the experimental points and the theoretical curves will be computed and simultaneously displayed. A fit will be obtained by varying the vertical and horizontal scales by means of the parameter knobs of the LINC A/D converters. These will be adjusted until the variance is minimized. The basic membrane parameters can then be calculated from the scale factors and other data on current strengths both outside and inside the cells.

Once the general form of the spatial decay of voltage is determined, an attempt can be made to compute the transient response for each distance and to compare these to the observed curves. From all this analysis, one should be able to make reasonable estimates of the cell-to-cell resistance in cardiac muscle.

B. Ionic conductance changes during repolarization in cardiac muscle.

1. The experimental procedure is somewhat the same as described above in the determination of the equivalent circuit of cardiac muscle. An exception will be that the currents will be applied at various times throughout the prolonged cardiac muscle action potential (ca. one sec. in frog ventricle). The LINC can be programed to apply the current pulses at various times.

2. The equivalent circuit developed above will be used in an attempt to calculate membrane current density in the immediate region of the stimulating electrode. Such calculations will give membrane current density-membrane voltage curves from which specific membrane conductance can be calculated.

VI. We would definitely like to retain the LINC in our laboratory. Much time and effort has been expended in writing programs for the particular problems mentioned and in making the LINC an integral part of the experimental equipment and procedure. The data collection and storage, experimental control and data analysis features of the LINC have been found to be extremely advantageous in the experiments we have wanted to do. Our major complaint is in the time taken for programming and this is almost completed for our present problems.

We have been particularly impressed with the use of the LINC in experimental control. Experiments that would be marginal or impossible because of difficult procedures involved, are made possible with the control features of the LINC. The ability to collect and process easily large amounts of data is another feature of LINC that is particularly important to our research. Averaging of our low level noisy signals is essential to obtain good data. The LINC has become such an integral and essential part of our research program that it would be difficult to do without it.

SUPPLEMENTAL INFORMATION

I. Past and present research

Several investigators have used the LINC in this laboratory. These were mentioned above in section 4. Only the research of the principal and co-investigator will be discussed here. The reports on the research of the other investigators, Dr. D.R. Firth, Mr. W. H. Calvin, and Mr. W. L. Hardy are attached at the end of this report. Of special note is the discussion of the remote control use of the LINC by Mr. Calvin.

The use of the LINC in the determination of the electrical equivalent circuit of cardiac muscle was described above in rather general terms. More specific details of the data collection and storage program will be given here. This program involved use of much of the input-output, external sensing and control features. In all the program utilizes:

4 external A/D channels
8 potentiometer A/D channels
6 sense switches
4 external level lines
5 relays
both magnetic tape units
oscilloscope, 2 channels

The programs have been written for the 2048 word memory. All are entered from a control program under key board control.

A. Data Collection Program. The current pulse used in measuring the equivalent circuit of cardiac muscle is applied either through an extracellular suction electrode or through an intracellular microelectrode. In either case the potential measured by this electrode must be sampled. One of the SAM lines samples the output of the preamplifier with electrometer input. A 50 mV. calibration signal is sampled on this line when a calibration relay is closed. This same relay puts a 50 mV. calibration signal on the input of the microelectrode used to measure the voltage change. This calibration signal is used as a comparison level by LINC to determine when the microelectrode is actually measuring a "respectable" resting potential. While the investigator is manipulating this microelectrode trying to impale a cell, the voltage measured by this microelectrode is sampled and compared to the 50 mV. calibration signal. When the measured resting potential is large enough, the LINC then proceeds to the voltage-current sample program. This can be by-passed when the extracellular potential is being measured with an SXL. The calibration signals and the potentials measured by the current and voltage electrodes before the current pulses are stored with the record for display later. After the current-voltage data is collected, these potentials are sampled and stored for comparison with the initial values. This enables the investigator to check that conditions did not change during the data collection.

The main data collection program samples both the current and voltage while the pulse of current is being injected. Both current and voltage are displayed on a Tektronic 502A oscilloscope which is used both for monitoring and amplification. The output of the dual-beam 502A vertical amplifiers are then sampled by 2 A/D channels. Because the voltage response to the current pulse may be in the millivolt range while the resting potential is about 70 mV. the gain must be changed.before the current is applied. This is done using one of the LINC relays to switch the output of the voltage amplifier from a variable attenuation position to AC coupling. The automatic performance of all these routine functions releases the experimenter for other functions and permits the collection of data which would otherwise be missed; intracellular electrodes are easily dislodged from the small cardiac muscle cells and records must be obtained rapidly.

The data collection program samples the current and the voltage at rates which can be set from the keyboard. To make the program applicable to both current pulses in the quiescent heart and in the active heart, the sample rate can be changed from an initial fast rate of either one voltage sample every 32 or 64 microseconds to a one voltage sample every 96 microseconds to every 200 milliseconds. This changeover of rate can be set to occur at certain fixed intervals from the start to the completion. This is done so that the rising phase of a cardiac action potential could be sampled at the fast rate needed while the plateau and repolarization phase could be sampled at a much slower rate.

The start of the sampling is triggered by a pulse on one of the external level lines. The delay from this trigger can be set to any particular value up to about 16 seconds under keyboard control. To attain the maximum sampling rate while sampling both current and voltage, the program is set up to sample two voltage readings, one current reading, two voltage readings, etc. This causes no confusion of "holes" in the important voltage data since the LINC averages a number of records before the sampling is concluded. The starting of sampling is change in a cycle of three so that there will be a voltage sample on one record where there was a "hole" in the previous record. Thus to obtain two voltage data points at any one time from the trigger pulse, three voltage traces must be recorded. The number of voltage records in the average is set by 2 sense switches with a maximum of 24 or 16 "complete" records. The averaging is necessary because of the noise in the low level signals. Much of this noise comes from the high resistance (10-80MΩ) of microelectrodes. In a complete record there are 252 voltage points and 63 current points. The usual sampling rate is once every 32 microseconds.

A calibration signal for the voltage records is provided by closing a LINC relay which puts 2mV. at the voltage input. The LINC relays are slow to close, hence the AC coupling time constant must be fairly long so that the calibration signal will not have decayed too much before it is sampled. To get around this problem we have had the LINC relay close a fast relay closing in less than 1 msec. Activation of this relay triggers the LINC to proceed via an external level line. Thus the delay can be much less. The calibration signal is stored along with the data.

The distance that the voltage electrode is from the current electrode is set on a helipot which is sampled using two LINC relays to give the maximum, zero, and actual setting. This information is stored along with the record.

B. Data Display and Store. Once the data has been collected, it is displayed while LINC waits for a key to indicate storage on mag tape or rejection. The data display includes the voltage average, the last voltage record, the last current record, a voltage axis generated by the calibration signal, a time scale set by the sample rate, the cross over point where the sample rate changed the value of the voltage calibration signal, the values of the time scales for the two sample rates, the delay from the trigger pulse, the value of the distance from the current electrode in microns, the initial and final potentials measured by the current and voltage electrodes with their 50mV. calibration signals and the block number where it will be stored. All this is shown in Figure 1. In this figure, the upper trace is the output of the voltage amplifiers; the middle trace is the average of 16 complete voltage traces; the lower trace is the current pulse. Along the left side the voltage calibration increments are two mV. as indicated in the upper left hand corner. The time scale is one millisecond per division along the abscissa as indicated in the lower left hand corner. The distance reading is 95 microns, upper middle. The initial block number of storage is 6 as indicated in the lower middle. The figures in the upper left hand corner give the delay from the trigger pulse. The horizontal lines along the right side indicate the potential that would be measured by the voltage and current electrodes with 50mV. calibration signals. The flow chart for this program along with the data collection program is included in the appendix.

C. Display for Editing. Once the data is stored on tape, it can be retrieved and displayed. The block number of the data read in is controlled by the RSW. This data can then be stored in the appropriate blocks under keyboard and LSW control.

D. Transmembrance potential-voltage difference program. To obtain the transmembrane potential record, the difference between the voltage measured outside and inside the cell must be taken. This can be done under keyboard, RSW, and LSW control. Figures 2 and 3 show an example of the action of this difference program. Figure 2 shows the two records to be used in the difference. The upper record is the same as in Figure 1, corresponding to the intracellular record. The lower record corresponds to the extracellular record. Both are averages of 16 records. Figure 3 shows the difference between these two records (averages) displayed with all the record information as that mentioned for Figure 1 above. The transmembrane record can be stored under keyboard control.

II. Future research

A. Of most immediate interest is the calculation of the equivalent circuit of cardiac muscle using the data collected with the programs just described. This has been described in some detail in the first section of the report. It basically involves fitting of theoretical curves to the experimental points. The theoretical curves will be of the form of e^{-x}, or $H_0^1(jx)$, or $xH^{1}_{-1/2}(jx)$ depending on the geometry of the cardiac muscle section used. The parameters used in these functions will be taken off the knobs with some measure of the least squares fit to the data printed out on the screen along with the parameters. This program is in the planning stage. Data can be plotted out on a Moseley X-Y plotter by a method we have previously communicated.

B. Once this equivalent circuit is obtained, it can be used to calculate the effects of transmembrane current density of membrane potential from applied current-voltage curves obtained throughout the prolonged cardiac action potential. This work is described above in the future research section. Most of the programs written for the equivalent circuit study can be used here with minor modification. LINC will be particularly useful in controlling when the current pulse will be applied and also in calculating the effects of the applied current pulses from the equivalent circuit. These experiments are in the planning stage.

C. A couple of engineers here have wanted to set LINC up to obtain and store transistor characteristics to assist in circuit design. This seems to be feasible. The engineers are at present planning the sircuitry needed.

III. Training program

The LINC evaluation program has been of great value in demonstrating to many investigators the value of a digital computer inddata collection, experimental control, and data analysis. Although digital computers have been in use for some time here, we have benefited immeasurably by the LINC program. We have given two courses on digital computers and the LINC in particular, several graduate students have made LINC an integral part of their experiments and many others have learned more about digital techniques. The LINC is set up to demonstrate clearly many of the basic principles of digital computers.

During the evaluation program, it bacame clear to us that one of the main drawbacks was the amount of time taken in programming the LINC. More assistance in programs of a very general nature and in encouraging and promoting program sharing could have been helpful and still would be to a limited extent.

IV. Computer performance

A. We have experienced no serious breakdowns with the LINC. Minor troubles have come up with the keyboard picking up bits, but this has been remedied by the new keyboard sent to us from St. Louis. Several of the computer buttons stick occasionally and sometimes one of the switches does not function properly, but these are once a month occurances and are minor.

Little time was lost in the changeover to the 2048 word memory. The MDEL 2 setting was found to somewhat critical with the final setting somewhat shorter than anticipated, 1.56 microseconds.

It is felt that the relays used in the LINC are too slow and variable in their closing and opening times. We have as a result used these relays to drive faster ones, Magnecraft magnereed dry reed encapsulated relays, and sensed when these were activated using an external level. We recommend faster relays.

B. In our present experiments, we are operating well within the limits of the LINC computer. It is conceivable however that in the near future we will feel the need for a computer with more powerful calculational capabilities.

In this case it would be convenient to be able to transfer the data to a larger computer. This would require the production of IBM compatible tape or a direct communication system. The use of the expensive tape transport producing IBM compatible tape is certainly a possibility as has been shown at a number of labs. We hope to take another route by coupling the LINC directly with a larger computer, the Raytheon PB 440. At present the two computers are too far apart spatially. In the near future the two computers will be in adjoining rooms so that direct communication will be possible. For this procedure, the LINC is well equipped with the OPR instruction which has a wide variety of synchronizing levels and pulses.

There may be some problem in attaining high speed in this data transfer between the two computers. This comes because the LINC is not fully buffered on the input and output. Thus the whole computer is tied up while the data is being transfered. This is really not much of a problem in our laboratory, but it does point up one disadvantage of the LINC. However, the fully buffered operation is somewhat wasteful of the computer and costly since it does not make use of most of the available registers as do most of the LINC input-output operations. In our experiments we have not yet come upon situations where we would need this fully buffered capability but this is not to say that we will never need it.

In some experiments, a larger memory is essential. We have not had to face this problem. For those people with this problem it would be helpful if the LINC were somewhat more modular in construction so that extra memory, extra input-output devices could be incorporated with little difficulty.

"We have gradually come to realize that the most powerful feature of the LINC is the ability for experimental control. Our programs and experiments have evolved in this direction. The relays, external levles, A/D, etc. make it a powerful tool in experimental control. These features should certainly be retained. At times we have desired an output voltage (D/A on some register) that can be held for some time while the computer is sampling, etc.. As both the A and B registers are changing during this time, some sort of buffer register is needed. Of course we could wire this in externally. Would such a register be feasible if it could be utilized for some other task in the LINC (divide or fast storage and retrieval)?

With the price of transistor circuits falling daily, it is conceivable that a computer with greater computational capabilities could be built for about the same cost as the orginal LINC's. Modular construction for easy expansion of memory, input-output, could be of great advantage. The big question centers around how well this has been done already and whether this can be improved on.

V. Log Book

Attached are the important pages of our log book. The other pages contain copies of your letters to us on equipment changes,etc.

VI. Bibliography

The only published research which has used LINC to date is

Firth, D.R., Biophys J., Abstracts, Annual Meeting, 1965.

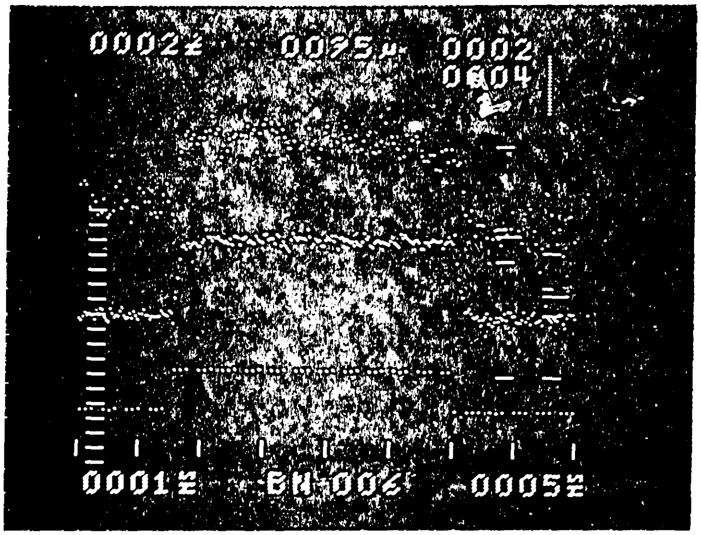


FIGURE 1

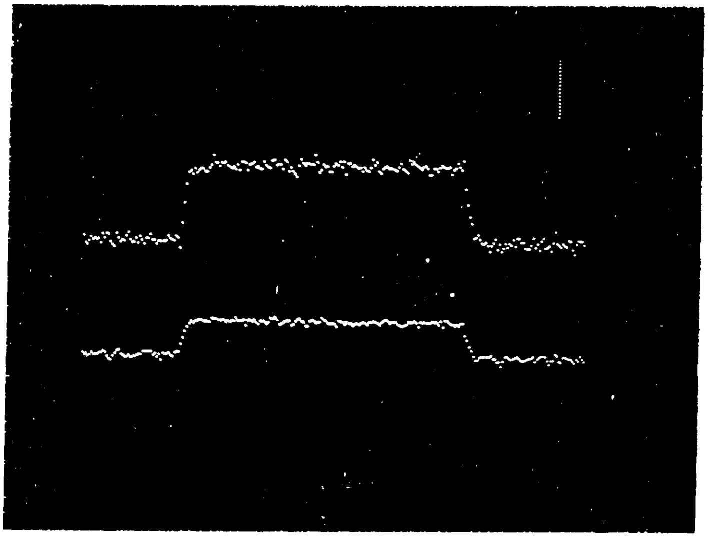


FIGURE 2

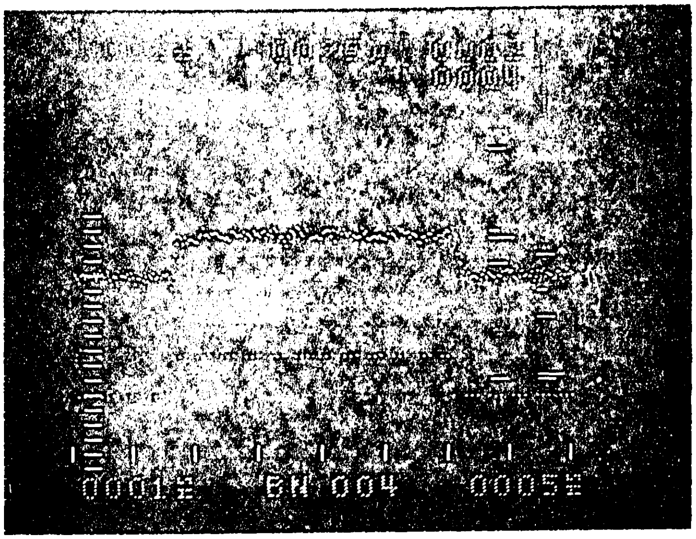
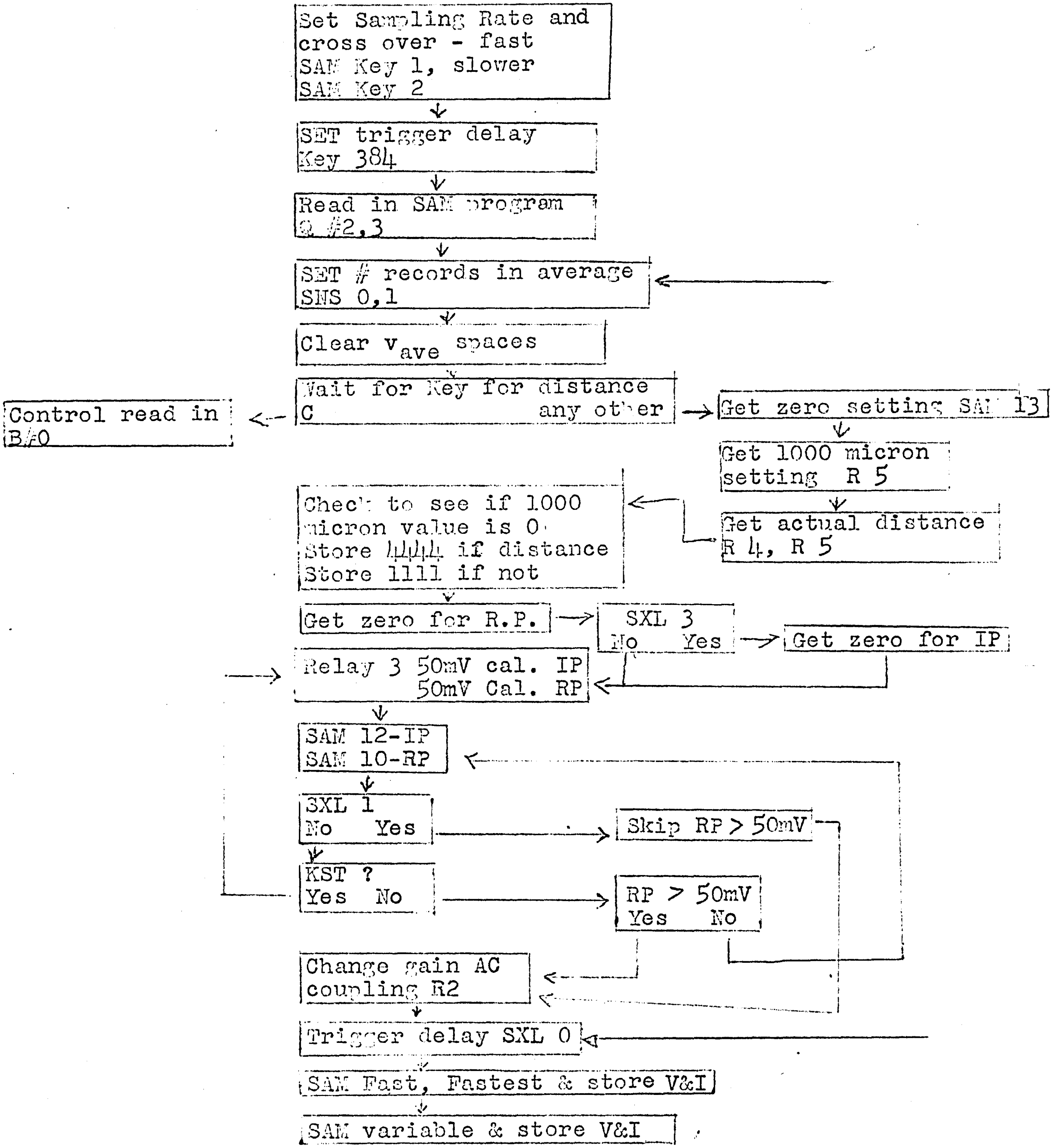
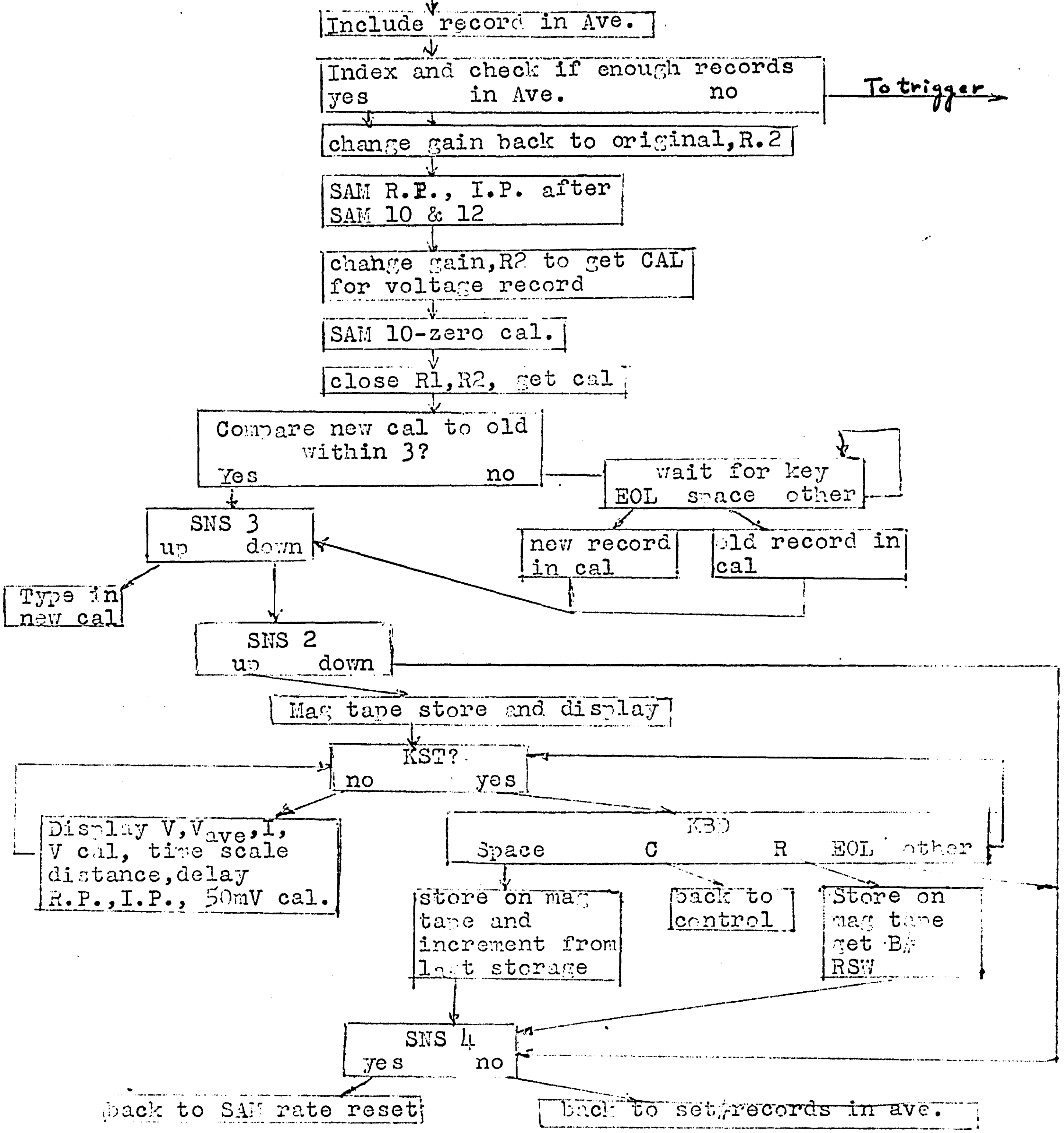


FIGURE 3

Flow Chart of Voltage-Current Sampling Program




Include record in Ave.
Index and check if enough records in Ave.
yes
no
To trigger
change gain back to original, R.2
SAM R.P., I.P. after SAM 10 & 12
change gain, R2 to get CAL for voltage record
SAM 10-zero cal.
close R1, R2, get cal
Compare new cal to old within 3?
Yes
no
wait for key
EOL
space
other
new record in cal
old record in cal
SNS 3
up
down
Type in new cal
SNS 2
up
down
Mag tape store and display
KST?
no
yes
Display V, Vave, I, V cal, time scale distance, delay R.P., I.P., 50mV cal.
KBD
Space
C
R
EOL
other
store on mag tape and increment from last storage
back to control
Store on mag tape get B# RSW
SNS 4
yes
no
back to SAM rate reset
~~Back to set records in ave.~~

COPY

April 13, 1964

To: S. M. Ornstein

From: J. W. Woodbury and A. M. Gordon

Subject: Plotting data from LINC with an analog X, Y plotter

Your memo of March 24, 1964, item 3 prompts us to report a simple and (if we do say so ourselves) elegant means of making X, Y plots on any analog X, Y plotter that has provision for raising and lowering the pen. The method requires no extra equipment in addition to the X, Y plotter and no alterations to the LINC. Further, the plotting rate is controllable from the LINC console and can be adjusted to fit the nature of the plot, i.e. several different curves can be plotted at once if the rate is made sufficiently slow.

Procedure: The X value to be plotted is put in a beta register (say 1) and the Y value put in the accumulator. (2) The plotting order is SET 2, 0001, where 2 is any beta register not used elsewhere in the program. (3) The left switches are set to 0002 and the XOE button pushed. (4) The start program button is pushed; LINC will stop at 0002 with Y in A and X in B. This is the first point to be plotted and analog values of these numbers appear at pins A (Y value) and J (X value) on the right hand display scope plug-in-unit plug. (5) The Auto-restart button is pushed and the program resumes until the program reaches the XOE stop at 0002 and the next X and Y values are in the B and A registers respectively. Thereafter, successive values appear automatically. The Auto-restart frequency can be varied according to the plotting job being done; if the points are close together and the pen does not need to be lifted in between successive points, the rate can be rapid, etc.

Pin E = GND

Pen raising and lowering. The pen can be raised and lowered by means of the ATR order and connection from the proper relay terminals on the data terminal box to the pen lifting circuit on the X, Y plotter. A sample program for loading a point, lowering the pen, then plotting the values in successive memory locations with the pen down and raising the pen at the end of the program is attached. Clearly, the points can be calculated between plotting orders, but this becomes too slow if the calculation time per point is appreciable. This method has been used successfully on the Moseley Model 2D and the EAI 1110 X-Y plotters. A spiral, plotted on the Moseley is enclosed. The distance between points is too large, accounting for the slightly wavy nature of the line.

Scope display. The data to be plotted can be checked on the scope unit with the same program simply by increasing the Auto-restart rate (with X, Y plotter turned off) and increasing the intensity of the scope beam by means of the screwdriver intensity control. The program enclosed can be made repetitive by changing 0062: JMP p-2 to 0062: JMP 20. This could be done with a SNS switch.

Additionally, only slight modifications are required to plot out any data displayed on the scope.

Sincerely yours,

J. W. Woodbury, Ph.D.
Professor

JWW:ch

Enclosure

Program for plotting Y against X

on an analog X, Y plotter

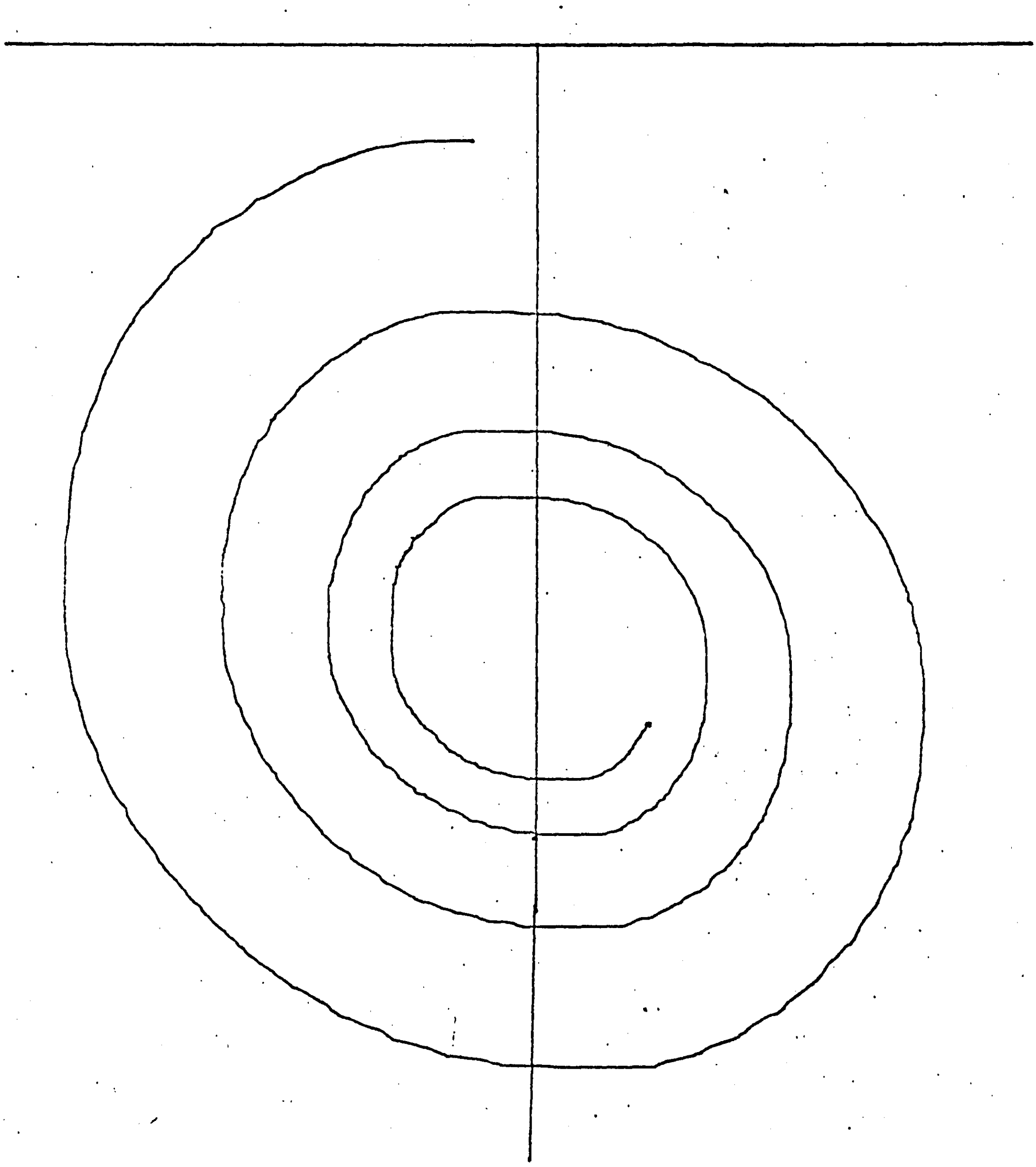
X values are stored in 100 to 777.
Y values are stored in 1100 to 1777.

Left switches set to 0004.

Start 20.

```
    0020 CLR
    0021 ATR
    0022 Set i 1
    0023 0077          9A.
    0024 Set i 2       9B
    0025 1037
    0026 LDA i 1
    0027 STC 3
    0030 LDA i 2
    0031 STA i                                X value in 3
    0032              #9C                     Y value in A
    0033 SET 4                                X value in B
    0034 0003
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  XOE STOP
    0035 LDA i
    0036 0001
    0037 ATR                                  Close relay, pen down
    0040 LDA    ← 9C
    0041 0032 Y value in A
    0042 SET 4
    0043 0003 X value in B_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _  XOE STOP
    0044 LDA i 1   #9D
    0045 STC 3                                X value in 3
    0046 LDA i 2                              Y value in A
    0047 SET 4                                X value in B
    0050 0003
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  XOE STOP
    0051 XSK 2                                Test for end
    0052 JMP 0044   9D                        repeat
    0053 STA i                                Save Y
    0054
    0055 CLR
    0056 ATR                                  Open relay, pen up
    0057 ADD p-3                              Y value in A
    0060 SET 4                                X value in B
    0061 0003
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  XOE STOP
    0062 JMP p-2                              End, pen stops at last point
```

Procedure: X and Y values come from the D/A converters on the B and A registers. These can be found at pins J (B register) and A (A register) of the plug on the right hand plug-in unit of the display scope. Pen raising-lowering contacts are connected to the proper terminals of Relay 1. To start, set left switches to 0004, push XOE STOP button, push START 20, and then AUTO RESTART. Have AUTO RESTART previously set for longest period. Once plotting has begun, AUTO RESTART delay can be decreased until maximal accurate rate is obtained.



UNIVERSITY OF WASHINGTON
SEATTLE, WASHINGTON 98105

School of Medicine
Department of Physiology
and Biophysics

PRESENT RESEARCH INVOLVING THE LINC - W. H. CALVIN

I. Analysis of Correlations between Interspike Intervals and Synaptic Noise

An analysis of interspike intervals has been undertaken to study the effect of synaptic noise upon interval variability. A mathematical model of interspike interval suggests that, under certain conditions, interval distributions can be predicted from synaptic noise data distributions. In this model, first passage time distributions have been found analytically by assuming that the membrane potential rises linearly towards threshold, with noise fluctuations added to this ramp which are correlated in time and Caussian in amplitude.

Experimentally, interspike interval distribution and in addition those parameters necessary to predict the interspike interval distribution from the theory must be measured. Measurements on the synaptic noise are probability densities and autocorrelations; intracellular recording is necessary for these measurements. Spinal motoneurons are being used because of their relatively large size. An intracellular microelectrode also permits some control over the average firing frequency of the cell by means of depolarizing or hyperpolarizing currents through the electrode. Autocorrelations require a long record of noise data uncluttered by spikes, thus action potentials are eliminated by depolarization or hyperpolarization block of the cell.

A variety of LINC programs are utilized to collect the data, to inspect it to see if it is in accord with theoretical assumptions, to correct for drifts in the mean intervals, to construct interspike interval distributions, and for histogram and joint probability displays.

In addition, LINC has been used to simulate the theoretical model in real time by generating ramps and adding noise signals fed in over the sample lines from noise generators of known autocorrelations.

Two appendices give details on a simple remote control which utilizes a telephone dial, and on a histogram display with coordinate lines and labels which are invariant under changes in scale and origin.

A. Data Collection Programs

Samples of the intracellular potential are taken every 200 microseconds (mean interspike interval is generally about 100 msec.). In one type of sample program, the voltage samples are stored sequentially, resulting in a record 307 msec. long before memory is filled and sampling must be briefly interrupted to store the memory contents on magnetic tape.

Another sample program counts and stores only the intervals between spikes, so that more than 1500 intervals may be stored before tape storage is required. All information about the potential between spikes is lost. Since the requirements for verifying the theory require that some information about the potential between spikes, a third type of sample program has been developed which stores 256 sequential interspike intervals, along with their threshold voltages, and four samples for each interval of the interspike potential, taken at fixed times following a spike.

A 200 microsecond basic sample loop is particularly convenient for timing purposes. A 64 microsecond sample interval program, triggered from the shock artefact, allows detailed study of the shape of spikes and afterpotentials.

B. Remote Control of LINC

These experiments are carried out in a neurophysiological laboratory which is about 150 feet from the LINC. At present, four lines connect the laboratory with the computer: one data line, one experiment-to-computer control channel, and two computer-to-experiment feedback lines. Existing power distribution lines, formerly used to connect each room to a central DC battery supply and timing pulses, were found to have excellent high frequency response (microseconds at least) and noise pickup in the millivolt range. Data is amplified to the ±1 volt input range of LINC before transmission to the LINC; added noise is negligible.

Remote control of LINC is accomplished by having LINC sample the voltage on the control channel: a negative voltage on this line means to sample data, a positive voltage terminates sampling and sets the LINC for further instruction by dial pulses. The telephone dial is used in much the same manner as the keyboard: to select sample programs and to change constants in the programs. The digits one to nine are used to select sample programs. A zero tranfers the computer into a dial-a-constant subroutine; the memory address and contents of that location are dialed successively.

Feedback of information from LINC to the experiment utilizes two additional lines run off of the LINC relays. One line runs a veeder counter, which is pulsed to show how many tape blocks have been used. This information can be entered in the experimental protocol to show the exact decimal location of a given record on the tape. The veeder is pulsed only after sampling has been halted by throwing the SAMPLE-STOP-DIAL switch from sample to stop, not after each intermediate memory dump.

The second feedback line runs a red light on the panel, which is on whenever sampling is taking place. It goes off during memory dumps, allowing the experimenter to guage how fast the mag tape is being used. (The veeder is not pulsed until sampling is finally terminated). Also, should the computer hang up, the light will cease responding to switch, since the light should go on shortly after the switch is thrown to SAMPLE. This line may also be used to activate the shutter of a Grass camera to give a film record of exactly what the computer has taken in.

The addition of a slave of the computer scope, would give the experiment almost a complete LINC (input, output, and controls) by remote control. Work is currently underway to utilize a Tektronics 360 scope as a slave. Printed displays will tell the experimenter the current tape status and which sample program is loaded. Data processing and display of the results on the slave scope will also be possible.

This is not the ultimate, however. The experimenter still must run frantically down the hall in order to change tapes. An additional tape deck would be a worthwhile improvement.

C. Analysis and Display Programs

For the examination of raw data, a general purpose display program is used, with knob control of first and last location, abscissa expansion and compression, and ordinate zero. Decimal legends for each of these is provided in the upper left hand corner of the screen. This program includes a cursor, also under knob control, which will read out the value of a point upon which it is superimposed. A straight line generator allows curve slopes to be determined, with the line slope also printed on the screen. Thus the drift in the mean interspike interval (ISI) may be determined. In addition, drift may be taken out of the data by another feature of this program. Upon fitting a straight line to the plot of the successive ISIs, the program will then subtract the value of the line ordinate from each successive datum, resulting in ISIs that now fall along a horizontal straight line. The ISIs may be smoothed by means of a sliding average, so as to make it easier to fit the straight line to them. The straight line slope thus determined will then be used on the original unfiltered data to obtain drift-free mean ISIs for statistical computations. It should be noted that this tatic is used occasionally only because of the great difficulty of obtaining completely stationary statistics from intracellular recordings. In general, this would not be necessary with good extracellular recordings which remain statistically stationary for sufficient periods of time.

Interspike intervals are compiled into histogram form by sorting them into bins, generally of one msec. width (5 counts of 200 msec. each).

This program uses the ISI to determine the address of the corresponding bin which is then tallied by one.

The interspike interval distributions (i.i.d.) are displayed by means of a histogram program, which displays each bin as a filled-in bar on the screen. The width of an individual bar on the screen is adjustable on the knobs and the number of bins represented by each bar can be changed. The vertical scale is controlled by a knob. Coordinate lines, both horizontal and vertical, are selected one-by-one by typing in their decimal location in milliseconds or spikes per bin. A coordinate line at, say, 50 milliseconds will stay with the 50 msec. bin even if the scale is changed or the origin shifted. A three line legend can be typed in at the top of the screen. For further details, see the complete write-up in Appendix B.

For the analysis of joint probability densities, another type of display program has been devised. A three dimensional display is gotten around in the standard manner, by displaying each data pair as a point on the screen, and then looking at the density of points on the screen. This is used for examining joint interval distributions (a given interval as a function of the immediately preceding interval) for correlations between succeeding intervals. It may also be used for plotting interval as a function of threshold potential, in order to examine the effect of accommodation upon i.i.d. Time exposures are taken as the program goes through all of the raw data. Drifts in mean ISI can be detected very well with this program, as the concentration of dots will appear to move about the screen as they are plotted.

D. Simulation of i.i.d. model

Whereas the equation for the i.i.d. may be obtained explicitly from the theoretical model, it is more difficult to get the i.i.d. explicitly when slight modifications are made in the theory. In order to find out how these modifications (such as corrections for accommodation) influence the shape of the i.i.d., it was decided to simulate the process in real time, which is possible with LINC. Analog noise sources of known autocorrelations are fed into the A-D sample channels and ramps are generated by programming, knobs controlling the slope and starting point of the ramp. When the sum of the ramp and noise reach the threshold level, the interval is stored in the same format as used for actual ISI data. The generated ISI's are then processed in the same manner as actual data.

E. Keyboard Control of Programs

In addition to the keyboard commands within the individual programs, there is a master program control in block 0. The various programs are selected and read in by this program, which also contains read and write subroutines corresponding to the block format used by the sample programs.

Some of these block numbers typed in octal, while others use decimal block numbers, so as to correspond to the veeder counter used in the remote control. Except for the initial reading of the master control block, no operations are done with the toggle switches and DO TOG. It is anticipated that many of the operations requiring keyboard interrogation will utilize questionnaires on the scope in future revisions.

II.

A. Receptive Field Representation by Matrices

Mathematical representation of receptive fields by single-valued functions being impossible, it was decided to use a matrix representation with the matrix elements representing the sensitivity of the receptive field (RF) at a corresponding point in space. Positive values indicate the relative size of an EPSP due to light at that point in the RF; negative values of a matrix element indicates the relative size of an IPSP due to light at that point.

To compute the net potential change (net EPSP or IPSP) the receptive field and the spatial distribution of the light stimulus must be known. The light spatial distribution may also be represented by a matrix with the value of the elements representing the light intensity at that point. The net response of the cell is obtained by multipling each element of the light matrix with the corresponding element of the RF matrix, and summing the elemental responses. This is not standard matrix multiplication, though it may mathematically be put into that form with two Kronecker deltas.

B. Output Receptive Field from Input Receptive Fields

We would like to be able to predict the RF of a cell from knowledge of the RFs of its inputs. In particular, we would like to see how center-surround RFs of retina and lateral geniculate may combine to produce the line and edge type RFs seen in primary visual cortex, and how they in turn might combine to produce the more complex (and "hypercomplex", yet) RFs seen in the visual association areas.

One of the obvious models for the line type RF of the primary visual cortex is that its inputs come from a series of center-surround RFs whose centers lie along a line in the visual field. If A_{ij} is the basic RF matrix of the center-surround type, then other similar matrices whose centers lie along a horizontal line could be obtained by shifting the columns of A_{ij} to the right or left (the matrix is assummed large enough that all of the peripheral elements are zeros and thus there are no problems with what to do with those shifted out the edge of the matrix). These shifted matrices are $A_{i,j-1}$, $A_{i,j-2}$ etc. The RF of a cortical cell would be represented by the weighted sum of the input RFs. If each were weighted equally, the output

RF matrix C_{ij} would be $C_{ij} = A_{i,j} + A_{i,j-1} + A_{i,j-2}$. This C_{ij} bears a strong resemblance to the line-sensitive RFs of the cortex. It predicts that the ends of such lines ought to have some inhibition, which has not been reported. Thus the model has suggested a simple experimental test of the model (experimental verification of end-effects is only a necessary and not a sufficient condition for the usefulness of the hypothesis, unfortunately). Similar such models may be constructed for the other types of cortical RF, with the end in mind to produce similar as-yet-unobserved RF characteristics with which to test the validity of the model. An extension of the model into the temporal behavior of the cell from various kinds of moving stimuli has been done for chopped intervals of time, although the continuous case appears possible, theoretically if not in simulation. One goal is to produce alternate theoretical models for each RF type, which is possible in many cases, and to then devise experimental tests that would distinguish one type of neuronal hookup from another.

III Comments on Advantages and Disadvantages of LINC

The tape block format has proved very useful, and the display scope invaluable. The limitation of the 12 bit word length has been felt, not just in more complicated computations but even in adding lists of numbers. Sample rate could be improved with a buffered input that did not tie up the accumulator. The full-address class instructions (ADD, STC, JMP) prevent one from using upper memory for programs. It hardly seems worth the restriction on upper memory to have ADD and STC when ADA and STA are available in two word format. A two word JMP instruction allowing use of upper memory for programs would seem to be a worthwhile design improvement. Similarly, with the index class instructions such as XSKi, it is a nuisance to have to program around the 1777-2000 step.

.0013

```
      -1
  -1-1 3-1-1
-1-1 3 3 3-1-1
-1 3 3 3 3 3-1
-1-1 3 3 3-1-1
  -1-1 3-1-1
      -1
```

$A_{i,j-2}$

.0013

```
      -1
  -1-1 3-1-1
-1-1 3 3 3-1-1
-1 3 3 3 3 3-1
-1-1 3 3 3-1-1
  -1-1 3-1-1
      -1
```

A_{ij}

.0013

```
      -1
  -1-1 3-1-1
-1-1 3 3 3-1-1
-1 3 3 3 3 3-1
-1-1 3 3 3-1-1
  -1-1 3-1-1
      -1
```

$A_{i,j+2}$

.0057

```
      -1   -1   -1
  -1-1 2-2 1-2 2-1-1
-1-1 2 2 5 1 5 2 2-1-1
-1 3 2 6 5 3 5 6 2 3-1
-1-1 2 2 5 1 5 2 2-1-1
  -1-1 2-2 1-2 2-1-1
      -1   -1   -1
```

$B_{ij} = A_{i,j-2} + A_{ij} + A_{i,j+2}$

.0083

```
    2   1   2
  2 2 5 1 5 2 2
3 2 6 5 3 5 6 2 3
  2 2 5 1 5 2 2
    2   1   2
```

Net excitatory part of B_{ij}

-0032

```
        -1   -1   -1
    -1-1  -2   -2  -1-1
  -1-1                -1-1
  -1                    -1
  -1-1                -1-1
    -1-1  -2   -2  -1-1
        -1   -1   -1
```

Net inhibitory part of B_{ij}

TELEPHONE DIAL REMOTE CONTROL SYSTEM

By W. H. Calvin

Department of Physiology & Biophysics
University of Washington
Seattle, Washington

There are three separate parts to the remote control system: the control channel, run off of the voltage divider circuit with the telephone dial and the DIAL-STOP-SAMPLE switch; the veeder counter, run off of LINC relay 0; and a red RUN LIGHT, run off of LINC relay 1.

A negative voltage on the control channel (SAM 11) always means to commence or continue sampling: this occurs when the switch is in SAMPLE position. When the switch is thrown to STOP, the control voltage goes positive into the 0.00-0.25 volt region ($0-37_8$ for SAM 11) and the sample program jumps to 1P, which causes the last data to be dumped and the veeder counter to be pulsed. In addition, a "control block" is written in the next tape block containing any parameters the experimenter may wish to insert (in a predetermined manner). The program then waits for the control voltage to either go negative again (switch back to SAMPLE) or for it to go further positive.

When the switch is thrown to DIAL, the control voltage jumps into the 0.25-0.50 volt ($40-77_8$) region and the dial reader control jumps into its dial reader subroutine (1D via 2P). One then dials a single digit specifying a sample program, which is read into Q1. (When a zero is dialed, one may go to a special subroutine which allows one to dial in program constants, etc.). When the switch is thrown to SAMPLE, sampling is commenced with the new sample program.

Note that data dumps that occur during the time when the switch is in SAMPLE do not pulse the veeder nor write control blocks. Thus minimum dead time is governed only by the tape-writing time, which is speeded up by using WRIiu for six blocks without checking the tape.

The run light, operated by relay one, gives the experimenter some idea of how the sample and control programs are operating. It goes on during sampling, and is off during data dumps. Thus one can get some idea of how rapidly the data is being filled up by counting the number of times the light winks off and on, before the veeder is pulsed at the end of sampling.

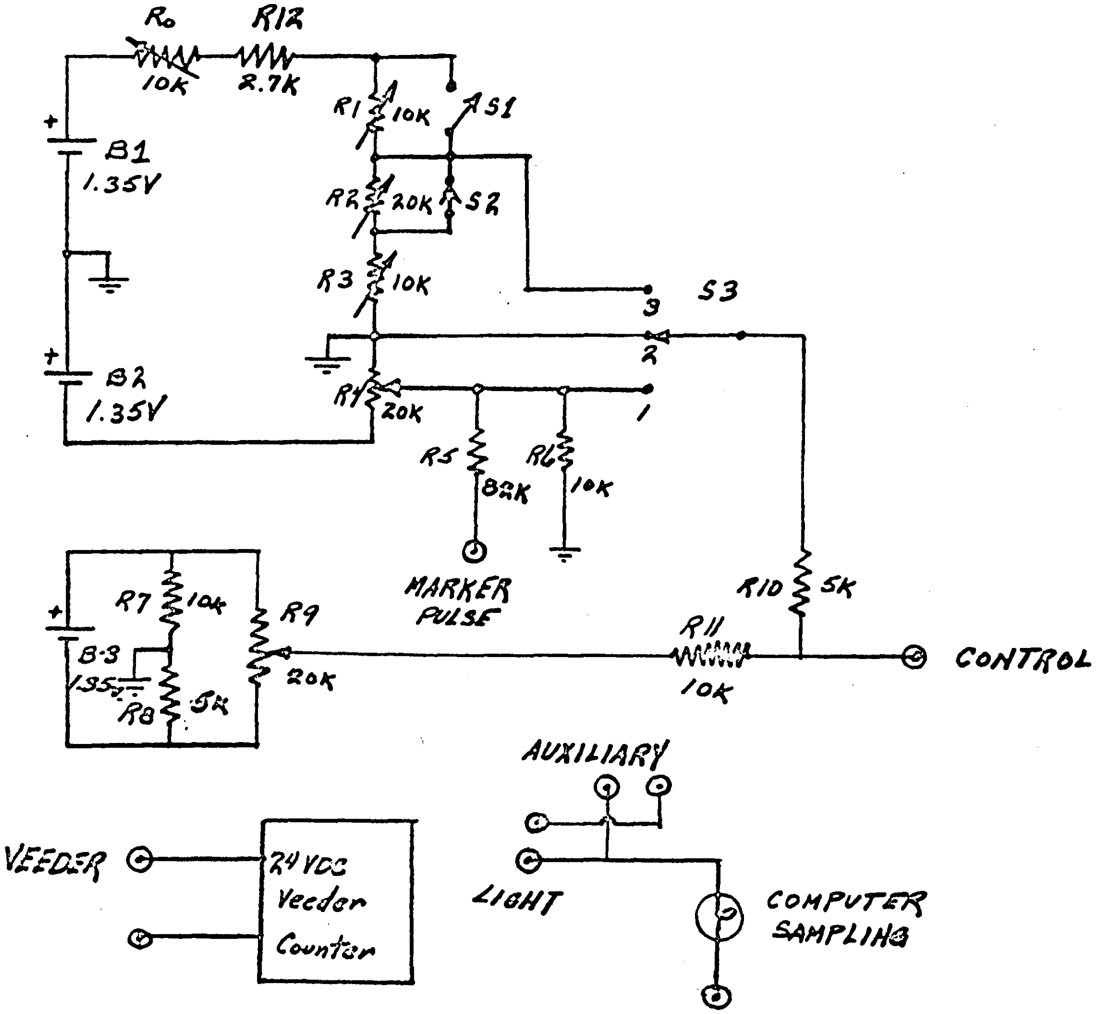
Work is presently underway to replace the veeder with a scope display in the lab. A Tektronics 360 scope will serve as a slave to the LINC console scope, so that block numbers may be displayed on the screen. In addition, questionnaires may be displayed on the screen and the ??? filled in by dialing. This will give considerably more versatility to the remote control, since even processed data can be shown after sampling has ceased. In short, one is getting almost a complete LINC by remote control for the price of a few analog lines and some very simple circuitry: one gets input, output, and controls (via a telephone dial (instead of a keyboard).

13-65

Computer Control- Dr. Steven's Rack

(Bill Calvin)

D.C.2.



Note:

S1 - N.O. contacts on phone dial

S2 - N.C. pulser contacts on phone dial.

S3 - 3 pos. lever switch; locking in all pos.

Position	Label
1.	SAMPLE
2.	STOP
3.	DIAL

-13-65

Computer Control Adjustment proceedure

1. Remove B-1 and adjust R-0 so that the total resistance of R-0 and R-12 are about 4k.

2. Install all batteries, place a scope and a 39k load across CONTROL terminals. Set lever switch to STOP. Adjust R-9 OUTPUT BIAS to a reading of .125 V.

3. Move lever to DIAL position and advance the dial sufficiently to close S-1, but not far enough to open S-2. Adjust R.3 D. Tone for a reading of .625 V.

4. Release the dial and then adjust R.1 D. GATE for a reading of .375 V.

5. Advance the dial, then back it off to open S-2. Adjust R.-2 DIAL PULSE for a e reading of .875 V.

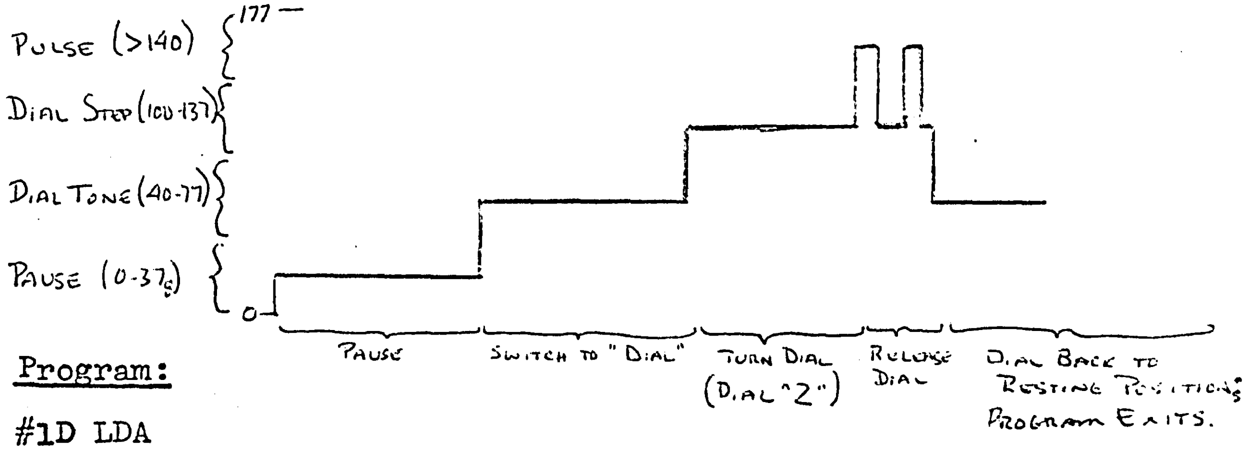
6. Release the dial, and move the lever switch to SAMPLE, and adjust R-4 N. BIAS to a reading of -.5 V.

(R-0 should be kept as large as possible to prolong battery life)

Dial Reading Subroutine
W. H. Calvin
University of Washington

- Exit when control channel voltage falls to dial tone region ($40\text{-}77_8$) with number of pulses left in ACC.
- When no pulses appear on top of dial step, or when a pulse is too long, it exits with -0 in ACC.
- When a zero is dialed resulting in ten pulses, the ACC is set to + 0.
- Uses index registers 12, 13, 14, and 15.
- Pulses must be of a minimum length in order for them to be counted (about 10 msec.). This is due to the contact bounce problem encountered with telephone dials.

Control Voltage Diagram:



Program:

```
#1D LDA
    0
    STC 2D
    SETi12
    0               set pulse counter to zero
    SAM 11          control channel
    ADAi
    -40
    APO
    JMP 3D          if V is not above Pause region (0-37),
    ADAi              it exits with ACC set to -0
    -40
    APO
    JMP p-10        if in dial tone region, waits for dial
    SETi15           step or return to Pause region
    1700
    XSKi15
    JMP p-1         time delay
```

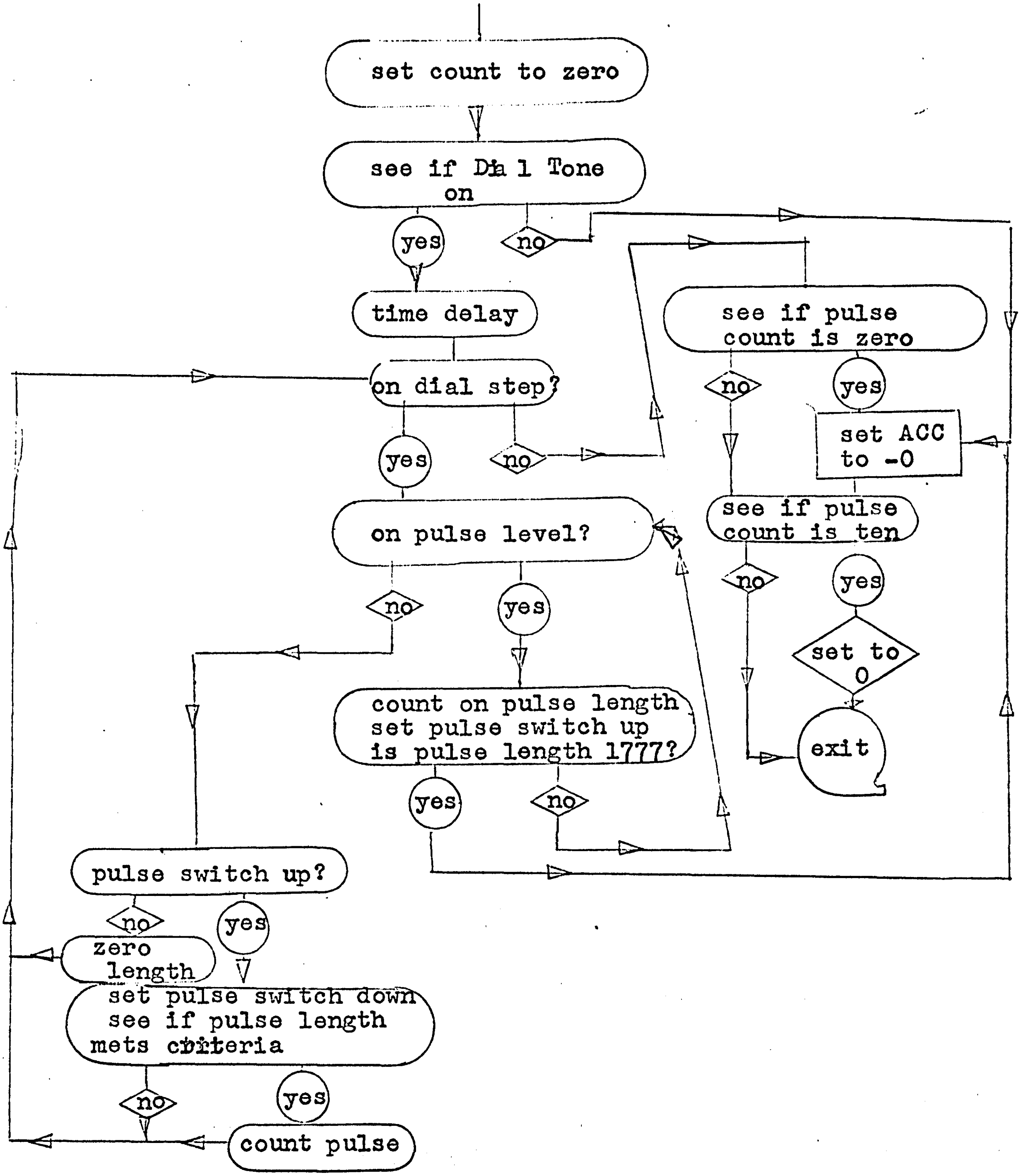
```
#5D SAM 11
    ADAi
    -100
    APO              dial step completed?
    JMP 4D           yes
    ADAi
    -40
    APOi             at pulse level? (140-177)
    JMP p+6          yes
    XSK 13           no, still at step.  Is this return from a
    JMP 6D          /yes/ pulse (is pulse switch up)?
    SETi14           no
    0                set pulse length to zero
    JMP 5D
    SETi13

    0                set pulse switch up
    XSKi14           count on pulse length
    JMP 5D
    JMP 3D           set output to -0 if pulse length reaches 1777

#6D SETi13           get here at end of a pulse (with switch up),
    1777              so set switch down
    LDA
    14               get length count
    ADAi
    -100             length criteria (about 10 msec.)
    APOi
    XSK i12          long enough, so tally pulse counter
    JMP 5D           not long enough
    JMP 3D           if ever get 1777 pulses (!), exit with -0.

#4D LDA
    12               get pulse count
    AZE
    JMP p+4          not zero
    ~~JMP 1D+3~~ NOP     if zero, ~~returns to start~~ exits
#3D LDAi             entry for -0 exit
    -0
    SAEi
    12               ten pulses? If so, set output to +0
    JMP p+2
    CLR
#2D JMP              out to p+1 of originating program
```

Flow Chart for Dial Reading Subroutine



Dial Reader Control Program
(use with Dial Reader subroutine and sample programs)

```
#2S SAM 11              get control voltage
    ADAi
    -40
    APO
    JMP 2P              in pause region (0-37)
    JMP 1D              in dial tone region or above;jump to subrt.
    APO                 return from subrt.: is result -0?
    JMP 2S              no,result, try again
    SAEi
    0                   test for zero dialed
    JMP p+2
    JMP 4S              dial-a-constant subroutine (not included)
    ADAi
    1030
    STC p+2             sample programs stored beginning in 31 BN, u0
    RDC
    --                  reads in sample program to Q1
    JMP 2P

#kRxkRkx
#1P CLR
    ATR                 turns run light off (run by relays)
    JMP 4P              dumps data
    JMP 1W              pulses veeder counter (relay zero)
#2P SAM 11
    SETi15
    1767                stalls about 300 msec due to non-shorting
    SETi14               switch (SAMPLE-STOP-DIAL) which sends control
    0                    voltage briefly into dial step region when
    XSKi14               switching from SAMPLE to STOP.
    JMP p-1
    XSKi15
    JMP p-5
    APOi
    JMP p+5
    LDAi                if control voltage negative, returns to
    2                    sampling after turning run light on
    ATR                  via relay one
    JMP 600             sample programs all start at 600
    ADAi
    -40
    APOi
    JMP 2S              if in dial tone region or above, jump
    JMP 2P               to dial reader control
```

WRITE subroutine
(use with sample programs & with dial reader control program)

```
#4P LDA
    0
    STC 7P
    ATR                         turn off run light
    SETi2
    -6
    LDA
    6P
    BCLi
    7000                        first quarter dumped is Q2; into BN
    ADAi                        following last one used.
    2000
    STC 6P
    WRIiu                       WRI used since check notneceaasry for data
#6P                             dumps Q2,3,4,5,6,7
    LDAi
    1001
    LAM
    6P                          neceaasary to prevent end around carry
    STC 6P                       after Q7 is dumped
    XSKi2
    JMP p-10
    CLR                         clear link bit
    MTBu                        stop tape
    0
    LDAi
    2                           turn on run light
    ATR
#7P
```

Veeder Counter Pulser
(use with Dial Reader Control)

```
#1W LDA
    0
    STC 3P
    LDA
    6P                get last BN+1
    BCLi
    7000
    COM
    ADAi
#5P                   get first BN (set)
    AZEi
    JMP 3P            do not count when equal
    ADAi
    1776              gets difference plus one (for control block
    STC 2                                        at end of data)
#2W LDAi
    1
    ATR               turns on relay zero (veeder counter)
    SETi3
    -1
    SETi4
    0
    XSKi4
    JMP p-1
    XSK13
    JMP p-5           determines time that relay is on
    CLR
    ATR               turns relay off
    SETi3
    -2
    SETi4
    0
    XSKi4
    JMP p-1
    XSKi3
    JMP p-5           determined time relay is off between pulses
    XSKi2             counts on number of pulses
    JMP 2W

     set up in Q2 any parameters you wish to save (sequence numbers,etc)

    LDA
    6P
    BCLi
    7000
    ADAi
    2000
    STA p 7
    ADAi
    1
```

```
     STA
     6P                    store incremented BN back in WRITE subrt.
     STC 5P                set new first BN in veeder pulser
     WRCu
     ---                   write control block following last data
#3P  JMP                   return to originating program
```

A sample SAMPLE Program Utilizing Dial Reader Control

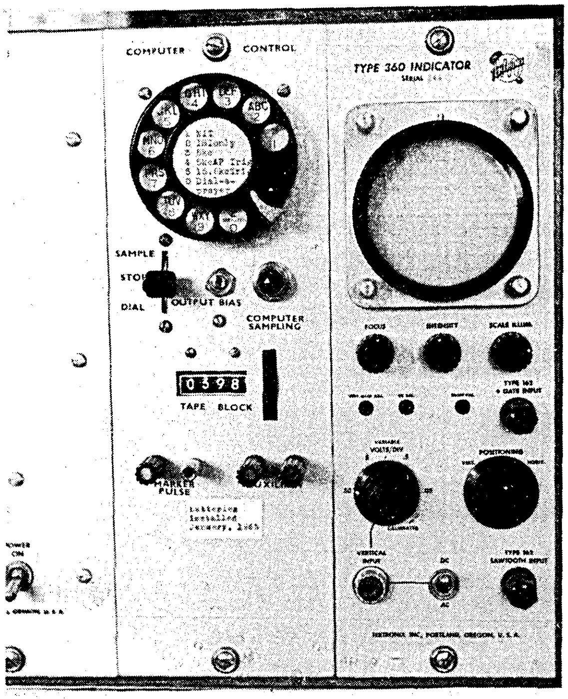
64 microsecond sample, triggers on shock artefact (-177)

```
origin 600
#1U SETi3
    -2
    SETi1
    777
    SAM 11              test control channel
    APOi
    JMP 1P              if positive, go to pulse veeder and dump last data
    SAM 10
    SAEi
    -177
    JMP p-6
    SAMi10
    STAi1
    XSK 1
    JMP p-3
    XSKi3               test for 1777-2000 transition
    JMP p+3
    JMP 4P              dump data when 3777 is filled
    JMP 1U
    SETi1
    2000
    JMP p-12
```

Telephone Dial

Remote Control System

for the LINC Computer



William H. Calvin
Department of Physiology
and Biophysics
University of Washington
School of Medicine
Seattle, Washington 98105

HISTOGRAM WITH INVARIANT COORDINATES

By W. H. Calvin

Department of Physiology & Biophysics
University of Washington
Seattle, Washington

General Description

This program displays filled-in bars for each point, with the width of the bar (and hence the total time width of the display) being adjustable. The origin may be shifted in time, so that the Histogram display could start at, e.g., 100 msec. Coordinate lines are typed in individually, in decimal milliseconds and spikes/bin, with the dot density of each line also individually adjustable. These coordinate lines remain at the appropriate place despite changes in either vertical or horizontal scale, or changes in origin. The value of eachcoordinate line is displayed in small characters at the beginning of each, again in the decimal millisecond or spikes/bin form. A three-line legend may be typed in at the top of the screen. Data blocks may be read in by keyboard control, and the usual differential distributions may be integrated. Sense switches allow one to turn off the legend or the coordinate lines, and to stall after one complete loop so as to achieve uniform time exposures when photographing.

A. The Histogram Proper

The vertical scaling $2^{\pm n}$ is accomplished with knob 5. The horizontal scaling is accomplished by setting the width of the bars via knob 7. Normally there will be one bin per bar, but this may be changed by typing "B" and then the desired number of bins per bar. Normally the display will start with the zeroth bin, but by typing "D" and then the delay in milliseconds (three digits, right-justified), one can start anywhere. Since a space of one is left between each bar, the maximum number of bars which may be displayed is 234_{10} (since the first bin is displayed at 54_8 to leave room for the display of the ordinate values at the left side of the screen). The data is kept in 3000-3777 and may be read into those locations from unit 0 by typing "T" followed by the BN of the first BN in octal.

B. The Invariant Coordinate Lines with Labels

By typing "N;" followed by a one digit code and a three digit decimal number, one may locate a single coordinate line. The one digit code is as follows: 0, 1, 2, 3 are vertical lines of increasing dot density; 4, 5, 6, 7 are horizontal lines of increasing dot density. Thus N7000 is a horizontal line of maximum dot density (spacing of 4) at zero spikes/bin. A vertical line of minimal density (spacing of 40 between dots) at 34 msec.

would be N0034. Since coordinate line display is terminated upon reaching a 0000 in the list, one may not have a minimal density vertical line at zero msec. One may delete the last coordinate line typed in by typing "N," then "del". The maximum coordinate, either horizontal or vertical, is 511_{10}. If the first of the four digit code word did not have to denote dot density and line direction, then this range could be expanded.

A vertical line starts at -340_8 and terminates when a dot exceeds 237 (so as not to run into the legend at the top of the screen) and have their labels beneath them. Horizontal lines start at 54_8 and run to 777_8 with their labels in the $0-53_8$ region.

Invariance of coordinate lines to changes of scale and changes in time origin is accomplished by examining the relevant constants in the histogram proper, and using them to calculate the coordinates of the line.

After the line has been dotted out, the last three digits of code word are converted to decimal and displayed in the appropriate place with a character generating subroutine. This subroutine utilizes the format of the DSC instruction, but with reduced spacing between dots. Dot spacings of two give characters one-fourth the area of a DSC character (with spacings of 4) while spacings of three give characters 9/16 the area of DSC characters. The label "MSEC." is also displayed at the lower right-hand corner of the screen.

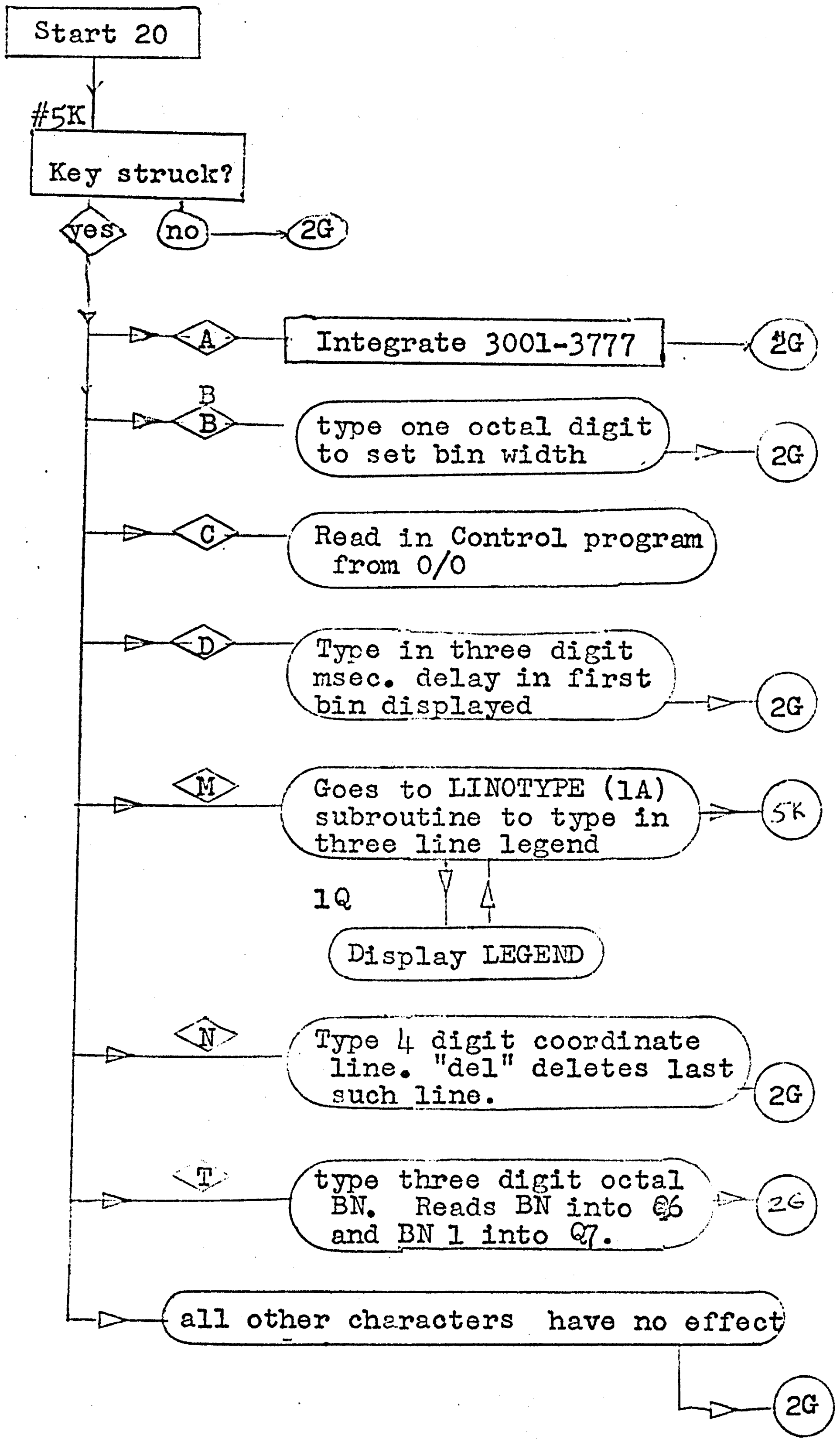
Caution should be exercised to avoid end-around effects. For instance, if the end of the screen is at 50 msec. and a vertical coordinate line is typed in at 75 msec., it will appear around the end of the screen at about the 15 msec. location.

C. Legend Display

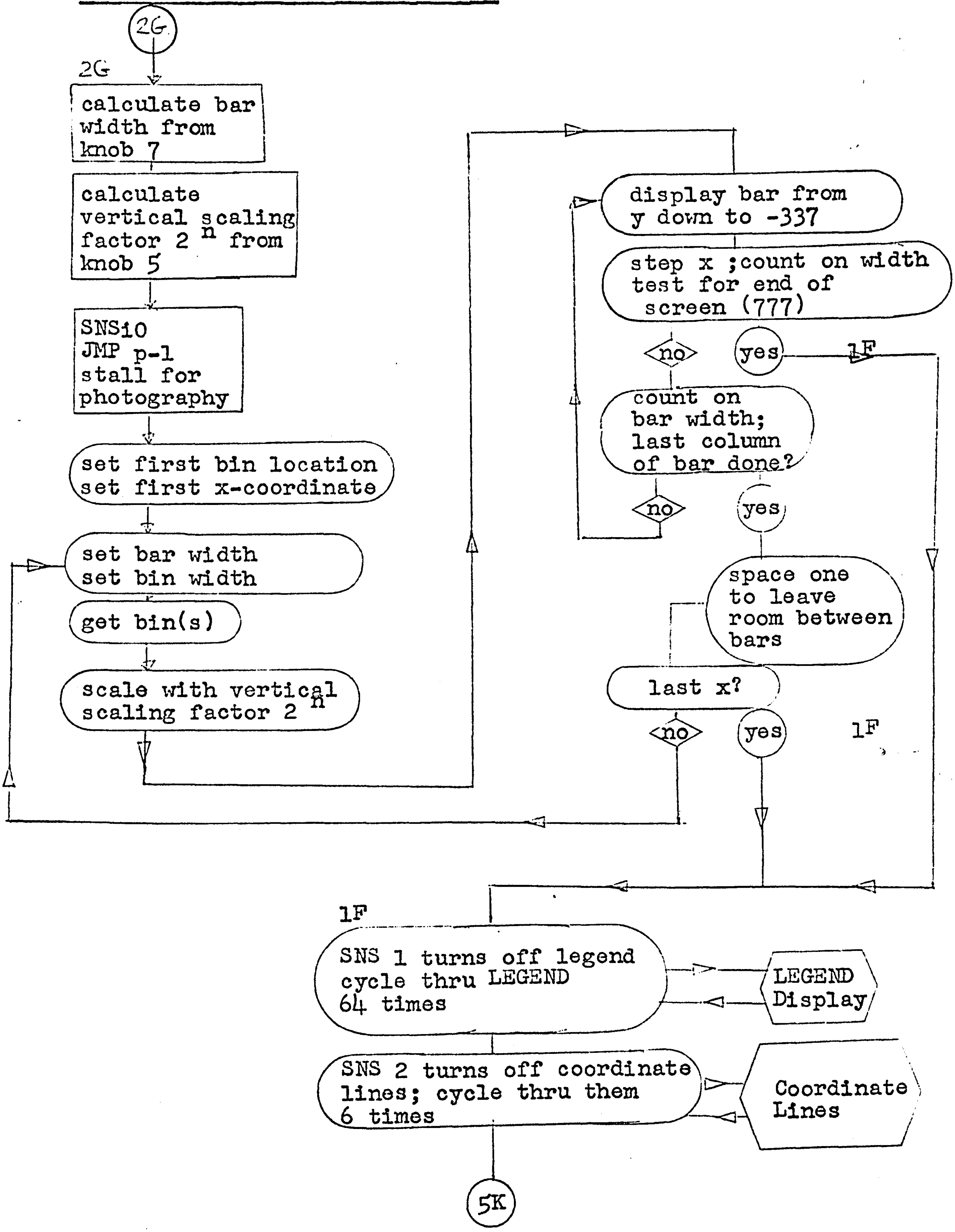
A three line legend is displayed at the top of the screen utilizing the DSC instruction. To type it in, one types "M" and then proceeds to type in the first line. An EOL shifts down to the next line, and after the third EOL, control is returned to the main program. The width of the screen will allow no more than 25_{10} characters per line. Pressing "del" will delete the preceding character, but care should be exercised not to delete back past the first character of the first line as one will start deleting program.

Since the DSC instruction's intensification pulses are shorter than DIS, one must display the legend nearly 100_8 times to have it appear equally as bright as the bars of the histogram displayed once, when taking time exposure photographs.

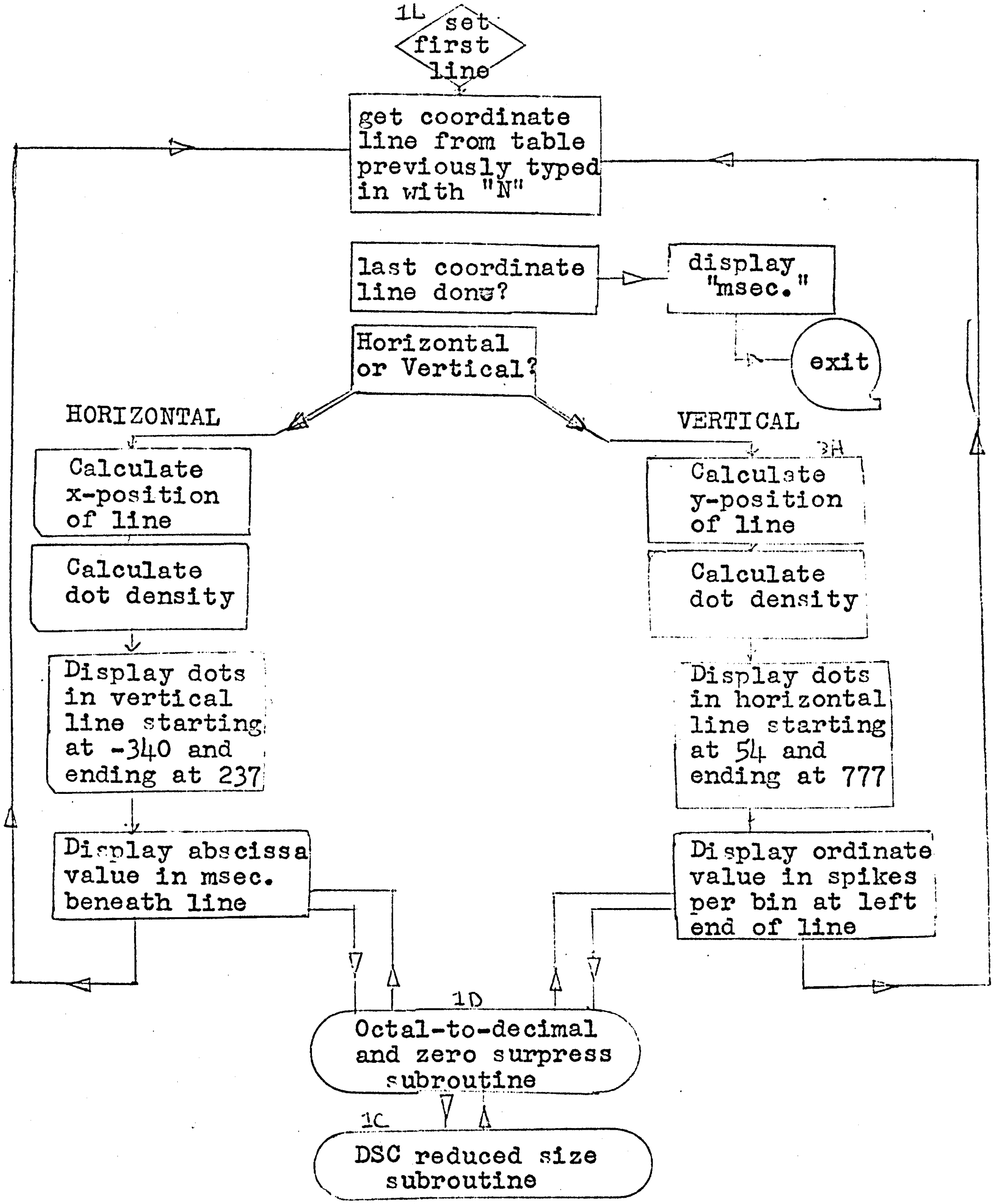
Flow Chart for HISTOGRAM



Flow Chart for HISTOGRAM Proper



Coordinate Lines Flow Chart



Summary of Operating Instructions
HISTOGRAM with Invariant Coordinates

knob 5 scales amplitude by $2^{\pm n}$
knob 7 sets bar width and thus horizontal scale
SNS 0 stalls for taking time exposures
SNS 1 up turns off legend
SNS 2 up turns off coordinate lines

"A" integrates bins 3001-3777
"B" sets bin width (type one octal digit)
"C" reads and and transfers control to Control Program (0/0)
"D" delays first bin displayed: type in 3 digit msec. delay
"M" jumps to LINOTYPE to type in legend; get back after 3 EOLs.
"N" type in coordinate line (4 digits)

first digit sets for vertical or horizontal and sets dot density:

number	line	dot spacing
0	vertical	not allowed
1	"	20
2	"	10
3	"	4
4	horizontal	40
5	"	20
6	"	10
7	"	4

next three digits are location of line in milliseconds or spikes/bin (type in decimal)

a vertical line of minimum density at zero spikes/bin is not allowed since this would come out 0000, which serves to terminate displaying of coordinate lines.

"N" followed by "del" deletes last coordinate line.

"T" reads two blocks from tape (type three digit octal BN) unit 0.
BN will go into Q6; BN+1 into Q7.

Data is kept in quarters 6 and 7.

HISTOGRAM with Invariant Coordinates (Mark III)

W. H. Calvin
University of Washington
Department of Physiology & Biophysics

```
origin 14
    3X-1
    NOP
    RDG
    3010          location of program on tape
#5K KST
    JMP 2G
    KBDi
    SAEi
    24            A for integrate
    JMP p+2
    JMP 2F
    SAEi
    26            C for control block
    JMP p+2
    JMP 3F
    SAEi
    47            T for tape read
    JMP p+16
    JMP 2K        3 digit octal keyboard subroutine
    ADAi
    6000          places data in Q6
    STA
    p+5
    ADAi
    1001
    STC p+4       reads BN+1 into Q7
    RDC

    RDC

    JMP 2G        out to histogram proper
    SAEi
    40            M for LINOTYPE subroutine
    JMP 2X
    JMP 1A        LINOTYPE
#2F SETi2         Integrate starting at 3001
    3000
    LDAi2
    ADMi2             Adm 2
    XSK 2         ending at 3777
    JMP p-2
    JMP 2G
#3F SETi17        Out to Control Program
    0
    JMP 16        Read in control program at 0/0
#2X SAEi
    41            N for coordinate line
    JMP p+23
    KBDi
    SAEi
    13            delete?
```

```
     JMP p+10
     CLR
     STA 14        delete last line
     LDAi
     -1
     ADM
     14            set back counter
     JMP p+7
     ROL 11        shift first digit into left position
     STC p+3
     JMP 4K        three digit decimal keyboard subroutine
     ADAi

     STAi14
     JMP 2G
     SAEi
     25            B for bin width
     JMP p+13
     KBDi
     AZE
     JMP p+3
     LDAi          if type zero, set to one
     1
     STA
     1Y
     COM
     STC 4B
     JMP 2G
     SAEi
     27            D for delay in first bin displayed
     JMP 2G        change this if ever add additional letters to test
     JMP 4K        three digit decimal keyboard subroutine
     MULi
     1             for 1 msec bin width; change to 5 if use 200 umsec.
     COM
     STA                                     basic
     6F            notifies Coordinate Lines of/bin width if not 1
     COM
     ADAi
     2777
     STC 1S        location of first bin displayed
     JMP 2G
 #2G SAM 7         get bar width from knob 7
     ADAi
     200           make positive
     SCR 5
     ADAi
     2             range is 2-9 (change to 1 if eliminate bar spacing)
     STA
     p+6
     COM
```

```
     ADAi
     2000       makes actual bar width one less than specified,
     STC 3B       to allow for one space between bars.  Change
     LDAi         to 1777 if eliminate spacing.
                bar width
     MULi
#1Y             multiples of basic bin width
     MULi
     1          reciprocal of basic bin width (5 if 200 usec bins)
     STC 5H     tells Coordinate Lines where to locate, given msec.
     SAM 5      get vertical height scaling from knob 5
     APOi
     JMP p+6
     COM
     SCR 4        2^-n
     ADAi
     SCRi       manufactures shift instruction for later use
     JMP p+4
     SCR 4
     ADAi
     ROLi         2^+n
     STA
     2S         for use in Histogram proper
     STC 1G     for use in Coordinate Lines
     SNSi0      serves to halt computer after one complete
     JMP p-1      loop thru the program for uniform time exposures
#3G  SETi2
#1S  2777       first bin location -1
     SETi4
     1054       x-coordinate of first bin displayed
#4G  SETi5
#3B             bar width ~~nx~~
     SETi3
#4B  -1         bin width
     LDAi2      get bin
     XSKi3      count on bin width
     JMP p+2
     JMP 2S
     ADAi2      add suceeding bins (if any)
     XSKi3
     JMP p-2
#2S  SCRi n     or ROLi n (set earlier)
     LZEi
     JMP p+3
     ADAi
     1          round up
     ADAi
     -737         net effect is -337, but keeps bar height from
     APOi         exceeding screen height and thus going around
     CLR          again.  If bin contents are so large as to
     ADAi         exceed ~~3777~~, this obviously doesn't work.
     400                 4737
     STA
     4H
```

```
#5G XSKi4        x-coordinate of bar
    JMP p+2
    JMP 1F       last x.  Go to legend & coordinate line control.
    LDAi
#4H              y-coordinate
    DIS 4
    ADAi
    -1
    SAEi         fills in bar down to zero (-337)
    -340
    JMP p-5
    XSKi5        count on bar width
    JMP 5G
    XSKi4        [illegible] Change to XSK 4 if
    JMP 4G        you wish to eliminate spacing between bars.
    JMP 1F
[illegible]
                 Legend & Coordinate Line Control
#1F SNSi1        sense switch 1 up will turn off legend
    JMP 1L       [illegible]
    SETi 10
    1700         sets number of times legend is displayed, so that
    XSKi10          it will appear the desired brightness in time
    JMP p+2         exposure photographs.
    JMP p+3
    JMP 1Q       jumps to legend display subroutine
    JMP p-4      return from "       "       "
#1L SNSi2        sense switch 2 up will turn off coordinate lines.
    JMP 5K       back to start
    SETi13
    1770         sets number of times that Coordinate Lines are
                     displayed relative to Histogram proper.
    XSKi13
    JMP 1H       jumps to Coordinate Lines.  Return is to p-1 (XSKi13).
    JMP 5K       back to start.
```

Table of coordinate lines begins at 3X.
Each coordinate line is specified by one word. The highest three bits specify the line direction and the dot density. The low nine bits specify the spikes per bin or the milliseconds.

```
#1H SETi10
    3X-1        table of coordinate lines
    LDAi 10
    AZEi        zero denotes end of table.
    7G+21       Go to print "msec."
    APO
    JMP 3H      Indicates horizontal line (4,5,6,7)
#2H SCR 11      get highest three bits (will be 0,1,2,3)
    ADAi           VERTICAL LINE (msec)
    SCR         manufacture shift instruction
    STC p+3
    LDAi
    40          widest dot spacing
    SCR
    STC 6H
    LDA 10
    BCLi        get lowest nine bits
    7000
    ADAi
#6F             add delay in first bin displayed
    MULi
#5H 1           reciprocal of basic bin width (5 if 200 usec. bins)
    ADAi
    54          x-coordinate of first bin displayed
    STC 4       x-coordinate of vertical line
    LDAi
    -340        start veryical line at -340
    DIS 4
~~#6H~~ ADAi
#6H             add dot spacing (40, 20, 10, 4)
    ADAi
    -237
    APOi
    JMP p+4     stop line at 237
    ADAi
    237
    JMP p-11
    LDAi
    -4
    ADM
    4           subtract 4 from x-coordinate of line so as to
    SETi11       place abscissa value under line.
    -377
    LDA 10      IR 11 holds y-coordinate, IR 4 holds x-coordinate
    BCLi          and ACC holds octal value of abscissa for
    7000          DSC subroutine., with octal-decimal conversion.
    JMP 1D
    JMP 1H+2
```

```
#3H BCLi
    4777        gets dot density (0,1,2,3)
    SCR 11
    ADAi
    SCR         manufactures shift instruction
    STC p+3
    LDAi
    40          widest dot spacing
    SCR
    STC 7G
    LDA 10
    BCLi
    7000
#1G SCRi n      set earlier (multiplies by 2^±n)
    ADAi
    -340        y zero at -340
    STC p+6
    CLR         clears link bit
    LDAi
    54          starts horizontal line at x= 54_8
    STC 4
    LDAi

    DIS 4
    LDA
    4
    ADAi
#7G             add dot spacing (40,20,10,4)
    ADAi
    -777        stop at end of screen
    APOi
    JMP p+4
    ADAi
    777
    JMP p-16
    SETi4       ordinate label
    0           set x-coordinate of first number to zero
    SET 11
    1G+11       y-coordinate of horizontal line & of ordinate label
    LDA 10
    BCLi
    7000
    JMP 1D      to DSC subroutine with octal-to-decimal conversion
    JMP 1H+2
    SET i4      "msec." label
    700
    SETi11
    -377
    SETi10
    5Q-1
    SETi6
    -5
```

```
    LDAi 10
    JMP 1C      DSC subroutine.
    LDAi 10
    JMP 1C
    LDAi
    3           space of 3 between ~~numerals~~ letters.
    ADM
    4
    XSKi 6      count on number of letters
    JMP p-11
    JMP 1L 4
#5Q 3077
    7730        M
    5121
    4651        S
    5177
    4151        E
    4136
    2241        C
    0001
    0000        •
```

Octal-to-decimalling, zero-surpressing
Display Decimal Numeral of reduced size
Subroutine

Displays decimal numbers from 0 to 511 with leading zero supress.
X-coordinate must be in 4, y in 11, number in ACC.

```
#1D STC 2D
    ADD 0
    STC 7D
    SETi5
    0            set hundreds register to zero
    LDAi
#2D
    COM
    ADAi
    144
    APOi
    JMP p+3
    XSKi5        tallies hundreds register
    JMP p-5
    ADAi
    -144
    SETi6
    0            set tens register to zero
    ADAi
    12
    APOi
    JMP p+3
    XSKi6        tallies tens register
    JMP p-5
    ADAi
    -12
    COM
    STC 7        set ones register
    LDA
    5
    AZEi
    JMP 3D       if hundreds is zero, do not display a zero
    ROL 1        x2
    ADAi
    6D           first word in list of code words
    STA
    p+5
    ADAi
    1
    STC p+5
    LDA
                 first half of character
    JMP 1C       DSC subroutine
    LDA
                 second half off character
    JMP 1C       DSC subroutine
    JMP p+5
```

page
Display Decimal
Numeral of
Reduced Size

```
#3D LDA
    6           tens
    AZEi
    JMP 4D      zero surpress
    LDAi
    3
    ADM
    4           spacing of 3 between numerals
    LDA
    6
    ROL 1       x2
    ADAi
    6D          first word in list of code words
    STA
    p+5
    ADAi
    1
    STC p+5
    LDA

    JMP 1C      DSC subroutine
    LDA

    JMP 1C
    LDAi
    3
    ADM
    4           spacing of 3 between numerals
#4D LDA
    7           do not zero surpress ones
    ROL 1        x2
    ADAi
    6D
    STA
    p+5
    ADAi
    1
    STC p+5
    LDA

    JMP 1C
    LDA

    JMP 1C
#7D JMP         return to p+1 of originating program
#6D 4136
    3641        0
    7721
    0001        1
    4321
    3145        2
```

page

Display Decimal
Numeral of
Reduced Size

```
4122
2651       3
2414
0477       4
5121
4651       5
1506
4225       6
4443
6050       7
5126
2651       8
5121
3651       9
```

Reduced Size DSC Subroutine
W. H. Calvin

y-coordinate in IR 11 (left unchanged)
x-coordinate in IR 4 (left stepped by 6)
Code word in ACC

```
#1C STC 2C
    ADD 0
    STC 7C
    SETi1
    -2
#3C SETi2
    -6
    LDAi
#2C                code word
#4C BCLi
    7776           save lowest bit only
    AZEi
    JMPmp+4        do not display dot if bit is zero
    LDA
    11             y-coordinate
    DIS 4          x-coordinate
    LDAi
    3              (change this to change character size)
    ADM
    11             step y by 3
    LDA
    2C
    SCR 1
    STA
    2C             shift code word right one
    XSKi2          count on number of dots in vertical line
    JMP 4C
    LDAi
    3              (change this to change character size)
    ADM
    4              step x by 3
    LDAi
    -22            reset y (6 times dot spacing)
    ADM
    11
    XSKi1          do vertical line twice
    JMP 3C
#7C JMP            out to p+1 of originating program
```

Three Digit Decimal-to-Octal Keyboard Subroutine page
W. H. Calvin

```
#4K LDA
    0
    STC p+14
    STC p+10    zero count
    SETi2
    -3
    SETi1
    p+7
    KBDi
    MULi1
    ADMi

    XSKi2
    JMP p-5
    JMP         out to p+1 of originating program
    144
    12
    1
```

Three Digit Octal Keyboard Subroutine

W. H. Calvin

```
#2K KBDi        hundreds
    ROL 3
    STAi

    KBDi        tens
    ADD p-2
    ROL 3
    STC p-4
    KBDi        ones
    ADD p-6
    JMP 0       out to p+1 of originating program
```

LINOTYPE
for Histogram with Invariant Coordinates
W. H. Calvin

```
#1A SETi7
    2A-1          list of locations of first character in each line
    SETi10
    -3            three lines
    LDAi7         get location of first character in line
    STC 2
#3A KSTi
    JMP p+3
    JMP 1Q        Legend display subroutine
    JMP 3A
    KBDi
    SAEi
    13            delete?
    JMP p+15
    LDAi
    -2
    ADM
    2
    CLR
    STAi2
    STAi2         delete last character
    LDAi
    -2
    ADM
    2             backspace
    JMP 3A
    SAEi
    12            EOL?
    JMP p+4
    XSKi10        count on lines
    JMP 1A+4
    JMP 5K        jump out
    ROL 1         x2
    ADAi
    7A            first location in list of character code words
    STA
    p+5
    ADAi
    1
    STC p+5
    LDA

    STAi2
    LDA

    STAi2
    JMP 1Q
    JMP 3A
```

```
#2A 4X-1
    5X-2
    6X-2
#7A           list of code words for keyboard characters

#7A 4136
    3641       0
    7721
    0001       1
    4321
    3145       2
    4122
    2651       3
    2414
    0477       4
    5171
    4651       5
    1506
    4225       6
    4443
    6050       7
    5126
    2651       8       10
    5121
    3651       9       11
    0
    0          EOL     12
    0
    0          del     13
    0
    0          space   14
    1212
    1212       =       15
    0
    0100       period (u p)   16
    0400
    0404       minus   17
    0400
    0416       plus    20
    0000
    0077       / (origin)   21
    7777
    7777       ▯  (brackets)   22
    0
    0          case    23
    4437
    3744       A       24
    5177
    3651       B       25
    4136
    2241       C       26
    4177
    3641       D       27
    5177
    4151       E       30
```

5077		
4050	F	31
4136		
2645	G	32
1077		
7710	H	33
4100		
4177	I	34
0102		
7601	J	35
1477		
4122	K	36
0177		
0301	L	37
3077		
7730	M	40
3077		
7706	N	41
4136		
3641	O	42
4477		
3044	P	43
4536		
3743	Q	44
4477		
3146	R	45
5121		
4651	S	46
4040		
4077	T	47
0176		
7601	U	50
0176		
7402	V	51
0677		
7701	W	52
1463		
6314	X	53
0770		
7004	Y	54
4543		
6151	Z	55

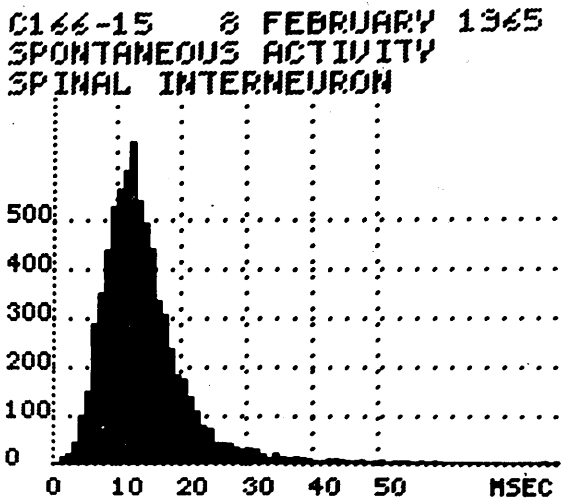
Legend
HISTOGRAM with Invariant Coordinates page

Displays three lines at top of screen (240-377) using DSC instruction. If wish smaller characters, can use JMP 1C in place of DSCi3, changing x-coordinate from IR 1 to 4, and y from 3 to 11.

```
#1Q LDA
    0
    STC 7Q
    SETi5
    -3          sets for three lines
    SETi6
    340         y-coordinate of first line at 340
    SETi3
    4X-1        beginning of table of characters
    SETi1
    1777        initial x-coordinate is zero -1
#4Q SETi4
    1746        25_10 characters per line
#2Q LDA
    6
    DSCi3
    DSCi3
    XSKi4       count on number of characters
    JMP p+2
    JMP 3Q
    LDAi
    4
    ADM
    1           space between characters
    JMP 2Q
 #3QXSKi5       Count on line number
    JMP p+2
#7Q JMP         jump back to p+1 of originating program
    LDAi
    -40         spacing between lines
    ADM
    6
    JMP 4Q-2
#4X
    5X is at 4X+62
    6X is at 4X+144
    3X is at 4X+226   empty spaces inbetween
```

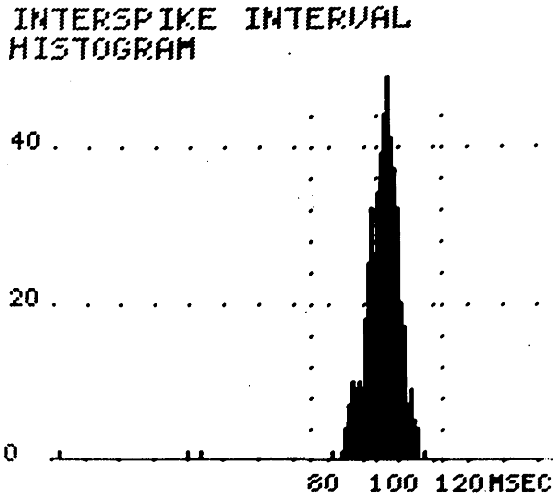
10-60

Histogram with Coordinate Lines Invariant to Changes in Scale and Origin

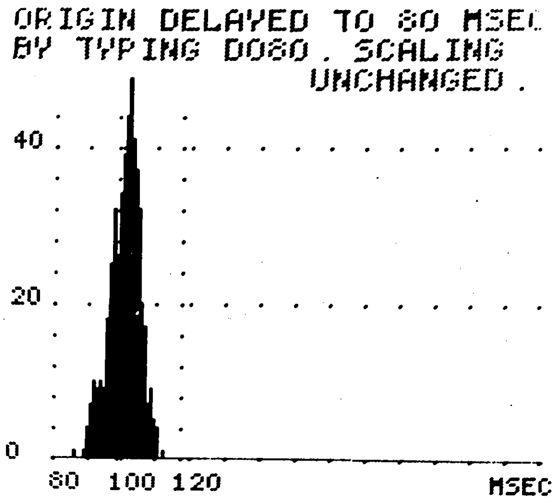


William H. Calvin
Department of Physiology & Biophysics
University of Washington
School of Medicine
Seattle, Washington 98105

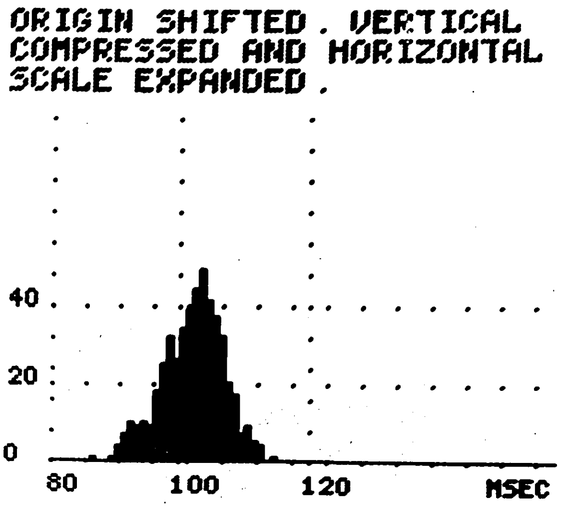
10 March 1965



ISIs are clustered about 100 msec.



Shift origin to 80 msec. Coordinates shift also.



Coordinate shift accordingly when change scale.

FLUCTUATIONS OF THE INTERSPIKE INTERVAL OF THE CRAYFISH STRETCH RECEPTOR

By David R. Firth
Department of Physiology
and Biophysics
University of Washington
Seattle, Washington

INTRODUCTION

The crayfish stretch receptor emits a very regular series of nerve impulses at a rate dependent on the degree of stretch. It was thought that fluctuations of the interval between impulses (interspike interval τ) would reflect fluctuations in the membrane potential and in the excitability mechanism. With these basic sources of membrane noise in mind a study was made of the standard deviation (s) of the interspike intervals as a function of mean interval ($\bar{\tau}$).

RECORDING OF DATA

For reasons mainly connected with the evolution of the experiment from simple beginnings, the interspike intervals were recorded on moving film instead of being timed directly by the LINC computer. The particular display used on film is somewhat unusual and is made necessary by the small size of the fluctuaton (e.g., when $\bar{\tau}$ = 20 ms, $S \sim 0.15$ ms. This contrasts greatly with the frog muscle spindle, where the fluctuation at τ = 100 ms is about ± 25 ms.

(See Hagiwara 1954, Buller, et al. 1953). The film moves continuously but quite slowly, and the oscilloscope beam rapidly sweeps perpendicular to the direction of film motion and once for each impulse. The sweep is triggered at the end of a constant time delay, which is itself initiated by each impulse. By adjusting the length of the time delay and the sweep it is possible to arrange that every nerve impulse arrives during the sweep, and on arrival it triggers the beam brightener. The result is a band of short blips at right angles to the length of the film and lying anywhere across its width. (See Fig. 1.)

USEFULNESS OF DISPLAY OF DATA

By means of this device the two dimensions of the film both represent time scales (though of different size), the small fluctuation is "magnified," a vivid display results, and film is used economically. It is clear that the LINC scope display could be used to achieve the same effect, thereby giving the "eye-brain" computer a chance to do what it best can, to recognize patterns and unusual effects in the sequence of intervals.

TRANSFER OF DATA INTO LINC

The positions of the blips on film represent the interspike intervals whose values are entered into LINC in a semiautomatic manner. A film projector was modified in such a way that, by moving a lever which turns a mirror and slides a potentiometer, a D.C. voltage proportional to the height of a blip is produced. By A-D conversion of this voltage the intervals are stored in LINC and on tape in digital form. About one or two intervals can be read each second with this device.

THEORY OF ANALYSIS TO REMOVE DRIFTS

If the standard deviation of the intervals is calculated in the usual way from the sequence (τ_1, τ_2, . . .) a serious error arises owing to the fact that the "true" mean interval at any time is subject to drifts, partly due to adaptation, but also due to less regular effects. The drifts are often much larger than the fluctuation from interval to interval so that the fluctuations must be calculated about some kind of drifting local mean. We have foreknowledge as well as afterknowledge and must clearly give distant intervals less weight. As a result of mathematical trials using various local means it appeared that the use of differences was a powerful way to remove the effect of drifts.

Thus if τ_r is the r'th interval in a sequence we define

first difference $\Delta_1 = \tau_r - \tau_{r+1}$

second difference $\Delta_2 = \tau_{r-1} - 2\tau_r + \tau_{r+1}$

third difference $\Delta_3 = \tau_{r-2} - 3\tau_{r-1} + 3\tau_{r+1} - \tau_{r+2}$

If s is the true standard deviation of the τ's, it is a very good approximation, even in the presence of drifts, to write:-

$s^2 = s_1^2 \equiv \overline{\Delta_1^2}/2$ where s_1, s_2, s_3 are the estimates

or $s^2 = s_2^2 \equiv \Delta_2^2/6$ of s.d. by means of the various

or $s^2 = s_3^2 \equiv \Delta_3^2/20$ differences.

In the absence of drifts or correlations between intervals

$$s_1^2 = s_2^2 = s_3^2 = s^2 = \overline{\tau^2} - \overline{\tau}^2$$

LINC PROGRAM

A program was written by Mr. Kurt Beam, (an undergraduate summer student), to deal with the input, storage, calculation, and output of the necessary quantities.

The input to LINC consisted of the sequence of interspike intervals of a given run as described earlier, along with calibrations of the time axes of the film. The intervals were stored on tape in blocks of 255.

The following quantities (defined above) were calculated:

$$\overline{\tau}, \overline{\tau^2}, s_1^2, s_2^2, s_3^2 .$$

The output form was somewhat inconvenient, and consisted of a decimal number multiplied by a power of 2. A histogram of Δ_2 was read off sequentially in binary form from the accumulator.

CRITIQUE OF DATA HANDLING

The system described above, for data acquisition, analysis and display appears rather makeshift and inefficient. The real reason for this is that the use of LINC was grafted onto an earlier system and the whole experiement was intended to be a preliminary and exploratory one. In addition there was a purely personal time factor; it was more important in this case to achieve a limited aim quickly than to do a fuller experiment in a more efficient way. The amount of data to be handled was not so large that time would have been saved by redesigning the system. Another reason for this mixture of old (photographic) and new (computer) systems was the importance of having the visual display, described earlier, in order to increase the chance of finding unexpected effects. The idea of using LINC to achieve a similar display did not occur until later.

A valid (but post hoc) reason for using film may actually lie with the smallness of the interval fluctuations, (e.g., s $\sim$ 0.15 ms at $\bar{\tau}$ = 20 ms) for in order to avoid adding appreciable measuring error to s, the graininess of the [illegible] measurer should be considerably less than the smallest values of s; this may pose complications owing to the cycling time.

RESULTS

Some of the results were presented at the Biophysical Society meeting (Firth 1965) and consist essentially of the curve relating s and τ, shown below in Figs. 2 and 3. The main features of the graph are its upward curvature, non-zero slope at the origin, and the size of the effect as a whole.

DISCUSSION

It is believed that a simple model consisting of a constant source of noise voltage (independent of the mean interval) superposed on the generator potential can explain the shape of the curve. If the generator potential were to rise linearly from start to firing level a linear s-τ curve would result. However, intracellular recordings (Kuffler and Eyaguirre 1955) indicate that, at larger τ values especially, the generator potential rises in a very non-linear fashion, approaching the firing level much more slowly than if it had risen linearly. This allows a given fluctuation in voltage to cause a larger fluctuation in time of firing than it would have done on a linear generator potential.

A source of noise voltage is at hand in the Johnson noise of the membrane resistance. Calculations show that in an infinite axon of 10μ diameter having typical membrane constants, the fluctuations in voltage would be large enough to explain the observed fluctuations of interval. In fact the crayfish axon is about 20μ diameter, the cell is somewhat larger and the dendrites are smaller. It appears possible, therefore, (~~tough~~ though the calculation is rather crude) that another source of noise (such as spontaneous release of inhibitory packets) is not needed to explain the fluctuations of interval. Intracellular recording of the generator potential is necessary to pursue the question further. This certainly offers a good opportunity for the use of LINC in analyzing the fluctuations of the generator potential and of the firing level.

REFERENCES

Hagiwara, Jap. J. Physiol. 4:234, 1954.

Buller, et al. J. Physiol. 122:409, 1953.

Firth, Biophys. J. Abstracts, Annual Meeting, 1965.

Kuffler and Eyaguirre, J. Gen. Physiol. 39:102, 1955.

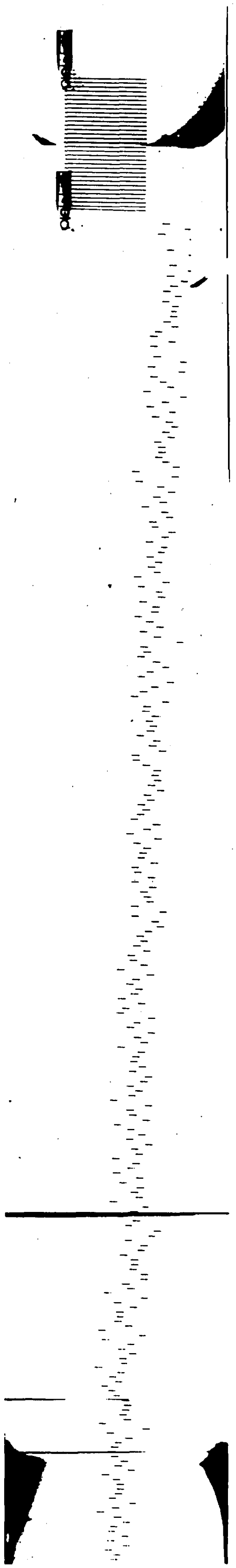


Fig. 1

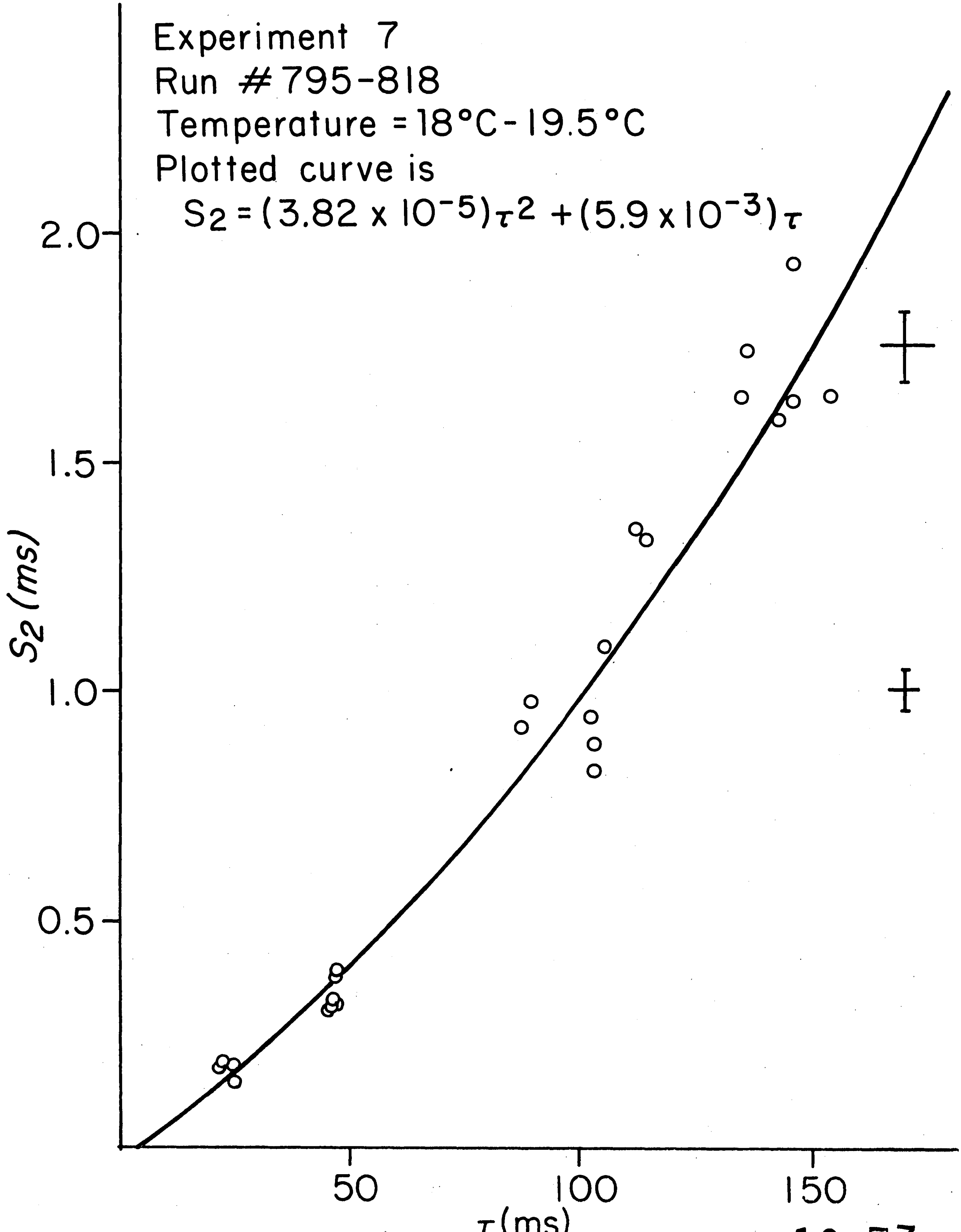
Experiment 7
Run # 795-818
Temperature = 18°C - 19.5°C
Plotted curve is
$S_2 = (3.82 \times 10^{-5})\tau^2 + (5.9 \times 10^{-3})\tau$
2.0
1.5
1.0
0.5
S_2 (ms)
50
100
150
τ (ms)

Fig. 2.

10-73

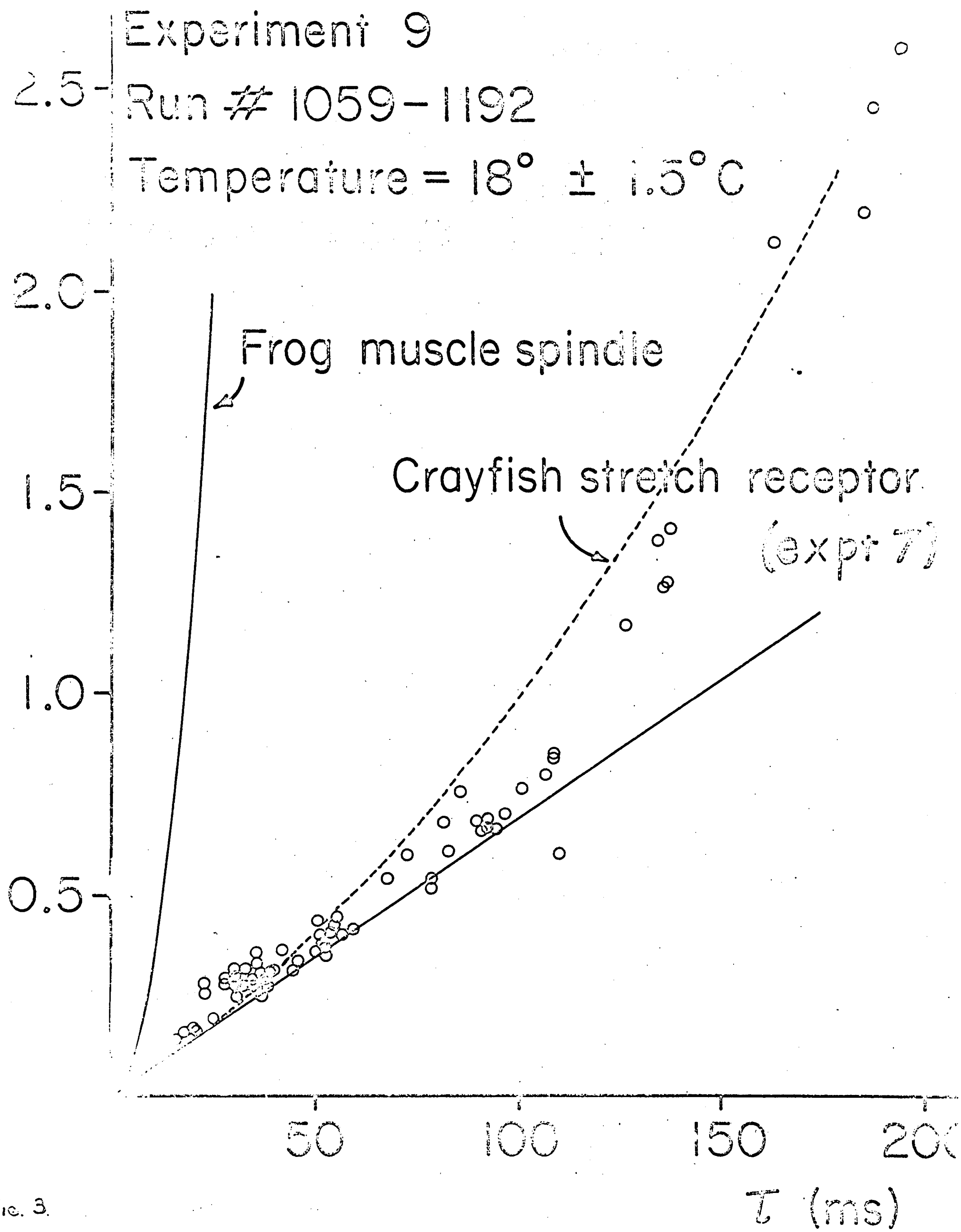

Experiment 9
Run # 1059-1192
Temperature = 18° ± 1.5°C
2.5
2.0
1.5
1.0
0.5
Frog muscle spindle
Crayfish stretch receptor (expt 7)
50
100
150
200
τ (ms)

FIG. 3.

UNIVERSITY OF WASHINGTON
SEATTLE, WASHINGTON 98105

School of Medicine
Department of Physiology
and Biophysics

CONDUCTION VELOCITY EXPERIMENTS - W. L. HARDY

LINC is being used by a graduate student, William L. Hardy, in an attempt to determine the quantitative effect of external sodium concentration on single fiber conduction velocity in a frog sciatic nerve.

The experimental set up consists of a sciatic nerve preparation held in a perfusion chamber, a set of three equidistant stimulating electrodes and a microelectrode with necessary preamplifier and electrometer input for recording the action potential intracellularly from a single fiber.

With a given perfusing fluid sodium concentration, LINC is programmed to sequentially select using the relays a stimulating electrode. The nerve is stimulated from an external pulse generator a designated number of times (set by the sense switches). The intracellular potential is sampled every 32 microseconds with a total of 150 points per record. The record is averaged with previous records at that stimulating position. Averaging is necessary because of the high noise level. The procedure is then repeated for each of the other two stimulating electrodes and the three average curves are stored on the magnetic tape under keyboard control. Using LINC to control the experiment through selection of site and time of the stimulus has made this experiment more possible. Because of the very small size of these myelinated nerve fibers, it is difficult to hold the microelectrode in the cell for an extended period of time.

It is planned to use the LINC to calculate the conduction velocity directly from the stored data. This information can then be stored with each record along with data on the sodium concentration in the perfusion solution. In addition it may be of use to record the time since the perfusing solution was changed. For this a real time clock would have to be installed.

Details of the sampling program will not be included as there is nothing very unusual about the method of changing stimulus positions with the relays, sampling at 32 microseconds between samples after a trigger pulse, storing and averaging data, displaying data, and storing on magnetic tape under keyboard control.

FINAL REPORT

Washington University School of Medicine

Department of Neurology

LINC COMPUTER EVALUATION - March 3, 1965

I. PROPOSED RESEARCH PROGRAM

(Supplemented in detail in subsequent section).

A. Background and Introductory Statement: Two years ago our group, then containing Sidney Goldring, M. D., now Professor of Neurosurgery at Pittsburgh, was invited to join a program to evaluate the LINC Computer. In the summer of 1963 Dr. Goldring and our electronics engineer, Mr. Lloyd Simpson, participated in the MIT program and assembled the LINC computer assigned to us there. For the several subsequent fall months we worked with the computer on test runs establishing an averaging program and modifying our own electronic equipment to its needs. Commencing in winter 1964 we moved to the neurosurgical operating room in Barnes Hospital and there did a series of studies upon the potential field of the somatic evoked response. Whereafter we returned to the experimental laboratory to establish further details and re-work the program in studies upon experimental animals. Meanwhile, Dr. Robert Wurtz was starting a new program on aversive and approach problems in stimulation of the caudal diencephalic and rostral mesencephalic territory of the rat with implanted electrodes. In winter and spring 1964 Drs. H. G. Schwartz, Professor of Neurosurgery, Dr. William Coxe, Associate Professor, and Dr. Wurtz took the W. U. courses given on the LINC computer. In fall 1964 we added a second memory and Dr. Roy Wright took the LINC computer course. We now have two people working on the LINC who have constructed their own programs with the aid of the LINC computer group and are now engaged in active daily research using that instrument as a primary tool of the research.

B. Objectives and Program

It is our purpose to study the interaction of communities of neurons using whatever electrical signs of their activity are appropriate to the particular needs of an experimental design. The working group has had experience with ultraslow potentials, extra- and intracellular unit potentials, and evoked potentials of somatic auditory and visual types. We have excellent laboratory facilities, dark rooms, animal quarters, experimental operating rooms, and access to neurosurgical patients. The present shielded rooms which house the LINC were set up with the view in mind of doing implanted electrode studies upon the human thalamus with natural stimulation and percutaneous electrical studies. We also have a scheme in mind for programming the computer to differentiate between the electromyograms of muscle units recorded percutaneously which are indicative of neuromuscular disease as compared with normal muscle. If this proves feasible we will have established a practical clinical use for the LINC. Similarly, in our studies upon the human thalamus we hope to provide a program which will differentiate between spontaneous activity at various depths and positions with the ultimate view of being able to decide more effectively where to place lesions for alleviation of Parkisonian symptoms.

Dr. Sidney Goldring who commenced with us has moved to Pittsburgh and there applied for and received a LINC computer. Thus we already have one satellite laboratory in operation.

One of the most perplexing problems we have met is that of providing a sufficiency of skilled operators who are capable of adapting the LINC to their uses. We now have a schedule of training for those who will take future LINC courses and we hope gradually to make our laboratory 100% LINC competent.

Details of this generalized proposal are offered in the ensuing lengthy supplement which follows the headings outlined by the Evaluation Committee.

II. SUPPLEMENT

1. Past and Present Research.

a. Average evoked somatosensory responses. Averaged evoked responses in animals and man were studied utilizing three different averaging programs.

Program #1: The program causes a relay to activate a Grass stimulator once per second. Sampling is begun after a 7 msec. latency and continued during the entire interval between stimuli. During the first 150 milliseconds sampling frequency is rapid to capture the fast response components. Thereafter, a slower sampling frequency records the later and slower components. The response is displayed on two lines; the upper on showing the initial 150 milliseconds of response, and the lower line, the remainder (850 msecs.). A sense switch is used to either continue display or clear the memory in preparation for recording the next response. Stimulation and samplings are started by depressing the Start 20 button.

Program #2 is basically the same with these additional features. After the last response is added, the averaged response is automatically stored on tape and then, using a Sense switch, either the memory is cleared or responses stored on tape can be called back into memory and subtracted from each other. In addition, two vertical cursors which are under knob control are displayed and used for comparing response latencies. Also, two horizontal lines, also under knob control, are displayed as base lines from which response deflections can be measured.

The two programs described above were used predominantly in the experimental laboratory. For patients, a more elaborate program (written by Severo Ornstein) was used. It records and recalls data via keyboard request and displays a table of contents upon request. The program is capable of recording on one or two channels (traces) and at the end of a sampling run writes the data on tape. Records are labeled from the keyboard and once recorded may be recalled by label. One can record on one trace while holding a previous recording on the other - or record on both traces - or show post records on both.

The basic process consists of typing a label and then asking to record ("R") or display history ("D"). When labeling, the desired trace must always first be indicated by "U" (upper) or "L" (lower). A label consists of a station number (from 0 to 39 decimal) followed by an "S", a "W" or a "T" (to designate surface, white matter or transcortical recording) followed by a dash followed by a version (repeat) number (from 0 to 9 decimal).

After labeling, striking "R" will cause recording on any trace with a label not previously used. That is: (1) A trace holding a history display will not be affected. (2) A duplicate of an already used label will not be accepted for recording action. (3) An unlabeled trace will be unaffected. The upper trace accepts input via the SAM 10 line; the lower, via SAM 11.

Recording can be done with either of two sampling intervals controlled by sense switch 2. Up = 600 μsec/sample; down = 304 μsec/sample. This should never be changed in the middle of any sampling run. The initial dot on a trace after sampling indicates the sample frequency used on that run. A dot above the trace indicates 600 usec. Sampling, below indicates 304 μsec. sampling. If during recording it is desired to reset, hit the "X" key. This erases the data for trace(s) that were being recorded but leaves the label(s) so that a new "R" action can be taken when ready to go again.

At the end of each sampling run the data are recorded on tape for future reference. The first record is stored in block 6 and subsequent records are stored in subsequent blocks. A table of the labels for recorded traces is kept updated in block 1 (backup duplicates of this block are also in blocks 2 and 3). For redundant recording on unit 1, sense switch O should be left down. This should hold for a complete experiment. The data as well as the label table will then be redundantly recorded.

Each record consists of the sum of 50 sets of samples. Each sets consists of 256_{10} samples where sampling starts immediately following stimulation (via relay 0). Stimuli are approximately 1 second apart. Individual samples are 304 or 600 μsec. apart (depending on the setting of sense switch 2). To call for display of an earlier record, merely type the label on the desired trace and hit "D." Requests may be made on a single trace or both both traces. Requests for non-existent traces will be ignored.

At all times except during actual recording a pair of vertical display lines are accessible if sense switch 1 is put up. The X positions are controlled by knobs 0 and 1.

A maximum of 58_{10} records can be handled on any one tape. An ordered tabular display of the labels of all records entered on a tape so far is obtained by hitting the case key. This display is terminated by the EOL key.

To study a tape recorded on an earlier occasion read in block 1 rather than block 200 at the outset. (If for any reason block 1 will not read and check, blocks 2 and 3 are duplicates of 1.) Starting at 20 in the usual fashion gets the program started and then one can inspect the table of contents and call up histories (or add new records if desired) in the usual manner.

To make a copy of the tape for other experiments, copy blocks 0 to 5 inclusive and blocks 177 to 210 inclusive onto the corresponding blocks of the new tape. A tape may be reused after an experiment by rereading block 200, etc., but further recording will then write over old records.

The program consists of 4 separate manuscripts which are located at blocks 240, 250, 260, and 270 respectively. The tapes also contain LAP3 in the standard place.

b. Steady potential correlates of reinforcing intracranial stimulation. This experiment was designed to determine whether the steady potentials recorded from cerebral cortex with direct coupled amplifiers change consistently with electrical stimulation of two kinds of subcortical structures, one producing aversive behavior and the other approach behavior (self-stimulation). Initially rats are tested behaviorally over a period of several months to determine whether they will either (1) press a bar to turn on current through an implanted electrode (self-stimulation) or (2) press a bar to turn off a current pulsed at one per second through an implanted electrode (escape). Once the behavioral significance of the electrical current application has been established non-polarizable electrodes of a type developed in our laboratory are implanted and the changes in steady potentials following stimulation are recorded using a cortical surface electrode and an indifferent one on the nose. The potentials are amplified by a Grass model V polygraph with a chopper stabilized preamplifier. Using a "reverter" supplied by Grass the signal from the driver amplifier is converted to a signal of 3 volts above or below a zero-volt level. A 1-volt signal is obtained by attenuation of the amplifiers.

LINC both averages the responses and controls the reinforcing stimulation. The steady potential described above is averaged (added) before and after each stimulus over a course of 100 successive stimulations. Two programs are used. In one the stimulus is triggered by the computer and in the other by the rat. Where the computer triggers the stimulus a sample is taken for 1. 28 sec., the stimulus is given, and a sample is taken at 10 msec. intervals for another 1. 92 sec. and additionally for 5. 12 or 15. 36 succeeding seconds depending on the setting of the sense switches. Where the rat does the triggering, the stimulus is given immediately, and samples are taken over a 3. 84 second period. LINC is also sampling and storing between bar presses and when the bar is pressed LINC saves the last 1. 28 sec. before the bar press so that a sample is averaged both before and after the stimulus in this program, also.

In both programs, sampling is done continuously between bar presses and a weighted average is updated as these samples are taken. This weighted average reflects the constantly changing d.c. level arising from movement of the animal, drift in electrodes, the "reverter", or the sample amplifier of the computer. In each period where samples are taken for averaging the weighted average is subtracted from each sample so that what is averaged is a difference between the pre-sample period and the sample period. LINC is triggered after a bar press by a pulse-former on peripheral equipment — a Massey-Dickenson transistorized programming unit. The 60-cycle stimulation was initiated by a relay pulse from LINC and duration was controlled by the peripheral programmer. The equivalent of the programmer could have been built into the data terminal box and would have been had the programmer not already been available and compatible with LINC.

c. <u>Plastic properties of synapses</u>. A change in the functional connection between presynaptic terminal and postsynaptic neuron is assumed by most microphysiological theories of learning. One variation of such theories predicts that if the postsynaptic neuron fires after a presynaptic terminal has been active, the connection between the pre- and postsynaptic neuron is facilitated while other inactive synaptic junctions are unchanged. In the future this EPSP produced by the presynaptic activity would have a higher probability of discharging the postsynaptic neuron. The paradigm is much like operant conditioning in the intact animal and it seemed possible to interpose LINC in the pre- and postsynaptic system in a manner similar to its use in other operant conditioning situations. LINC has been programmed therefore to recognize a particular EPSP waveform and either record the frequency with which it is followed by a spike discharge or pass current through the postsynaptic neuron to produce a spike.

The neurons chosen for this experiment are in the visceral ganglion of the sea slug, Aplysia californica. They are of large size, and consequently are easy to record from intracellularly over several hours. Glass micropipettes are used to penetrate the cell and a bridge circuit permits current to be passed through the same electrode used for recording. Input to the sample amplifier of LINC is via an oscilloscope amplifier and is capacity coupled. Output from LINC is via an operate pulse which triggers a pulse generator which in turn passes current through the microelectrode.

LINC is programmed to recognize a particular EPSP and respond to this particular EPSP but not to any other using a program similar in general but not in detail to one written by Simon. The program is divided into two segments: a selection mode involving a selection of and EPSP as the standard for recognition, and a running mode in which EPSPs are compared to the standard and either current passed through the microelectrode or the frequency of spontaneous cell discharge after the EPSP determined.

In selecting the appropriate EPSP, samples are taken at 4 msec. intervals with a continuous one-line moving display of the most recent samples. Pressing a keyboard button stops the sampling and displays four lines of data with eight consecutive intensified points which may be moved along the lines under knob control. A standard EPSP is selected by a keyboard instruction which transfers the eight points which are intensified to a catalog and this catalog is then displayed. The catalog may contain up to six possible standard EPSPs and one of the six standards in the catalog is picked as <u>the</u> standard. The continuous display may then be restarted and each EPSP identified as equivalent to the standard is marked. The acceptable difference between standard and current sample is under right switch control. Once the threshold is decided upon, the current required for spike production is set by another subroutine and the experimental mode of the program is ready to be run. At any time a subroutine may be interrupted and any other subroutine called up by a meta command given via the keyboard.

The standard sample actually used for comparison is not the sample displayed but a normalized value which permits recognition of the shape of the EPSP independent of any slower baseline fluctuations on which it might be found. The normalized value is obtained by taking the mean of the last eight sample points, and subtracting this mean from each of the samples. As each new sample is taken, it is added to a list of the eight most recent samples, and the oldest sample is discarded. A new mean is found (by adjusting the old mean) and a normalized value for the last eight points is obtained by subtracting the new mean from each point (actually by adjusting the previous seven normalized points and adding one new normalized point). These eight normalized points are stored in a second list. Comparison is then made to the eight points of the normalized standard, the absolute difference between each point is taken, the differences summed, and the sum compared to the threshold value. If the value is less than an acceptable threshold, a recognition is made; if not, no recognition is made, another sample is taken and the comparison cycle is repeated.

When the experiment is started the frequency of spike discharge following the first 16 recognized EPSPs is recorded and current is then passed after the next 124 recognized EPSPs. The 16-EPSP control period is then repeated and the experimental (current passing) and control periods then alternate. Data output is currently in the form of a bar graph of the number of EPSPs followed by a spike (1 to 16) in successive control periods.

Programming of this experiment, particularly the normalizing procedure and cursor program has been greatly benefited from the aid and suggestions of Cox, Ornstein, Sandel and others at the Computer Center. The experiment and program have been in the process of refinement since mid-January, 1965.

2. Future Research

a. Electrophysiological Studies on the Cingulum and Cingulate Gyrus. Two projects are currently active. The first of these involves the use of averaging to trace the cingulum conduction spike anteriorly and posteriorly at conduction distances beyond which a conduction spike is clearly discernable with standard techniques. This kind of approach is most valuable in elucidating both the origin and distribution of the cingulum bundle as well as providing information about the conduction rate of its component fibers. These same experiments will provide data which will permit a much more detailed electrophysiological analysis of the cingulum conduction spike in its supracallosal portion than was possible in earlier computation by this investigator.

The second project is underway inasmuch as data taken for study of the cingulum conduction spike includes much useful for it. This project involves a study of the instantaneous electrical fields for given cross sectional positions in the cingulate gyrus. These electrical field patterns will be correlated with the cross sectional configuration of the gyrus in an attempt to elucidate the virtual generators and the sequence of events in the cortical evoked response which follows cingulum stimulation.

A program for sampling an evoked response in the cingulum bundle and cortex of the cingulate gyrus has been developed with assistance of Severo Ornstein. Two sampling rates, 32 and 512 μsec. are used. The number of samples at each rate and/or the rates are variable. A total of 777 octal samples are possible. This program permits averaging of 32 responses and includes a feature for saving the average on tape with appropriate run number and indication of regular or special mode of recording. A display subroutine permits observation of the last response sampled as well as the build up of the average. This display subroutine also includes sense or right and left switches selected amplitude controls, switch selected horizontal (time) gain controls, amplitude calibrations for both current and average traces, and vertical cursors. The latter are coupled to the traces in such a manner that both the time difference between any two points and the amplitude at a given point (with respect to the calibration voltage) are computed.

In summary, a program is in use which permits saving of an evoked potential with sufficient on-line data analysis for optimal direction of the experiment. Although designed for this particular purpose this program is readily adaptable to a wide range of similar experiments in neurophysiology.

A data analysis program utilizing the above data saving program and its display subroutine is partially complete. This includes features permitting keyboard selection of data on tape to be read in for comparison of the response from different recording positions (even from different experiments) with the option to add or subtract the two responses and display the result. A feature permitting selection of the potential at any given instant of time for each of a large number of responses is included and will permit use of the computer in plotting the instantaneous potential vs. recording electrode position either as a single or a family of curves. A special program using this data to plot the electrical field contour lines on the computer oscilloscope has been discussed with Dr. Cox. When it becomes available this program will markedly facilitate the correlation of the electrical fields with the anatomy of the generators involved as well as permitting studies of the changes in the fields with respect to time.

For stimulation use in made of the OPR i instructions, which provides a pulse sequence. Only the first pulse is used here.

This is amplified via an indicator light driver and used to trigger a Tektronix waveform generator-pulse generator combination. This pulses a Grass stimulator which delivers the stimulus to the preparation. In order to have the sampling timing uniform with respect to the stimulus a pulse which occurs at the same time as the stimulus is taken from the Grass stimulator and inverted with a Tektronix waveform generator and led to the computer via external level line (XL). This ends the computer pause state and sampling begins within a few microseconds.

The input to the data terminal box for sampling is derived from special vacuum tube amplifiers in use in this laboratory. These are fed by cathode follower amplifiers adjacent to the preparation.

b. Use of LINC in the Analysis of Steady Potential Changes in Chronically Implanted Animals. Two series of experiments are planned at this time.

Relation of steady potential shifts to unit responses during wakefulness and sleep. Work in the laboratory on cats with chronically implanted electrodes for d. c. recording indicates that steady potential shifts occur during wakefulness and a phase of sleep referred to as "deep sleep." Using LINC it will be possible to relate these spontaneously occurring changes quantitatively to unit responses using the following general approach. Two types of data will be compared – a d. c. shift sampled about 50 times per second and unit firing frequency. As the cat awakens or goes into deep sleep the unit firing frequency will be taken in a series of time periods each lasting five to 15 seconds. The length of each time period will be determined by the rate of the d. c. shift which will be sampled continuously. Thus when the shift has reached 100 μV, the first time period will end, at 200 μV the second time period will end, and so on. Over a number of periods of sleep the frequency histogram for each time period will be summed. The significant correlation will be between successive d. c. amplitude changes and successive changes of unit firing. The same program methods for elimination of drift used in the previous experiments on steady potential will again be used. Unit responses will be discriminated by amplitude.

Cortical d. c. potential changes during conditioning. The purpose of these experiments is to determine whether consistent d. c. potential changes occur in cortex during conditioning and if the shifts are similar to those arising spontaneously in the sleep – wakefulness cycle. Cats with chronically implanted electrodes for d. c. recording will be conditioned using a flashing light for CS and a shock for UCS. Potentials will be recorded over six areas of cortex including sigmoid, suprasylvian, and posterior lateral gyrii before, during and after the onset of the CS-UCS period using the Grass polygraph amplifiers and "reverters." CS, UCS, sampling periods, etc., will be controlled by the program and a data terminal box. Samples will be averaged over blocks of ten trials. Since sampling will be of the order of 50 to 100 samples per second there would be enough time to sample all six available polygraph channels on each behavioral trial. A rough program has been drawn up as a class exercise at the Computer Center.

c. Plasticity of single neurons. This work will be continued with further effort concentrating first on such variables as initial probablity of an EPSP firing a spike or ratio of experimental to control periods and later on the changes associated with different EPSPs and changes in several neurons. The general format of the present LINC program may be simply modified to encompass many of these additional experiments.

3. Training Program

a. (S. G.) The best feature was the month course at Cambridge. This was the best period of intruction I have ever experienced. The remainder of the program was essentially, learning by doing. The fact that we were able to ask for and receive help from Clark, Cox and associates was instructive and valuable in carrying out our own program.

b. (R. W.) I did not participate in the month at MIT but the course I had with Dr. Cox proved adequate to understand and begin programming of LINC; it did not intend to nor did it provide a basis for repair. Direct contact with the group in St. Louis has been a unique and very profitable advantage.

Other people in the Department desiring to use LINC have also taken the Computer Center course(Drs. Kelly and Wright) but an individual must wait until the course is taught; teaching only one person at a time is very time-consuming. Consequently, I cannot pass up the opportunity to suggest that a program could be constructed using the standard question and answer routine on the oscilloscope so that LINC could instruct on the use of LINC – a very expensive teaching machine. Coupled with the console function, binary numbers, and order code write-up and some practice programs, the rudiments could be taught on an individual basis in any laboratory at any time. Once the individual started to program, instruction could easily be continued by questions to the more LINC-knowledgeable people in the laboratory.

4. Computer Performance.

a. Maintenance. Maintenance of LINC in our situation was made at practically no loss of time, thanks to the Computer Center and its personnel, in particular, Drs. Jerome Cox and Severo Ornstein. Overall the computer required much less maintenance and upkeep than we anticipated as our log will verify. Defects were usually of a minor nature, easily repaired once located.

Our power supply would kick out the +18 volt breaker but this occurred over a short period of time and eventually corrected itself. No cause was determined. If the computer was allowed to remain overnight in an air-conditioned area the L A P 3 program would not perform until the computer had been turned on for at least 30 minutes. This was solved by simply turning the air-conditioner off at night. The tape units gave us trouble failing to perform tape instructions occasionally. This was determined to be a bad diode located on the tape unit. The computer failed to put data into memory on one occasion and this was found to be a defective transistor in the E P 105 series write card. This same transistor had previously been replaced at MIT during the assembly phase of the course there in August, 1963. During the installation of second memory a transistor was again found to be defective in a series read card. No other difficulties of consequence were experienced. It is our opinion that the computer has performed exceptionally well for us insofar as maintenance is concerned.

b. Performance in experiments. At the time we received LINC, no one in the laboratory had had any previous experience or contact with computers or digital computer techniques. Consequently, a substantial amount of our time has been spent in developing our skills in the use of the machine and our appreciation of its capabilities rather than in the development of more extensive input-output connections. In addition, our LINC is physically situated so that it may be conveniently used on-line and all of our experiments have been done on-line with LINC analyzing the data and controlling the experiment. For our purposes, the performance of LINC has been superlative and we would not suggest any basic modifications. The comments that follow we feel are relatively minor and arise when comparisons are made to an "ideal."

The order code has been adequate and we have actually used or seen the need for all the instructions with the possible exception of RTA. An instruction that would skip if the contents of a beta register equal Y would frequently save an extra counter or the time required for an LDA and an SAE and would be more useful than RTA if instructions could be so juggled. For our purposes, word length has been entirely adequate, and with the addition of second memory core memory size has been satisfactory.

All of our recording has been analog data using the sample amplifiers, either capacity-coupling from oscilloscope amplifiers or direct coupling from the postamplifier of a Grass polygraph (after demodulation of the choppered preamplifier stage). The sample amplifiers have been found to be adequate. Any drift in them has been small compared to that of the preparation and electrodes when d. c. recording was used and has been taken out in programming. Tiggering LINC from other equipment by the use of an external level has been no problem and is an essential feature of the machine for our purposes.

To control other equipment from LINC, the pulses produced by the relay or operate instructions have been preferable to the use of relays. The one exception is the control of alternating current. We now trigger from LINC a variety of laboratory instruments with various triggering requirements. Grass stimulator, +50 volts; Tektronix pulse and waveform generators, +5 volts; stimulus isolation units, ±5 volts, or +20 volts; Massey-Dickinson transistorized programming apparatus and other behavioral apparatus, ±12-28 volts. Indicator-drivers and Tektronix pulse generators have been used to convert the 3-volt pulse to the required amplitude and we anticipate doing all such triggering using only D. E. C. cards with reduction in cost (from tied up pulse generators). Since this problem appears completely and inexpensively solvable by the use of appropriate D.E.C. cards, it would have been useful initially to have a list of cards, including transistor substitutions, that could be used to produce low current pulses at various positive and negative voltages. It would also be ideal to have a higher voltage source available in the data terminal box (about ± 30-50 volts) to avoid the use of batteries or an external supply.

The data terminal box has been admirably suited for our purposes; the standard box would be more useful if only the sample amplifier connections were made and other terminals installed but left unconnected.

Because LINC has been compatible with existing equipment, we have added very little input-output hardware. The teletype has proved valuable for hard copy of programs and numerical data. The oscilloscope has been adequate to date for displays of analog data. We have not had available an adequate tape recorder, but hope to acquire one in the near future.

One of the greatest advantages of LINC, its flexibility, also is related to what has been its greatest disadvantage, the time required for programming. While this has decreased with proficiency, the same proficiency has also permitted construction of more elaborate programs, and the time consumed for a program beginning to utilize LINC capabilities has been substantial. Time spent programming, of course, relates in part to the clarity of the individual's experimental ideas and methods and his ability to translate the program format, and no hard- or software seems likely to change this. The software tools now available for constructing and storing programs, control of binary programs, and standard routines such as manuscript print-out are necessary to use of the LINC so that, of course, we never program without these tools. At this time, these assembly devices seem to be adequate. What we have felt a need for is not more trowels but more brikcs, that is, program subroutines. These have accumulated in the laboratory as more programs have been written although many core parts of programs are not of much transfer value. There remains, however, a category of routines which are used or could be used if they did not involve construction each time, for example, teletype print, octal and decimal conversion, display of numbers, and storage routines. All can be constructed and many of them have been in the laboratory, but it would be ideal to have them available on tape and debugged. As important as having the program would be the format: preferably a print-out of the manuscript, anotated so that it could be understood (which in turn greatly helps the LINC user's programming abilities) with the lines for the various parameters labeled.

Even with this aid, programming time will remain substantial. This limits the number of experiments in the laboratory which are programmed for LINC but it also motivates discrimination between experiments which would greatly benefit from LINC and those which would benefit but in relatively trivial ways. In net, where LINC is deemed appropriate the results produced more than compensate for the programming time. The programming simply is an added experimental "cost" as are the other instrumentations and techniques.

Evaluation of the cost of LINC is difficult since it has been used on experiments which would not have been done without it and so cannot be assigned a dollar value. Because of its versatility, the cost need not be assigned to one specific experiment but to a variety of experiments even over a number of years. In this light, the estimated commercial price seems to us to make LINC a reasonably priced laboratory instrument. This is particularly true in our case since we have been able to use it without large investment in other hardware.

In net, we have found LINC to be a remarkable machine and a powerful laboratory tool. It has been reliable, has connected to our laboratory equipment with little difficulty or expense, and has produced any analysis desired with sufficient programming effort.

If there is any inadequacy, it is in software rather than hardware. As our ability to use LINC has increased, we have been even more impressed with its flexibility and range of application and this has, in turn, suggested experiments not possible without its capabilities.

5. Log Book.

A Xerox copy of our log book is enclosed.

6. Bibliography.

a. Papers. (Copies are enclosed)

(1) Goldring, S., Kelly, D. L., and O'Leary, J. L.: Somatosensory Cortex of Man as Revealed by Computer Processing of Peripherally Evoked Cortical Potentials, Transactions of the 89th Annual Meeting of the American Neurological Association, 1964, pp. 108-111, Springer Publ. Co., Inc. New York City.

(2) Kelly, D.L. Jr., Goldring, S., and O'Leary, J.L.: Average Evoked Somatosensory Responses from Exposed Cortex of Man, Archives of Neurology, 1965. (In press)

(3) Wurtz, R.H.: Steady Potential Correlates of Reinforcing Intracranial Stimulation (In preparation)

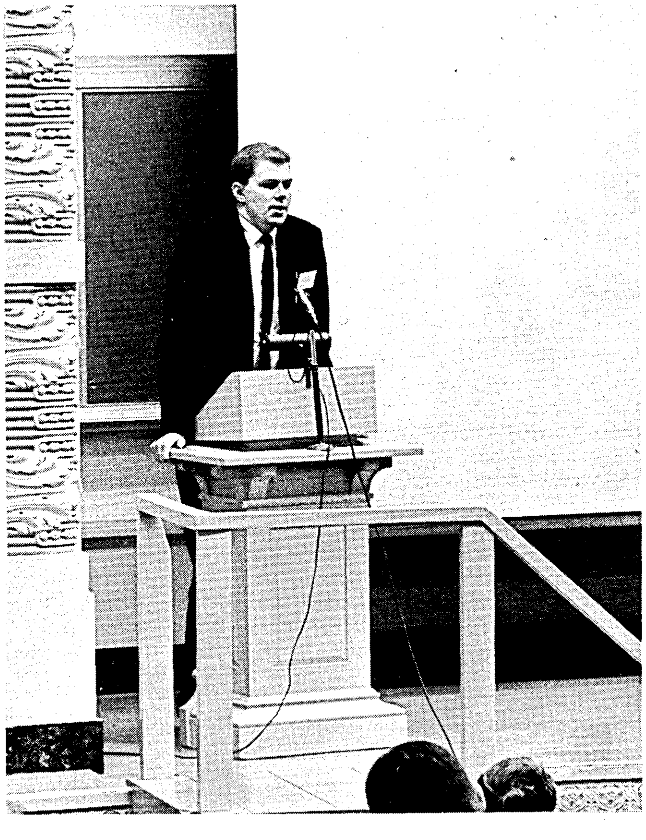
b. Talks and Abstracts. Work resulting from the studies on somatosensory evoked responses was presented at the 1964

Annual Meeting of the Southern Neurosurgical Society and also at the 89th Annual Meeting of the American Neurological Association in Atlantic City in June of 1964

7. Continued Use of LINC.

As time has passed other members of the laboratory staff await training in the use of the LINC computer and have ideas formulated for its use. For example, one of these concerns the possibility that LINC could differentiate between units in normal and diseased muscle and thus markedly speed up the laborious analysis of EMG traces. There are several potential field studies which it is predicted one should be able to conduct efficiently and rapidly by the LINC.

It is earnestly hoped that we will be permitted to continue the use of this instrument since more possible uses arise daily. The instrument is so versatile in meeting laboratory needs that in the future we would be decidedly handicapped if deprived of it as a readily available laboratory instrument.



FINAL REPORT ON LINC COMPUTER EVALUATION

LABORATORY OF NEUROPHYSIOLOGY

UNIVERSITY OF WISCONSIN MEDICAL SCHOOL

Partially supported by

NATIONAL INSTITUTES OF HEALTH GRANT MH-08354-01

March 11, 1965

J. E. HIND

C. D. GEISLER

I. INTRODUCTION

This is the final report of the participation of the Laboratory of Neurophysiology of the University of Wisconsin in the LINC Computer Evaluation Program, July 1, 1963 to March 31, 1965. Our participation has been of immense value to us. Not only have we received considerable training in computer technology, but our laboratory has had the use of a powerful new tool. We are grateful to the National Institutes of Health for their support of this program. We wish also to express deepest appreciation and admiration to the members of the LINC design group who made this development possible.

We have utilized the LINC extensively (for more than 2650 hours) and we are indeed enthusiastic about its performance both as a computer and as a laboratory instrument. The LINC has become an indispensable tool in our laboratory and many of our plans for future research are built around the machine (see section III of this report). We respectfully request that the LINC be assigned on a permanent basis to the Laboratory of Neurophysiology.

II. PAST AND PRESENT RESEARCH

A. Temporal Analysis of Single Neuron Discharges

1. General Development. Prior to the arrival of the LINC in our laboratory, the Control Data Corporation 160 and 1604 computers in the University's Computer Center were utilized to process data from auditory microelectrode experiments. The 160 was arranged to time the occurrence of neural discharges and stimulus events in multiples of 75 microseconds as these data were reproduced from analog magnetic tape. The digital tape produced by the 160 was then processed in the 1604 computer under Fortran programming. Thus the analog data required only one playback in real time into the computer system, after which an almost unlimited variety of analyses could be carried out on the digitalized data. The programs developed for the 1604 include the following: creation of inter-spike interval histograms and calculation of the mean and standard deviation of these distributions together with cumulative conditional probability curves; serial correlation coefficients among successive intervals; relation between mean and standard deviation of collections of interval values under varying stimulus conditions; and post-stimulus histograms including an analysis of the successive inter-spike intervals in a train of discharges evoked by repeated, brief stimuli. This system was just beginning large-scale use when the LINC project was initiated.

Upon arrival of the LINC at Wisconsin, initial effort was concentrated upon arrangements to permit determination of post-stimulus and inter-spike interval histograms on the LINC through introduction of the tape-recorded unit and stimulus pulses into the XL (external level) lines. A bank of Tektronix pulse generators (recently replaced by DEC digital modules, see p. 56 of log book) was inserted between the tape recorder and the LINC to shape the pulses and to set their duration equal to approx. 150 microseconds. Programs were developed which permit the time bins for the histograms to be selected in integral multiples of 100 microseconds; the display provides for two lines of legend typed on the keyboard together with a line of decimal constants which include the full-scale vertical bin value, the total number of spikes in the histogram, the total number of stimuli, and the full-scale time value.

While our initial approach to the LINC involved programs which carried out one specific form of temporal analysis as the analog data were played into the machine, the backlog of taped data awaiting processing soon brought the realization that it was impractical to operate with processes that require the data to be replayed in real time for each analysis or change in analysis parameters. Also fundamental to our thinking was the hope to establish communication between the LINC and the computers in the central university facility which would not only permit use of programs already in operation in the 1604 but would allow further development of statistical analyses of such complexity and size as to be awkward if not impossible on the LINC. Our first thought was to connect the LINC directly to the CDC 160, thus obviating the need for purchase of a digital tape system. However, plans for major changes in staff and hardware in the central computer facility argued against this approach and it was decided to obtain a digital tape unit which could produce tapes with so-called IBM format which would also be compatible with CDC systems.

After investigation of the market, a unit was purchased from the DATAMEC Corporation, the choice being based upon cost, delivery delay, and ease of interfacing with the LINC. The interfacing was carried out by means of 11 DEC plug-in modules installed in the LINC data terminal box. The system has been in satisfactory operation for twelve months and provides read and write capability at the standard density of 200 bits per inch, all operations being carried out under program control. The particular DATAMEC unit purchased is also capable of reading and writing at 556 bits per inch which will permit continuous storage of the highest 6 bits obtained by the SAMi instruction at a 25 KC rate. The program now in use provides the usual lateral and longitudinal parity checks as well as detection of missing characters while reading; end-of-file marks can also be written and searched for. Documentation of our use of the DATAMEC was prepared and sent to the LINC design group.

The acquisition of the DATAMEC tape system comprised one stage in the development of a general-purpose data processing package designed to handle single-unit data for the entire staff of the Laboratory of Neurophysiology. The first step in the process involves timing of the unit discharges and stimulus events as they are introduced into the external lines of the LINC in real time, and so a general timing program was written. Timing is accomplished by the repeated execution of a series of instructions that require 100 microseconds for completion. During each repetition of this basic time loop, the various input lines are sampled (by means of SXL) to determine if a pulse has occurred. If a pulse (either a unit discharge or stimulus marker) is detected, a code word is stored in memory to register the occurrence of the event. Stimulus markers are coded as 0; the code word for the single-unit pulse expresses the time interval which has elapsed since the preceding event, the accumulator being used as a counter which measures elapsed time in increments of 100 microseconds. The program monitors the accumulator and, if it overflows, a code word (7777) signifying this condition is stored in memory. Thus the end result is a table of words in memory which provides complete information about the timing of unit discharges relative to stimulus events. When either the memory has been filled or the end of the data sample has been reached, the appropriate contents of memory are written on LINC tape. Since the number of events which can be stored in the memory is relatively small, the program provides an option whereby the complete analysis of long sequences of data can be accomplished by memory-sized bites. Suitable information is recorded with the output data to permit these bites to be recombined automatically in later processing. As presently arranged, each experimental sequence may consist of from 1 to 21 LINC tape blocks.

Since many of the single-unit recordings in our laboratory involve upwards of 100 different sequences of varying experimental conditions, the ability to identify and edit the data is of great importance. An editing program has been developed which, in addition to permitting the addition of identifying information and comments in manuscript form, will re-arrange the experimental sequences in any desired order, thus simplifying further processing and interpretation. After production of the edited LINC tape the information can either be copied on the DATAMEC unit for processing on the CDC 1604 or can be further processed on the LINC by programs which perform a variety of analyses which do not require extensive arithmetic calculation. At this stage, many questions will have been answered satisfactorily and there will be no need for further processing; those data requiring the capabilities of a larger computer may be selected for 1604 processing. The larger machine has proven especially useful for analyses which require a great deal of data shuffling and/or extensive arithmetic computations. (See Appendix 1 for details of our 1604 programs). The display-oriented LINC programs include: post-stimulus and inter-spike interval histograms having scope displays with keyboard-controlled selection of parameters; dot displays showing the time of occurrence of unit discharges with respect to stimulus events; and plots of discharge rate versus time with variable count intervals.

Permanent output records from LINC are obtained by Polaroid oscilloscope photos, by a MOSELY X-Y plotter, or by a recently arrived Teletype unit. The basic scheme for utilizing the X-Y plotter is that of Woodbury and Gordon (April 13, 1964 communication) modified to allow the automatic plotting of waveforms in an array ranging up to 8 x 8 in size. Positioning of the waveforms within the array is controlled by the contents of the relay register; simple digital-analog ladder networks are connected to each of the two octal digits of this register to provide X and Y coordinate voltages which are added to the waveform signals (see p. 50 of log book).

2. Specific Projects. Discharges from INFERIOR COLLICULUS NEURONS have been studied in experiments which stressed binaural interaction (ref. 3). The LINC post-stimulus histogram display of these data has been particularly useful in depicting the time-course of inhibitory and facilitory effects from the respective ears through the use of stimuli which partially overlapped in time. The fine structure of the discharge trains has been examined by means of Dr. J. Rose's 1604 computer programs. (See Appendix 1).

Discharges of SUPERIOR OLIVARY COMPLEX NEURONS to tone-burst stimuli have been studied using LINC. Dr. J. Goldberg developed a special real-time histogramming program, because he found that the 100 microsecond resolution of the original programs was not sufficient to resolve the time-dependent behavior of units in the superior olivary complex. The program utilized a sequential array of SXL instructions and could achieve 8 microsecond resolution.

CORTICAL NEURON discharges have been studied using the LINC histogram programs. In addition, Dr. N. Dubrovsky extended Dr. Goldberg's program to provide the mean and standard deviation of the initial latency histogram, both values expressed as decimal numbers with precision equivalent to the individual latency measurements.

COCHLEAR NUCLEUS NEURON discharges have been studied extensively (ref's 1, 2). Most of the analyses were done on the CDC 1604, using the LINC-DATAMEC system to prepare the data for the larger machine. Statistical programs calculating mean interval, standard deviation of intervals, and correlation between intervals were utilized (see Appendix 1 for details). In addition, Dr. D. Greenwood developed and utilized a LINC program which displayed the data in a Lettvin dot-pattern display. Dr. Greenwood's program was one of the first which operated directly on the digital data previously written by the 100 microsecond timing program.

The discharges of THALAMIC NEURONS RESPONDING TO THERMAL STIMULATION OF THE TONGUE are being studied using the LINC (ref. 4). Data prepared by the 100 microsecond timing program is handled by a program of Dr. R. Bernard. The number of discharges in variable-width time bins is calculated and displayed both on the oscilloscope and on the X-Y plotter.

A study of NEURONS OF THE AUDITORY NERVE is just beginning and so far has utilized the standard histogramming programs.

B. Evoked Potential Averaging.

Although virtually all of our applications of the LINC have thus far been directed toward the temporal analysis of single-unit discharges, Mrs. J. Hirsch, a graduate student, has used the machine to average evoked potentials. Mrs. Hirsch's program, which has been used on-line, will sample up to 8 electrodes. After each stimulus the last response is displayed, together with the running average, the average being divided by the proper integer to maintain a constant vertical scale factor. The output can be displayed on the oscilloscope or written on the X-Y plotter.

C. Pilot Study of Automated Clinical Procedures.

Taking advantage of time when the LINC was not being used by the members of the Laboratory of Neurophysiology, Dr. P. Hicks of the Department of Medicine used it to study the feasibility of automating the processing of laboratory test results and the collecting of patient history data. Using our display subroutines, Dr. Hicks developed several programs for the manipulation of laboratory test data and for the display of medical history questions to patients. On the basis of his favorable experience with the LINC, Dr. Hicks hopes The Clinical Laboratory will shortly obtain its own computer equipment (see Appendix 3 for Dr. Hick's report).

III. FUTURE RESEARCH

A. Neurophysiology Computer Laboratory. Assuming that the LINC is assigned to the Laboratory of Neurophysiology on a permanent basis, we plan to expand our present applications including both the analysis of the temporal characteristics of unit discharge and the processing of evoked potential data. In our use of the LINC thus far it has proven impractical to use the machine on-line during experiments. This limitation arose because the LINC could not be installed permanently in the auditory microelectrode laboratory in which most of the experiments are conducted and which is located several hundred feet away from the only other available site. The difficulty was due not only to the severe shortage of space in the auditory laboratory but also to the fact that this facility was used frequently for other experiments which would have interfered with the use of LINC for off-line applications including processing of a large backlog of magnetically taped data. Attempts to move the machine between the auditory laboratory and the tape processing site were limited to a few trials by the awkward character of the procedure.

From this experience arose plans for a new Neurophysiology Computer Laboratory which will provide for the use of LINC both in on-line experiments and in off-line applications. The Medical School administration is furnishing construction funds and approximately 900 square feet of floor space for this project which is in the final stage of architectural design. A folding partition will separate the LINC room from a sound-insulated electrophysiology laboratory so that the two facilities can be used either together or independently.

The on-line use of LINC during experiments is important from two standpoints. There is first an obvious advantage in having the results of analysis immediately available in order to fully exploit the experimental situation. On several occasions, off-line analysis of our taped data has revealed results which were completely unsuspected during the experimental session. A second and potentially more significant benefit of on-line operation lies in the possibility of using the computer to control the experiment. In our typical auditory single unit recording procedure, much time is expended in deciding the programming of the stimuli and in setting and checking the many knobs and dials. If stimulus programming can be reduced to a systematic formula which may include response contingencies, computer control can effect a great savings in time and eliminate human error in adjustment of stimulus parameters. Moreover, the experimental procedure would then be rigorously defined in contrast to the subjective approach which characterizes our present efforts.

We are currently designing equipment in which the LINC will be able to control the timing, frequency, and intensity of tonal stimuli. This system will be used to continue our studies of stimulus coding in single units at several levels of the auditory system. This will include a continuation of our present investigation of binaural interaction in units of the inferior colliculus as well as a new series of experiments on eighth nerve fibers in the squirrel monkey. While the initial effort in the Neurophysiology Computer Laboratory will thus be concentrated upon auditory studies, the facility will be designed to handle a broad range of electrophysiological experiments.

B. Biomedical Computing Division and LINC/3600 Connection. Early in 1963, one of us (JEH) was appointed chairman of an ad hoc Medical Center Computing Services Committee which was requested by the Dean to determine means for facilitating the application of computer science to the research and teaching program of the institution. The committee's deliberations culminated in plans to establish a Biomedical Computing Division (BCD) as a part of the University of Wisconsin Computing Center (UWCC); an application for support of this activity is currently under review by the NIH. Our fruitful experience with LINC resulted in plans for the initial equipment configuration of the BCD to be centered around a second LINC which will be installed in a new Medical School Data Acquisition Laboratory with an on-line, priority interrupt connection to the CDC 3600 computer at the UWCC. The configuration will also include a line printer and card reader on-line to the 3600 together with a digital plotter and Teletype to be driven by the LINC (a block diagram of the equipment is included as Appendix 4).

As an initial step in this development, the LINC now in the Laboratory of Neurophysiology under the Evaluation Program is in the process of being connected to the 3600. Communication between the LINC and 3600 will be handled by a pair of shielded 3600 input/output cables, each cable 5/8 inch in diameter and containing 29 twisted-pair transmission lines. The cables will be routed through an existing tunnel beneath the street which separates Sterling Hall (site of the UWCC) and the Medical Sciences Building. The required length of cable has been estimated to be well below the 1000 foot maximum distance which can be handled by the 3600 circuits when high-powered line driver cards are used. The cables include 12 bi-directional data lines plus a parity bit together with status and control lines.

The design of the interface between the LINC and the CDC transmission cable is primarily the responsibility of John Keenan who recently joined the BCD organization as an engineer. Keenan has designated the interface as the SUTURE box; a draft copy of his description is included as Appendix 5. The design provides for the exchange of control information by means of a common flag register which is shared by the two computers. Keenan has proposed a special version of the OPR instruction which he has named the "FUNCTION" instruction which will perform a variety of tasks, e.g., setting, clearing, or sensing of bits in the flag register and a word-count register. A draft copy description of the proposed "FUNCTION" instruction is included as Appendix 6.

The system will permit two modes of transmission from the LINC to the 3600. The usual mode of operation will involve transfers of one or more blocks of data from LINC memory via GULP output at a rate of one LINC word every 8 to 14 microseconds. Under the influence of control words at the beginning of each block, the 3600 will either process each block as it is received or store the information on tape or drum until the last block of a sequence is received. A second mode will be provided for data which require relatively long periods of uninterrupted analog sampling at the maximum rate of one word every 24 microseconds. In this mode the data will be stored at the 3600 on tape or drum for subsequent processing. Output from the 3600 will be sent back to the BCD laboratory either to the line printer or to the LINC where it may be displayed on the scope or digital plotter.

IV. TRAINING PROGRAM

A. Strong Points

The greatest training benefit that we have received from the evaluation program is the experience of having had a computer in our laboratory during a critical time in our development. As described in section I, we had made considerable use of the University's computing facilities before LINC came.

Such use, however, was clumsy, because it involved three separate processes: editing and re-recording of analog data obtained during experiments, playback of the re-recorded analog data into the CDC 160 computer for digital representation, and computer analysis of the data by the CDC 1604 computer. Moreover, because the 160 is used as the input/output device for the 1604, the second stage of our process essentially shut down the University's main computer. The volume of our data and the load on the 1604 was small enough at the time to make such a system workable; it would not be workable today. Therefore, sometime between the spring of 1963 and today we would have had to greatly reduce our computer operations or to acquire a commercial computer for our laboratory. Such an acquisition would have been lengthy, probably costly, and would have been based on a very limited knowledge of computers. The presence of the LINC has given all of the workers in our laboratory easy access to a computer, and has given us invaluable insight into just what the computing requirements of our laboratory are and will be. It is this insight that has enabled one of us (JEH) to formulate plans for the Medical School's Biomedical Computing Division.

The use of the LINC by our laboratory has also given us an excellent introduction to digital engineering. Some understanding of digital engineering was, of course, achieved during LINC assembly and debugging. The design of the interfaces between LINC and additional output devices such as the DATAMEC tape system and the MOSELEY X-Y plotter required us to achieve a still more fundamental understanding of digital systems. The knowledge of digital engineering that we have gained using LINC provided the basis for plans for the LINC/3600 connection and for the peripheral devices to be used in the Medical School's Biomedical Computing Division.

Of importance in judging the training aspects of the evaluation program is a recognition of the number of scientists who have come to use computers because of it. We have taught two separate 1-week courses in the use of the LINC. The first group of 16 students was composed primarily of our colleagues in the laboratory; the second group consisted of 6 members of our laboratory, 16 from elsewhere in our institution, and 2 from Brown University. That these courses and the ease of using the LINC have stimulated the use of computers is demonstrated by the list in Section I of seven scientists besides ourselves who have programmed and utilized the LINC extensively. In addition, Drs. J. Goldberg and R. Bernard who have since left our laboratory for faculty positions elsewhere are both actively engaged in the use and development of computing facilities for their new laboratories. Moreover, an introduction to computing has been given to scientists from other departments in our Medical School including Dr. P. Hicks of the Department of Medicine (see section II-C) and Dr. F. Graham of the Department of Pediatrics who is presently implementing a system for the computer analysis of neonate polygraph records.

Finally, we found the interim report meeting to be valuable in learning how other participants in the evaluation program had made use of LINC and what kinds of programs they had written.

B. Weak Points

We can easily conceive of the LINC being applied to a well-defined problem such as response averaging which, once set up, could be operated for extended periods with little time required for continuing program development or the addition of new peripheral equipment. Our experience has, by choice, been quite to the contrary, with relatively great investments in time devoted to the continuing development of programming and associated apparatus. Another important characteristic of our operation arises in the fact that, while the LINC has been described as essentially a personalized instrument for use by an individual investigator, we have attempted to meet the experimental demands of a rather large group of investigators, thus placing a premium on the development of standardized operating procedures of the highest possible efficiency. Group utilization also entails scheduling, relatively large demands for continuing training and supervision, and careful documentation of programs and operating procedures. Taken together, these activities have required essentially full time attention from one of us (JEH) and the better part of half-time activity by the other (CDG) for the first year. Since the summer of 1964 these activities have demanded somewhat less time. Lest our position be misunderstood we wish to emphasize that we are convinced that this investment in time and effort has already proven valuable to the overall program of our laboratory and will eventually prove valuable to our own research activities. Nevertheless, we can see now that a professional programmer would have been a great help to us and would have saved us much time. For the future, therefore, we recommend that participating laboratories be advised to hire a programmer and perhaps an engineer. It is unreasonable to expect a scientist with a full-time research program to handle all the responsibility for LINC in a dynamic evolving application.

Along this same line, the dearth of documentation has significantly retarded our progress. First of all, we were given insufficient material for the training of other users which required that much of the information about the LINC and its operation be communicated verbally by the two of us. A programming manual would have been of great help. Secondly, there was insufficient documentation at the level of engineering description. The glossary was of limited usefulness because of incomplete cross-references. The engineering drawings provided valuable additional information but arrived late in the program. In our opinion, the revised write-up of the order code, except for the glossary and the description of the OPR and magnetic tape instructions, was inferior to the original. The only block diagram of the whole machine we have seen was one which one of us (JEH) made. Consequently, much time was spent in discovering how the machine works, especially the operation of the typewriter keyboard (see Appendix 2 for our write-up of it), the functioning of the OPR instruction, and the input/output connections. We hasten to add that the LINC's designers were quick to fill a specific request for information or help. S. Ornstein, for instance, supplied us with a parity-generating network for the DATAMEC system in a matter of a few days.

There was an almost negligible exchange of programs between us and other participants. Just learning about the existence of a program was not sufficient incentive to send for a copy, because documentation was typically deficient. The useful exchange of programs would almost certainly have required a large central staff to perform checking, documentation and distribution.

Finally, we feel that the lack of arithmetic subroutines hastended to bias users away from statistical methods. In our own laboratory, the parallel usage of the CDC 1604 computer for statistical computations tended to prevent such a bias from developing.

V. COMPUTER PERFORMANCE

A. Maintenance

Relatively insignificant time has been lost due to malfunction of the LINC.

On two occasions the front-back connectors developed high resistance and caused failure in Z register operations. Removal of all wax from these connectors seems to have eliminated this problem. Additional trouble due to poor contact was also encountered in the connectors at the rear of the data terminal box plug-in.

A minor but annoying difficulty encountered while using the Polaroid camera is the instability of the CRT beam brightness in the cart scope. Consultation with Tektronix suggested this is an inherent defect in the 561A oscilloscope.

Probably the most serious malfunction encountered is the tendency for the keyboard to exhibit contact bounce. While the contacts would be free of bounce at the time the KSTR level first came up, there would frequently be chatter between 1.5 to 6 or 7 milliseconds following this event. If the program happened to sample the keyboard digit lines during the bounce, erroneous characters would frequently be read. Attempts to eliminate this difficulty by adjusting the stroke of the rotary solenoid or by changing the timing of the common (synchronizing) switch contacts were unsuccessful. However, completely reliable operation was achieved by inserting 100 ohm resistors in series with the input lines to the B register together with 1 microfarad capacitors to ground at the LINC end of the resistors (see p. 37 of log book).

A diode and transistor failed on a decoder card (type 1151G) in October, 1964 and were replaced (see p. 54 of log book). We were later told by a DEC representative that the cards carry a life-time guarantee and that the company would have replaced it without charge.

The keyboard began to fail again in January, 1965, and the trouble was corrected by lubrication after a thorough study of keyboard action (see Appendix 2! for our summary).

One of our often-used magnetic tape reels has become so worn that it now slips on its hub. Replacement of the hub's rubber bands improved the situation somewhat, but not completely. Because we never had this problem with the old hubs, we feel the new hub design to be inferior to the original.

Finally, we replaced our original console CRT tube in January, 1965, because it burned at the place where the LAP3 line number is displayed. The replacement tube burned even more badly within one week, but we have left it in.

In these service operations as well as in the development of peripheral equipment we received a significant amount of assistance from Harry Ludwig, Director of the Medical School's Medical Electronics Laboratory.

B. Adequacies and inadequacies of the LINC for our application

The LINC has proven itself admirably suited for most of our needs. All of the users have found its easy acceptance of analog data, its ability to manipulate these data non-arithmetically, and its oscilloscopic display of the results most useful: LINC is an essential tool in our laboratory. We feel, however, that it has two weaknesses when operating without added peripheral devices: a lack of arithmetic subroutines and the inadequacy of the oscilloscope and Polaroid camera as the only means of data output. The lack of arithmetic subroutines has discouraged most of our users from doing any but the simplest statistical analyses on the LINC. Dr. N. Dubrovsky did write a program for finding means and standard deviations, but it was very slow and restricted in application. However, the use of the 1604 computer, via the DATAMEC tape system, has allowed us all the computing power we needed. As regards output, we have found Polaroid prints to be expensive, messy and inadequate for displaying numerical data. We have also used a MOSELEY plotter as an output device, but it suffers from chronic instability and the difficulty of writing alphanumeric characters. We believe the teletype, which was installed only this month, will greatly facilitate the handling of alphanumeric output and perhaps graphs also. It is clear, therefore, that the two weaknesses of the LINC mentioned above have been largely overcome by the use of easily connectable external devices.

There are several additional features that we would like to see on LINC. 1) Because of the number of peripheral devices we use, we would prefer a larger relay register, at least 12 bits. For us, the relay register need consist only of flip-flops, not relays, and a means of setting these flip-flops other than through the Accumulator would be useful (e.g. J. Keenan's EXTERNAL FUNCTION INSTRUCTION, Appendix 6). 2) A selection device for the control of several peripheral devices simultaneously connected to LINC seems desirable and probably should be standarized to facilitate program exchange. 3) An external interrupt which could be "masked out" under program control would be very useful in handling peripheral devices. 4) A second tape transport would be useful for situations requiring a program tape in addition to input and output data tapes. Such a situation arises for example, in our planned LINC/3600 system. 5) We would like the OPR instruction modified so that data could be strobed into the B register either by OPR 2.1 or by QKrestart. Presently, the data is strobed in by OPR 2.1, which means that the external device must maintain its data lines for up to 4 microseconds after its "ready" condition has been established. 6) A Grass Kymograph camera adapted for use with the LINC should provide faster, cheaper photographic recording than the Polaroid camera. 7) A delay to reduce the duty cycle of LAP3 when it is displaying only the manuscript line number might reduce CRT burning. 8) A package of arithmetic subroutines (e.g. single and double-precision floating point add, multiply and divide; square root; logarithm) is really needed and would be most useful. To a lesser extent, a simple LINC FORTRAN-like compiler would be useful.

The LINC, with the use of peripheral equipment and more softwave, approaches our idea of an ideal laboratory computer. In addition to existing LINC features and those we suggested above, we would find the following useful: asynchronous, buffered memory access by peripheral devices; a low-cost auxiliary bulk storage device such as the Potter RAM; multiple remote consoles, any one of which could be easily selected at the main frame. We would like the price to be no greater than $40,000, preferably near $30,000.

VI. BIBLIOGRAPHY

A. Papers and talks at Society meetings

1. J. M. Goldberg and D. D. Greenwood, "Response of neurons of the cochlear nucleus to tones and noise of long duration", in preparation.

2. J. M. Goldberg and D. D. Greenwood, "Responses of neurons in the cochlear nucleus to tones and noise of long duration". Science, 1964, 146, 432-433 (abstract).

3. J. E. Rose, N. B. Gross, C. D. Geisler, and J. E. Hind, "Some effects of binaural stimulation on the activity of single neurons in the inferior colliculus of the cat", Science, 1964, 146, 433 (abstract).

4. D. A. Poulos and R. M. Benjamin, "Response of thalamic neurons to thermal stimulation of the tongue," (talk to be given at the Federation meeting, April, 1965).

5. J. E. Hind, "Computer analysis of the temporal distribution of neuronal discharges" (talk to be given at the 3rd National I.S.A. Biomedical Sciences Instrumentation Symposium, April 19-21, 1965).

B. Talks to the University Community

J. E. Hind - Demonstrations of LINC for the Psychology Department Faculty and for the Limnology Journal Club.

APPENDIX I

1604 Programs

The Laboratory of Neurophysiology's use of the University's large Computer (CDC 1604) for the analysis of single-nerve pulse data has been extensive. Before the arrival of LINC, the data were prepared for analysis by the University's CDC 160 computer. However, our LINC programs which put single-neuron data onto IBM-compatible tape have been made to encode the data in almost the same form as the 160 did; the modification of one subroutine enabled the programs developed for the 160 to handle the LINC's output.

There are three subroutines which read data from the digital tape into the 1604's memory. Since the writing of the data onto digital tape by LINC program DATAMEC RUN is done in records of 64 words (48-bits each), these records are not readable by FORTRAN statements. The subrountines, therefore, are written in "machine" language. It should be noted, however, that DATAMEC RUN could easily be modified so that its output records could be read by FORTRAN statements. The three subroutines are RD1EOFL, RD1REC, and RD1SEQ. RD1EOFL reads successive records on a tape until an "end-of-file" is read. We use an end-of-file mark for punctuation on the tape, and consequently, this subroutine is used for initial positioning of the digital tape. RD1REC reads the next record from the tape into memory and arranges it into a FORTRAN-readable array. RD1SEQ reads the next entire "sequence" of data from the digital tape into memory and assembles it into a single FORTRAN-readable array. This subroutine utilizes the internal labelling of the data that is done by the LINC programs and it can read and assemble as many as 5373 (a convenient but arbitrary number) data words into the array.

So far, we have used the 1604 to analyze single-neuron pulse data taken mainly from three "auditory" nuclei -- the inferior colliculus, the superior olive, and the cochlear nucleus. Different types of data have been taken from these three centers and two basically different types of programs have been developed.

a. Inferior Colliculus Data

The analysis of single-neuron data from the inferior colliculus has so far been mostly concerned with the time of occurrence of neural pulses during a stimulus period. In a typical experiment, each stimulus presentation excites a series of pulses from the neuron under study. If it be imagined that these pulses are numbered from 1 upwards in the order of their occurrence, we have a label for each pulse (i.e. 1st pulse, 2nd pulse, etc.). Many repetitions of each stimulus condition are usually given and form a family of stimulus periods.

The program for the analysis of data taken from the inferior colliculus is written in the form of a package of subroutines. Due to the fact that not all subroutines are used on all data, regular 80-column punch cards are read by the main program during run time, and the values of various words on the cards determine which subroutines are to be called and also provide needed constants.

Subroutine ANN1 is concerned with the initial latent period (time to the first pulse) following the onset of a stimulus. The subroutine takes the initial latent period of each stimulus trial (of up to 150 trials) and for this group of periods calculates the mean, the standard deviation and the standard error of the mean. After this computation, the subroutine examines the latency of each initial pulse and rejects those trials whose initial pulse latency does not fall within 6 standard deviations of the mean. The subroutine then recomputes the same statistics on the accepted data. Various printouts of the data are given.

Subroutine BETTY is concerned with the intervals between the various pulses in a stimulus period. Operating only on the data accepted by subroutine ANNI, this subroutine examines all the accepted stimulus trials, collects together all the first inter-pulse intervals (i.e. the intervals between the first and second neural spikes), and then computes the mean, standard deviation and the standard error of the mean for this collection. Next it collects the second inter-pulse intervals (i.e. the intervals between the second and third neural spikes) for all accepted trials and calculates their statistics. Subroutine BETTY does this for all inter-pulse intervals up to the forty-ninth. The subroutine prints out the calculations for each inter-pulse interval.

Subroutine CAROL arranges the neural pulse data for display as a post-stimulus histogram. Time following each stimulus onset is divided into 1000 equal bins (the size of which is set by the control card) and a number is assigned to each bin. The number is the count of neural pulses which have an absolute latency that falls within the interval's boundaries. An overflow counter numbers the neural pulses which have absolute latencies too great to be represented in the histogram. Data from different sequences can be combined by CAROL. On command, it will print out the numbers it has calculated. CAROL also prepares a squeezed array (the original 1000 bins compressed to 100) for plotting as a histogram by subroutine HISTGM.

Subroutine CARMEN is used to obtain a 100 word array (for histogram display) of only a small portion of the 1000-word array used in subroutine CAROL. Control words tell the subroutine just where to look for its 100-word array.

Subroutine DIANA analyzes the temporal patterning of the labelled pulses. First, it identifies the initial neural pulse (providing it exists) of each stimulus trial. Those pulses having latencies differing from the mean latency (calculated by subroutine ANNI) by an amount specified on the control card are rejected and the remaining latencies are drawn up into a 50-word array for display as a histogram. Next, the second neural pulse (if it exists) of each stimulus trial is identified and the collection of second pulses treated in the same way: the group is pruned according to their absolute latencies (using calculations of subroutine BETTY) and all qualifying latencies are drawn up into a 50-word array. Every neural pulse up to and including the 50th one of the stimulus period is treated in this way. The subroutine has an extensive printout as well as its 50-word arrays for display as histograms.

Subroutine TIMECAL is used to analyze the timing of pulses originally generated by very accurate equipment. Its output not only calibrates our timing programs but also provides an indication of the time errors involved in our total system.

Subroutine HISTGM plots out on the 120-character line printer a 100-

member array as a bar graph (histogram). One line of print deals with one member of the array. On that line the member's vital statistics are printed as well as a row of asterisks numbering from 1 to 100. These asterisks form one bar of the (self-normalized) bar graph.

b. Superior Olive and Cochlear Nucleus Data

The analysis of single-neuron data from the superior olive and cochlear nucleus has been mainly concerned with the distribution of inter-pulse intervals during long periods of exitation. The use of this analysis has produced a significant new way of looking at single-neuron data (ref. "Response of Neurons of the Superior Olivary Complex of the Cat to Acoustic Stimuli of Long Duration: by J. M. Goldberg, H. O. Adrian and F. D. Smith, J. Neurophysiol. 1964, 27, pp. 706-749).

The program used to analyze these data also used punch cards to control calculations. The program includes four basic statistical packages.

INTERPULSE-INTERVAL HISTOGRAMS. Histograms are constructed of the distribution of intervals between successive neural pulses in a long stimulus trial. The binwidth of the computed histograms is 0.675 msec. The usual procedure entailed the pooling of data obtained from a number of stimulus trials.

CONDITIONAL PROBABILITY ANALYSIS. To understand the conditional probability measure, consider a small bin of time, k, of width Δt whose upper limit coincides with the time $t=k\cdot\Delta t$ after the occurrence of a neural pulse at time $t=0$. If the condition is met that no neural pulse occurs in the interval between the occurrence of a neural pulse $t=0$ and the beginning of the kth bin, then the probability that a pulse occurs within the limits of the kth bin is the conditional probability. It measures the likelihood of occurrence of a neural pulse as a function of the time elapsed from the occurrence of the immediately preceding pulse and, as such, may be assumed to mirror the excitatory state of the neuron during the period $k\cdot\Delta t - \Delta t \leqslant t \leqslant k\cdot\Delta t$. The conditional probability p_k is estimated from empirical interpulse interval distributions by use of the formula

$$p_k = N_k/N_{n \geqslant k}$$

where N_k is the number of intervals whose values fall within the limits of the kth bin and $N_{n \geqslant k}$ is the number of intervals whose values exceed the lower limits of the kth bin. When estimations were made from experimentally obtained distributions, p_k was found to be a monotonically-increasing function of k, the shape of the function presumably reflecting recovery from the effects of the immediately preceding discharge. In the estimation of p_k, the number $N_{n \geqslant k}$ in Eq. 1 may be considered the size of the impirical sample. As k increases and, as a consequence, $N_{n \geqslant k}$ decreases, the estimation becomes unreliable and eventually attains the value of unity in a trivial manner. For this reason, p_k is estimated only for those time bins for which $N_{n \geqslant k}$ exceeds 100 or so events.

MEAN VS. STANDARD DEVIATION. A stimulus period is first divided into groups of 21 consecutive spikes, the 21st spike of one group also functioning as the 1st spike of the next group. The 20 intervals derived from each group are pooled and a mean interval "$\bar{x}$" and standard deviation of intervals "s" computed, according to the formula

$$s = \sqrt{\frac{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}{N(N-1)}}$$

Where N = 20 and x_i is the value of the ith interval of the group.

The use of groups of 20 intervals rather than those of larger size is dictated by the presence of adaptation. Adaptation produces a continuous change in the mean interval and, hence, introduces a variance s_A^2 whose magnitude may be estimated by the equation

$$s_A^2 = (\bar{x}_2 - \bar{x}_1)^2/12$$

where $\bar{x}_2$ is the mean interval at the end of the group and $\bar{x}_1$ that at the beginning of the group; the two means may differ as a result of adaptation. By making the duration of the group short, $\bar{x}_2$ approaches $\bar{x}_1$ in value and the magnitude of s_A^2 becomes, at least in most instances tested to date, negligible compared to the total variance of the sample.

It was not uncommon to collect sufficient data from a single neuron to form 5000 groups of 20 intervals. Two procedures have been employed to reduce the number of data points to be considered. In the first, the corresponding statistics derived from five consecutive groups are averaged. In the second, the average value of the standard deviation is determined for all groups which possessed similar values of the mean interval.

CORRELATION ANALYSIS. Stimulus periods are divided into groups of 20 consecutive intervals. If it be imagined that all intervals in a record are numbered in the order of their occurrence, then each of the 20 intervals of the group - call it the ith interval - is paired with the (i+k)th interval, even when the latter is not among the original group of 20. On the basis of this pairing, a serial correlation coefficient r_k is calculated. For each group, correlation coefficients are computed for values of k ranging from k = 1-10, the computations being performed according to the expression

$$r_k = \frac{N\sum_{i=1}^{N} x_i x_{i+k} - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} x_{i+k}}{\sqrt{\left[N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2\right]\left[N\sum_{i=1}^{N} x_{i+k}^2 - \left(\sum_{i=1}^{N} x_{i+k}\right)^2\right]}} + C(N,k)$$

where N = 20, x_i is the value of the ith interval and C(N,k) is a term of value (N-k)/N(N-1). It is necessary to introduce the factor C(N,k) in order for the expected value of the statistic r_k to be zero when neighboring intervals are linearly independent of one another.

APPENDIX 2

LINC Keyboard & Keyboard Logic Operation

1. When any key is pressed upon the keyboard, the solenoid switch (near the center in the schematic, figure 1) is closed, which causes "picking" or closure of the KBD (keyboard) solenoid. This is done by completing (grounding) the circuit which goes through the KAR normally closed (NC) contact marked "A", then through the KBD contact labeled KCC (keyboard common contacts) "A" NC, and finally to the ground side of the KBD relay energizing coil. The -15 volt side of the KBD relay energizing coil is fed by pin 23 through the current limiting and filtering network of C2, R2, and C1.

2. The KBD solenoid and its mechanical linkages to the coding bars, cause the coding bars to move and open other contacts when it operates. The coding bars (there are six for the LINC) are allowed to move and operate keyboard data contacts (called KC1 through KC6) depending upon which key is pressed. When the KBD solenoid operates, all selected coding bars move (i.e. those coding bars which are not inhibited by stops associated with each key when it is pressed). When the coding bars move, they open the normally-closed contact on the KC contact set associated with the bit represented by the coding bar. When these KC contacts are opened the ground signal is removed and a resistor bias network places a negative voltage, representing a "one" upon the respective data line (pins 1 through 6 in connector J1).

3. Because of mechanical variations, the time at which the coding bars cause opening of the KC contacts varies over a range of about 2 milliseconds. For this reason, it is "safe" to read the data lines only after a "settling" period. The end of this settling period is signaled by the disappearance of the "RLSD" (released) signal and the appearance of the "KSTR" (Key struck) signal. This is accomplished by the changing of the connection on KCC contact "B" from the NC position to the NO (normally open) position. The "KSTR" signal is jumpered externally to the keyboard from pin 15 to pin 17 so that the condition "NOT KAR" is a part of "KSTR" when it is seen by the LINC. KAR is the keyboard anti-release relay. The "NOT KAR" portion is "anded" (logically multiplied) through use of contact B on the KAR relay. The adjustment of the KCC contacts is such that KCC changes about 2 milliseconds after all the data contacts (KC) have operated (see figure two).

4. When the KBD solenoid, which is a rotary action solenoid, operates another contact KCC "A" NC opens causing the resistor R3 to enter the KBD energizing coil circuit. This action reduces the current passing through the coil, while it is being held in the closed position. The movement of the KBD solenoid is such that in the operated position the KCC contacts are operated, yet in the released condition the KC and the KCC contacts are not operated (see figure two).

5. Within the LINC, the "KSTR" signal is used by the LINC to tell the program that a key has been struck. When this keyboard information is read from the keyboard data lines by a computer KBD instruction, a flip-flop (called RELIP--Release in progress) is set in the LINC. The RELIP signal is transmitted to the keyboard logic on pin 20 where it causes transistors T1 and T2 to place a ground potential (or nearly ground) on one terminal of the KAR relay; the other terminal is at -15 volts because of resistor R1 on pin 19. KAR is thus picked (operated) by RELIP; it can also be picked by pressing the reset switch on the

back of the keyboard.

6. While the KAR relay is closed the "KSTR" signal is removed from its path to the LINC by opening of KAR contact "B" NC. The picking (operating) of the KAR relay also causes interruption of the KBD solenoid power line by changing the contacts at KAR "A" NC. This interruption causes release of the KBD solenoid which in turn causes restoral of the "RLSD" signal to pin 18. The "RLSD" signal is used by the LINC to reset the RELIP flip-flop. Removal of the RELIP[1] signal from pin 20 will remove the ground from the KAR relay releasing the relay unless the operator has the key still depressed. If the operators finger is still on the key, the KAR "A" NO contact will keep the KAR relay operated.

7. In the adjustment of the KCC contacts two conditions must be avoided. If the KCC contacts close too late, mechanical movement of the KBD solenoid assembly may not be sufficient to give a reliable and consistent closure of the KCC contacts. If this happens "RLSD" will remain, "KSTR" will never be generated, and the LINC will not know that there is a character to be read from the keyboard. The LINC will continue to wait, no KBD instruction will be issued and thus the keyboard will only be released if the reset button is pushed by the operator. On the other hand if the KCC contacts close too early, data errors can result from failure of the data lines to "settle" properly before reading occurs. In the extreme case the KCC contacts may operate so early that they are always in the operated position, "KSTR" may be always generated, "RLSD" will never be generated and again the keyboard will stick because the proper reading and releasing sequence cannot take place. For example if "KSTR" operates allright but "RLSD" does not, the key will be read but no "RLSD" will appear to cause resetting of the RELIP flip-flop. In this case the KAR relay will remain in the operated position, the keys may remain down and operating the reset button by the operator will not clear the trouble.

LINC - SOROBAN KEYBOARD CIRCUIT

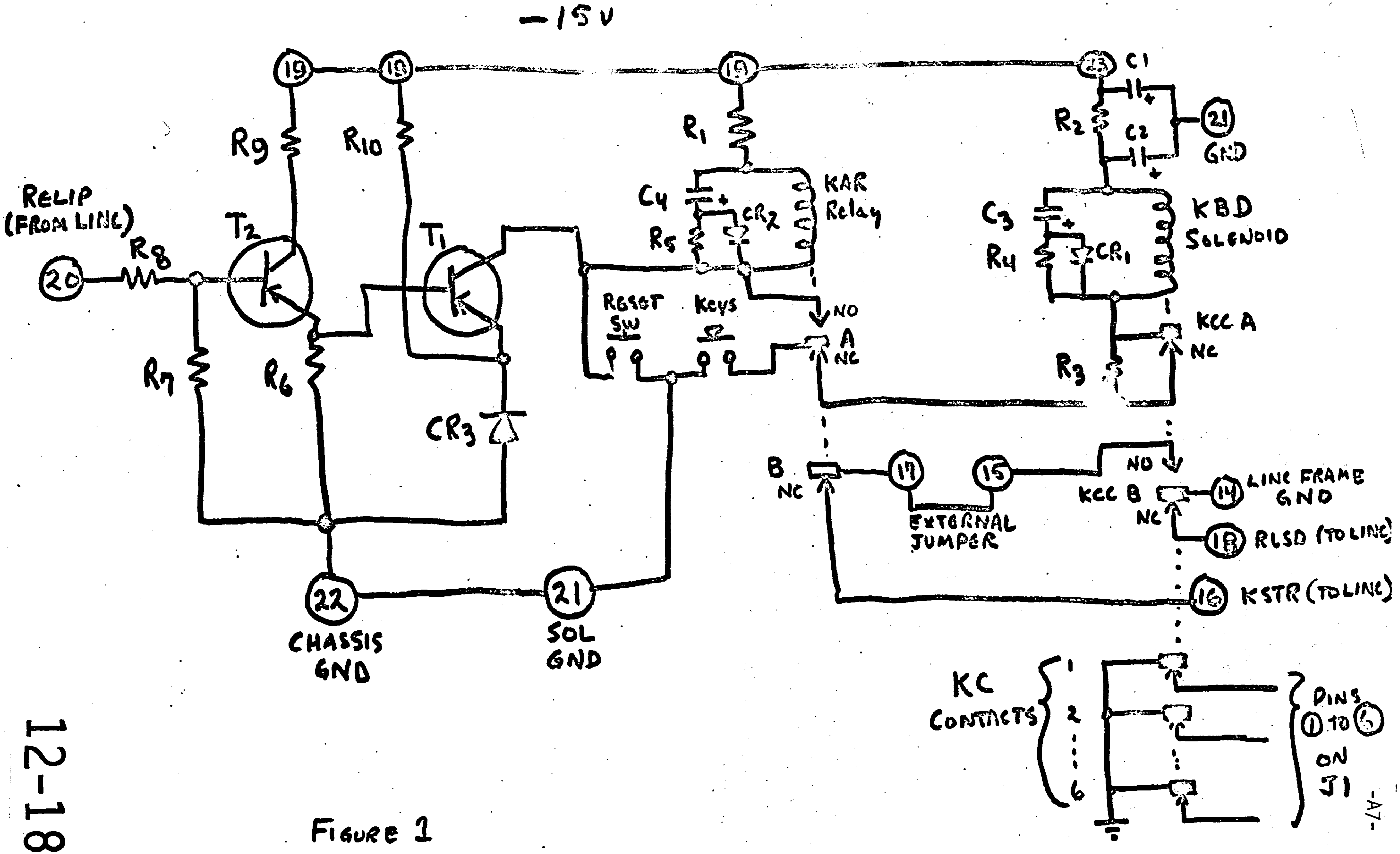
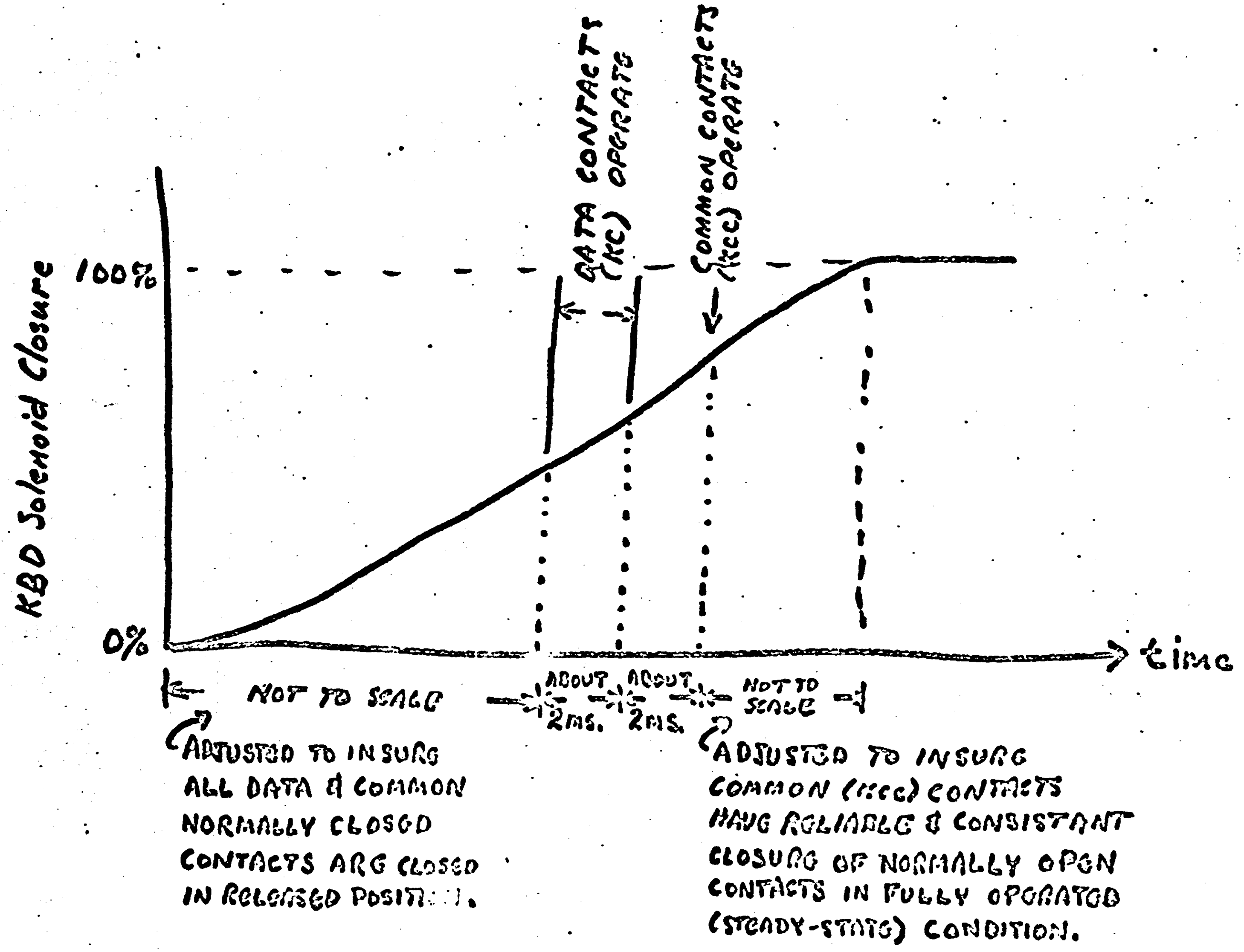


FIGURE 1



ADJUSTED TO INSURE ALL DATA & COMMON NORMALLY CLOSED CONTACTS ARE CLOSED IN RELEASED POSITION.

ADJUSTED TO INSURE COMMON (KCC) CONTACTS HAVE RELIABLE & CONSISTANT CLOSURE OF NORMALLY OPEN CONTACTS IN FULLY OPERATED (STEADY-STATE) CONDITION.

Figure 2 KBD Solenoid timing chart

Report of LINC use by the Clinical Laboratory, Dept. of Medicine

Programs have been written dealing with two kinds of data: laboratory test results and patient medical history data. The following is a summary of these programs.

LABORATORY

Programs have been developed which allow the direct input and manipulation of laboratory results through the keyboard. Laboratory results can be typed in as the decimal values familiar to the technologists, e.g. % for hematocrit, mg% for glucose, etc. After data is typed in, depressing the "S" key causes all data to be converted to octal values and stored on magnetic tape. Depressing the "P" key recalls the data and appropriate subroutines to sort the results into histogram bins. After sorting, the histogram display with appropriate legend comments is started.

In summary, three basic subroutines have actually been written for laboratory data:

1. Keyboard input for laboratory values
2. Decimal to octal conversion
3. Sort and search of randomly entered data

We are currently training one medical technologist in our clinical laboratory to program the LINC. One student who has had some programming experience on other systems spends part time helping with our laboratory computer activity.

Plans for the laboratory LINC activity in the immediate future include:

1. Development of programs to store laboratory data according to date (month, day, year) and to retrieve stored data in any time interval requested.

2. A master program to scale subroutines to handle a variety of laboratory data..

3. Modification of histogram routines supplied by Neurophysiology to allow complete operation from the keyboard, including termination and selection of other routines.

4. The integration of all of the programs for laboratory data with question and answer display programs so that the computer might be operated by a technologist not trained in programming.

MEDICAL HISTORY DATA

Programs have been written to display medical history questions to patients with the oscilloscope and accept the patient's response through the LINC keyboard. The first test program was written for allergy and the questions were designed by Dr. Warner Slack.

The Q & A subroutine of J. Cox was used for the basic display. A separate control program was written to present a standard response format along with each question so that only questions need to be stored on tape. The control program also returns to the appropriate location at the end of each question for instructions about the next display depending on the response of the patient. This control program will enable Dr. Slack to write medical history routines in the LINC without learning all aspects of LINC programming.

At the present time, the program to take medical histories is running. Immediate plans are to use the teletype output to generate medical history summaries after the patient has finished the examination.

FUTURE ACTIVITY

It is hoped that in April 1965 we can make firm plans toward the purchase or rental of computer equipment to be based in the clinical laboratory. The exact nature of these plans will depend upon our evaluations which will be based upon our test programming up to April.

Since many of the programs will eventually find application in the service functions of the medical center (taking histories, handling laboratory data), we do expect that our computing activity will eventually justify a laboratory based computer.

ACKNOWLEDGEMENT

We would like to acknowledge the generousity of the Department of Neurophysiology in making LINC computer time available to us. This time has allowed us to rapidly evaluate many aspects of our problems which would be impossible otherwise. This experience has had a very significant impact on our thinking about computers.

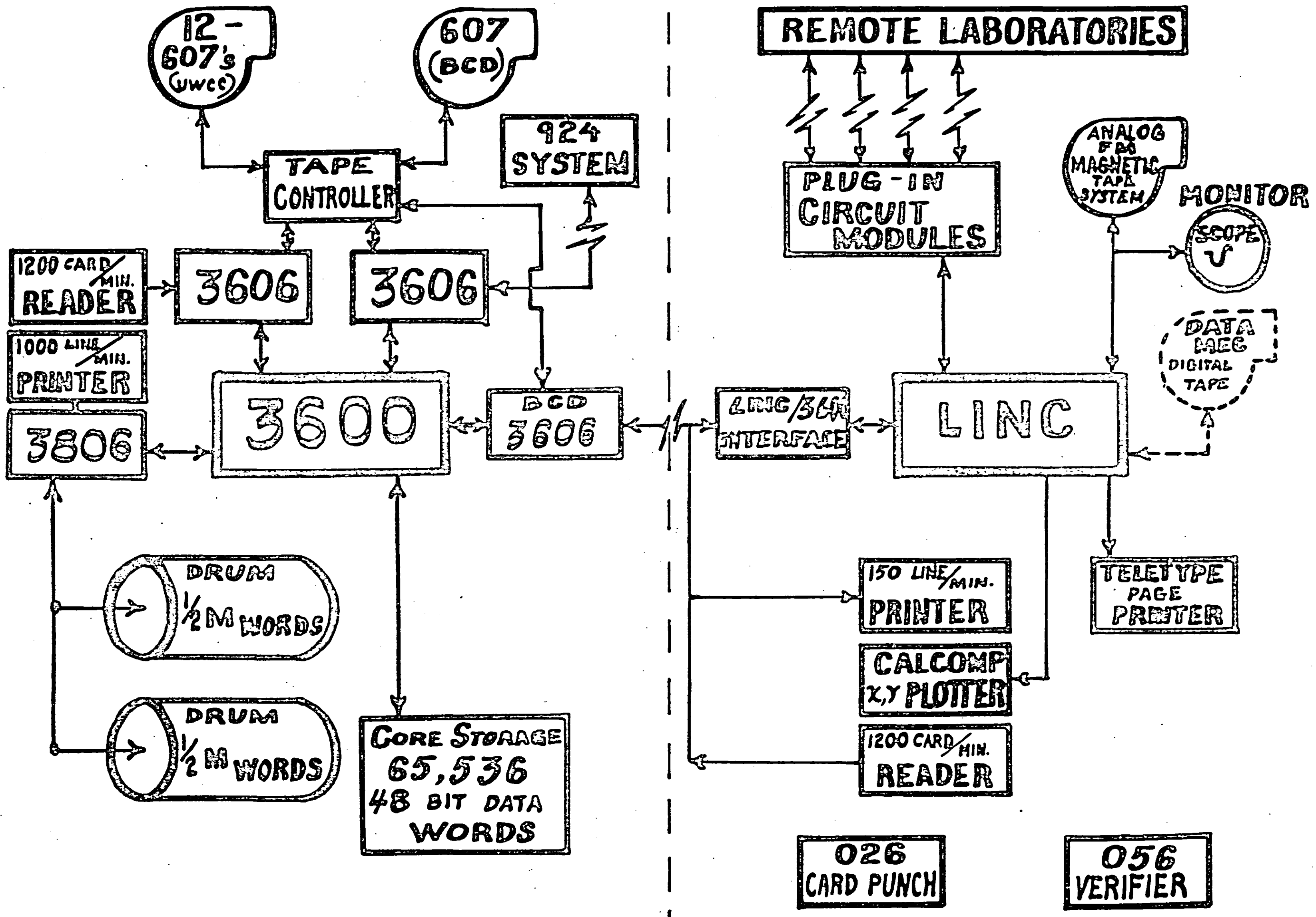
We would also like to acknowledge the many subroutines which have been made available to us and which largely account for the progress which we have been able to make.

G. Phillip Hicks, Ph. D.
Assistant Professor of Medicine
G Phillip Hicks

BLOCK DIAGRAM-UWCC/BCD PROPOSED EQUIPMENT

FACILITIES LOCATED AT THE UWCC
(STERLING HALL)

BCD FACILITIES LOCATED AT THE
DATA ACQUISITION LABORATORY
(MEDICAL SCIENCES BUILDING)



APPENDIX 4

Block Diagram - UWCC/BCD Proposed Equipment

APPENDIX 5

DRAFT

3600 - LINC INTERFACE

1.0 General

1.1 Three types of Data Transmission that will occur between the LINC Satellite and the 3600 computer are considered in this paper.

1.1.1 Mode 1 is a single fixed length block of data which will be processed or analized by a program in the 3600 with feedback provided to the LINC Printer or to the LINC computer at the Bio-Medical Computing Division (BCD) or with results stored on magnetic tape at the University of Wisconsin Computing Center (UWCC).

1.1.2 Mode 2 covers multi-fixed length blocks of data which would be stored on magnetic tape (or Drum) at the 3600 and processed or analyzed upon receipt of the last block of information (as priorities permitted). The results of this analysis to be transmitted to the [illegible] printer or to the LINC computer at BCD or stored on magnetic tape at the UWCC.

1.1.3 Mode 3, for which programming implementation would be delayed until after Modes 1 and 2, consists of variable length 12 bit word transmissions from the LINC to the 3600 for recording on magnetic tape. During this mode no other operations would take place on LINC-data channel; a twelve bit word would be transmitted every 24 microseconds. These tranmmissions might be for periods of 10 to 30 seconds initially with longer periods possible at still later date.

This latter mode (mode3). implies that an extended period of real-time operation would be required since partial retransmission or delays could not be accommodated. In addition, these data rates imply that about 624,000 - 48 bit words would be received each minute, thus a capability for changing output magnetic tapes would be required.

1.2 There are a number of general requirements (or assumptions) which are basic to this paper.

1.2.1 The 3600 will be the "master" computer and the LINC a "slave" for all transfers. The LINC, however, will have the capability to request 3600 action, and will have the capability of terminating a transfer that has been started should the need arise.

1.2.2 A control group of three 3600 words (12 LINC words) will always be transmitted as the first words of a block tramsfer. The control group format will be specified by the UWCC and will allow some area for user information. The first word (the first four LINC words) will include the number of words in the block and an indication as to whether another block follows. Bit 11 of the first LINC word, (bit 47 of the first 3600 word) will be a one if at least one more block follows this block. (This is the chaining bit position in the I/o control word for the 3600). Bits 24 through 32 indicate the number of 3600 words to be transmitted (the number of LINC words modulo 4), bit position 24 is the least significant. (This word count will occupy positions 0 through 8 of the second LINC word transmitted -- the choice of location is based upon the word count bit locations in the 3600 input - output control word.) The value of this count can vary from 3 to 703_8, but most transfers will be for 703_8 words (i.e. 3 control words and 448 data words) It is assumed that the 3600 will modify its I/o order to reflect the correct number of words after the first word is in the 3600.

2.0 Sequence of Events LINC to 3600 transfers (mode 1 & 2)

2.1 The 3600 Monitor program must be operating in the 3600, and the mask in the "Synchronizer and Universal Transceiver for University Research Experiments" (SUTURE) box at the LINC must be set to allow interrupt on "LINC Transmit" (status bit ~~4~~ 3 see chart).

2.2 The LINC loads its A register with the word count (value of the count in paragraph 1.2.2 times four), and sets the proper bits in location ~~zero~~ one to perform a specific LINC operate instruction (hereafter called a LINC function instruction). This specific function instruction will Set status bit ~~4~~ 3 in the SUTURE box, if the 3600 is running, put the A register into the count register, and indicate to the LINC whether the 3600 is running. If the 3600 is running, the LINC then loads A with the starting address in LINC memory to be used for the transfer (normally 364_8) and execute a 'Gulp' instruction (to enable the Gulp mode). The SUTURE box will cause loading of A through B into the memory address counter (S) and then wait, if necessary, for the 3600 read instruction. (If no operation takes place for an extended period, LINC would leave the Gulp mode and set STATUS bit 4.)

2.3 The 3600 would service the interrupt, determine that the LINC desired a "read" order. The 3600 would clear Status bit 4 in the SUTURE box, and issue a connect order (which would be accepted by the SUTURE box hardware). The 3600 would enable interrupt on Status bit 2 or 4 and issue a read instruction (normally for 703_8 words) which would enable timing to set the "MOUT" level in the SUTURE box causing an output gulp mode to take place.

2.4 Twelve bit words would be transferred automatically between the LINC and 3606 data channel, as governed by the "Data" and "reply" signals, until an "end" condition appeared.

(a) Normal End would be: (LINC count =0) and (3600 count = 0), the SUTURE box sets STATUS bit 2 to indicate normal end. If the 3606 sends another data signal after the SUTURE count = 0, the SUTURE will return "end of record" instead of "reply".

(b) Premature end by 3600 would be: the 3606 drops its read line (due to parity error received or due to some other malfunction), the LINC box must terminate its gulpmode and set status bit 4 (error determined by LINC)

(c) Premature end by LINC Would be: The SUTURE count = 0 thus LINC sends "end of record" on the next data signal. LINC also gets status bit 2 in the SUTURE box. At the 3606 the count would not equal zero.

2.5 If the end is normal from the LINC standpoint then status bit 2 setting will cause a 3600 interrupt. If the end is abnormal from the LINC standpoint status bit 4 setting will cause the 3600 interrupt.

2.6 The 3600 program will check the word count after the interrupt and check the status register to determine which end condition exists.

(a) NORMAL: (count = 0) (status bit 2 = 1) (no parity error)

(b) Prematurely ended by 3600: (Status bit 4 = 1)

(c) Error detected by 3600: (status bit 2 = 1) but (word count = 0) or (parity error = 1).

The 3600 must clear status bits 2 and/or 4 and, in addition, it must set status bit 0 if condition 'C' exists. In case of condition (b) or (c) the LINC is

12-26

expected to retransmit the same block again (or seek operator action).

2.7 When the LINC notes status bit 2 is clear, it must check status bit 0 as well. If status bit 0 equals 0, then LINC can proceed to next block transfer; if status bit 0 = 1, LINC must retransmit the block (or seek operator action). In either case step 2.2 is the next action required by LINC.

2.8 When 3600 has received a good block (condition 2.6 (a)) it must decide what further action is required (i.e. store on mag tape, store on drum, call in user's program, etc.)

3.0 Sequence of Events 3600 to LINC transfers (modes 1 & 2)

3.1. If the 3600 desires to transmit to the LINC, the 3600 sets status bit 2 in the SUTURE box and masks on status bit 2 (LINC ready) to wait for an interrupt.

3.2. The LINC checks for status bit 1 with a function instruction. If bit 1 = 1, then LINC loads 3414 into A, and issues a function instruction which will put A into the count register, clear status bit 1 and set status bit 2 (LINC ready) LINC then puts the starting location (364_8) into A and executes a gulp instruction (to enable the gulp mode). Gulp mode puts the A value into the memory address counter (via B) and waits, if necessary, for a write instruction from the 3600.

3.3 When the 3600 interrupts on status bit 2, it clears status bit 2 and enables interrupt on status bits 2 and 4 (normal or error ending) and on end of chain (in the 3606). The 3600 then issues a write instruction (which enables timing to set "HINP" in the SUTURE box and makes the LINC gulp instruction an input to the LINC type of transfer).

3.4 "Data" and "Reply" control the transferring of 12 bit words between the 3606 and SUTURE box until and end condition:

(a) Normal End would be: (SUTURE count = 0) and (3600 count = 0); the SUTURE box sets status bit 2 when its count is zero and leaves the gulp mode.

(b) LINC prematurely ends transfer would be: (SUTURE count = 0) but (3600 count ≠ 0); the SUTURE would set status bit 2 when its count equalled zero and leave gulp mode. Any future data signals from the 3606 would receive "end of record" instead of "reply". (This assumes 3600 looks for "end of record" on WRITE, an assumption which has not been checked).

(c) 3600 prematurely ending would be: the write line dropped (either by failure or by 3600 program if an error was detected). SUTURE senses write line dropping with count = 0; SUTURE sets status bit 4 (error detected by LINC) and leaves GULP mode.

3.5 When the 3600 interrupted by status bit 2, if the count = 0 and no transmission parity errors were detected, the block transfer was correct status bit 2 can be cleared and the next block or requirement can be handled by the 3600. If status bit 2 was a one but the count is incorrect at the 3600, or if parity errors were reported, the 3600 sets status bit 0, clears status bit 2, and prepares for retransmission when it notes that status bit one was cleared by the LINC. If status bit 4 is a one, 3600 must clear it and also prepare for retransmission (return to step 3.1). In case of excessive retries operator action may be required.

3.6 The LINC notes whether status 4 was set by the SUTURE box or if status 0 ~~on if status one~~ is set by the 3600 when status 2 was cleared. If either condition exists, the LINC must prepare to receive the block again. The LINC is responsible for reset of status bit ~~one~~ zero if it was set.

4.0 Sequence of Events (Mode 3 transfer)

Details to be provided later.

INITIAL DISTRIBUTION:

BAUMHOFF, L.
BIRKELO, D.
DAHLKE, R.
GEISLER, D.
HIND, J.
KEENAN, J.
RAYFIELD, W.

STATUS – INTERRUPT – MASK ASSIGNMENTS

3600 STATUS LINE POSITION	LINC STATUS LINE POSITION	Description of Signal	SET BY	CLEARED BY	3600 INTERRUPT Register Position	3600 MASK Register Position	Remarks
0	6	Error Detected by 3600	3600 Pgm	LINC Pgm	0	0	
1	7	3600 TRANSMIT	3600 Pgm	LINC Pgm	1	1	
2	8	End of operation/LINC Ready	LINC Pgm	3600 Pgm	2	2	
3	9	LINC Transmit	LINC Pgm	3600 Pgm	3	3	
4	10	Error Detected by SUTURE box	SUTURE Hardware	3600 Pgm	4	4	Loss of Read or Write but COUNT≠0
5	—	AVAILABLE BUT NOT CONNECTED	—	—	—	—	
6	—	} Not presently Available	—	—	—	—	
7	—		—	—	—	—	
8	—	LINC RUNNING	SUTURE Hardware	SUTURE Hardware	8*	8*	THIS SIGNAL MAY BE INVERTED AT 3606 – IT IS NOT KNOWN
9	—	LINC Read	SUTURE Hardware	SUTURE Hardware	9*	9*	Generated from 3600 write
10	—	LINC Write	SUTURE Hardware	SUTURE Hardware	9*	9*	Generated from 3600 Read
11	—	LINC Parity Error	(tied off)	(tied off)	8*	8*	No parity used in LINC
—	11	3600 RUNNING	3600 Hardware	3600 Hardware	—	—	

* TWO SIGNALS ON SAME POSITION

APPENDIX 6

DRAFT

The LINC Function Instruction

1.0 General

1.1 This instruction is an operate instruction performed by the LINC, hereafter referred to as the FUNCTION instruction. This instruction has been designed to perform under program control certain functions external to the LINC such as: reading the status of external counters or flip-flops, loading of external counters, and/or the setting of and clearing of external flip-flops. The 12 bit word in core memory, location one, hereafter called the control word, and the 12 bits of information in the A register are used to give detailed logical instructions to the FUNCTION instruction. The execution time is 24 u sec unless bit 3 in the control word is zero when a 3600 interface type FUNCTION instruction executed. In the latter case, which indicates that an external count register is to be read into the A register, requires 32 u sec for execution.

1.2 On all function instructions, the condition (one or zero) of the appropriate external status lines prior to execution of the instruction will cause complementing of the corresponding high order bits in the control word if the external status condition is a one. This complementing will change the configuration of the control word in memory. It is expected that this 'reading' of external status conditions will always immediately follow~~ing~~ a 'gulp' mode type operate instruction.

2.0. Procedure

2.1 The general procedure for performing a FUNCTION INSTRUCTION IS:

(a) Load core memory location one with the proper control word

(b) Load the A register with a count, if required

(c) execute the function instruction

(d) examine(or ~~store~~ Move from location two) the A register if the count register has been read

(e) examine control word to determine the condition of external status lines, if required.

2.2 Figure one shows the format for a control word to work with the LINC/3600 interface (SUTURE box). If the lowest order (zero) bit is set, the instruction will cause clearing of designated (see bits 6 through 11). status flip-flops; if the lowest order bit is zero no clearing will take place. If the next most significant bit (the one bit) is a one (i.e. is set), the instruction will cause setting of designated flip-flops; if bit one is a zero, the flip-flops will not be set. If the next higher bit, bit 2, is set then the value in the A register will be transferred to the external count register; if it is zero the 'count' will be read into the A register and into core memory location two.

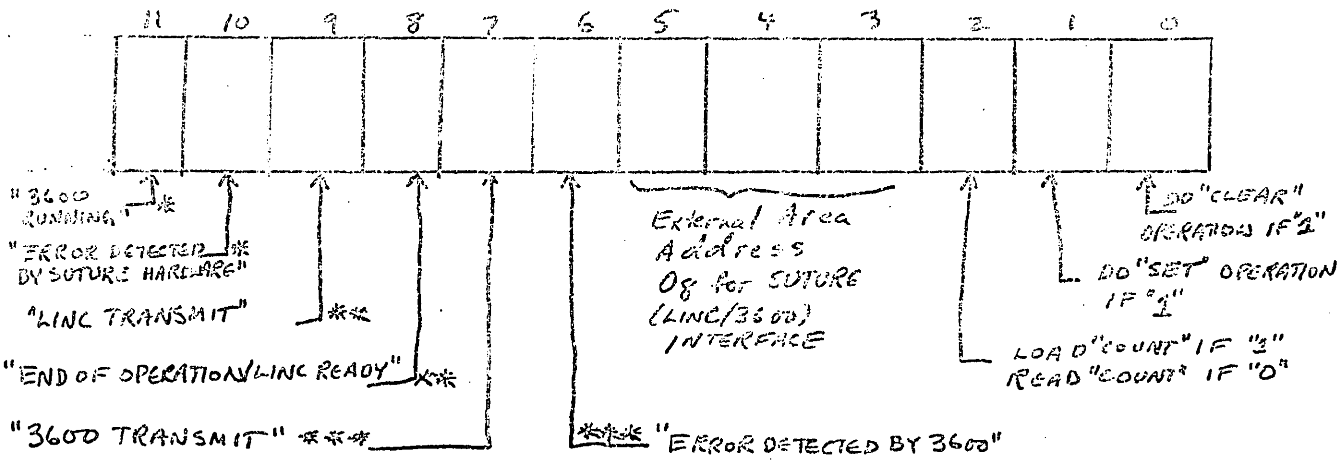
2.3 Bits 3, 4, and 5 in the control word designate the external area to be operated on by the **FUNCTION** instruction. When bits 3,4, and 5 are all zero, the **SUTURE** box (3600/LINC interface) is specified. The high order bits 6 through 11 indicate the condition of status lines and the status lines which are to be altered depending upon the setting of bits 0 and 1. The status lines which are a one always complement their corresponding bit in the control word (thus if bits 0, 1, and 6 through 11 are all zero, the [illegible] will be "read" into the control word. Since none of the status flip-flops can be both set and reset by the LINC, both setting and resulting (clearing) can take place with the same instruction. Bits 6 and 7 can only be reset, bits 8 and 9 can only be set, and bits 10 and 11 can neitherb be set nor reset, only 'read'. Since the reading of status is a complimenting act on, certain bits in locations 6 through 11 could be 'masked off' by setting them as a 'one' in the control word if it was expected that they would be in the 'one' state when read. A setting of zero when the status line was expected to be a zero would also achieve the same effect. The meaning of bits 6 through 11 are found on figure one.

2.4 In the A register is placed the values to be placed in the external "count" register. Normally this value would be 3414_8 for a block transfer. When the count register is "read" the count number of the next word which would have been transferred appears in the A register and in core location two.

3.0 Mechanization

3.1 Technically the mechanization of the function instruction is shown in

Figure [illegible] is a timing diagram. When the control decode lines indicate (at [illegible] time) that a FUNCTION instruction is called for, a flip-flops is set (using the leading edge of this level). This flip-flops enables 'MOUT' and 'MLF' to be generated. The former causes readout(in CY 2 time number ~~two~~) of core location one into B; the latter is used to prevent a 'pause' if the "I" bit has been set in the instruction. A second flip-flop signals that the second CY2 is beginning and the operations specified by the control word are to be commenced. If bit 2 of the control word is a one, only the first flip-flop will ~~reset~~ be reset, this will extend the FUNCTION instruction into a third CY2 and cause "count" to be read into A and core location two. Note that if bit 2 is a one, the value of A after the instruction has been completed represents the value placed in ~~count~~ "A" by the program (not by "count"), bit complemented by the contents of location one (the control word) as location one ~~it~~ appears after the instruction.



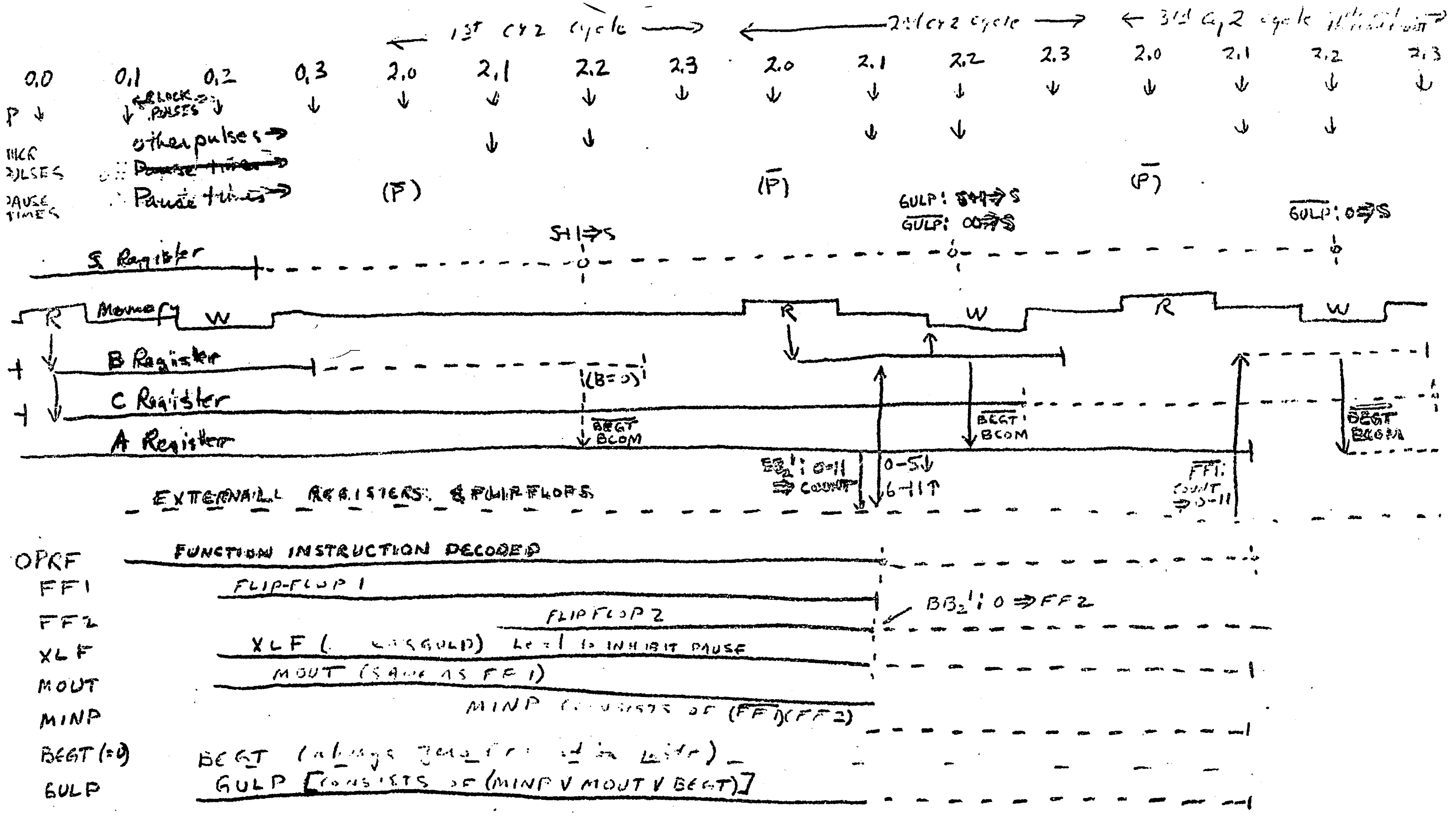
* These signals can only be read, not set or reset.

** These signals can be set if they are a 'one' and bit one is also a 'one'; these cannot be reset.

*** These signals can be cleared (reset) if they are a 'one' and bit zero is also a 'one'; these cannot be set.

FIGURE ONE

CONTROL WORD FORMAT — FUNCTION INSTRUCTION (for 3600/LINC-SUTURE-interface)



Notes

$(\overline{FF1})(OPRF);\ 1 \Rightarrow FF1$

$(FF2)(2,1)\ ;\ 0 \Rightarrow FF1$

$(FF1)(2,1)(\overline{FF2});\ 1 \Rightarrow FF2$

$(2,1)[(FF1)(BB_2^1)+(\overline{FF1})(BB_2^0)](FF2);\ 0 \Rightarrow FF2$

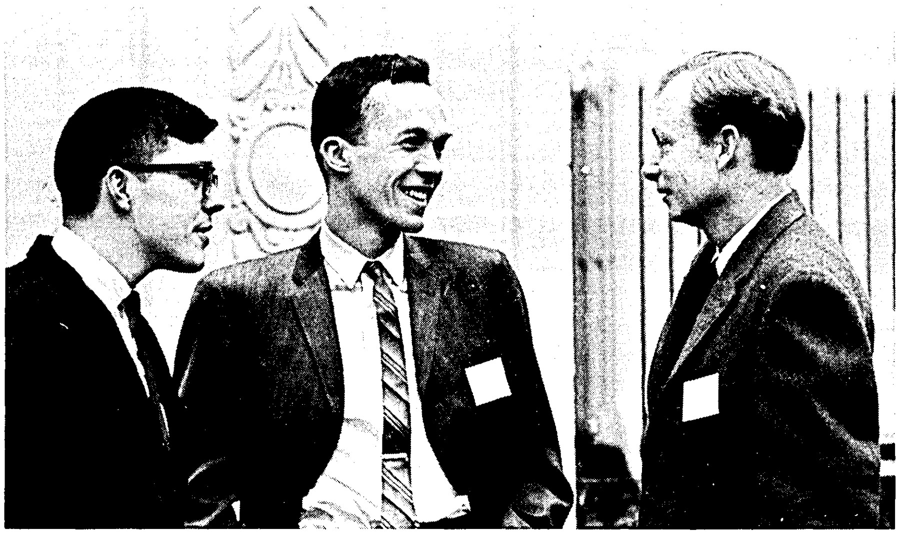
+ = V = INCLUSIVE OR

R = Read

W = Write

FIGURE TWO

FUNCTION INSTRUCTION TIMING



FINAL REPORT

LINC Evaluation Program

J. Lederberg

L. Hundley

Department of Genetics
Stanford University

March 1965

13-1

CONTENTS

I. Introduction

II. General Use I-O Equipment and Programs

III. Utility Programs

IV. Experiment-Related Programs and Hardware

V. The LINC Evaluation Program as a Training Technique

VI. Computer Performance

VII. Conclusion

Appendix A: Selected Programs

13-2

I. INTRODUCTION

The Instrumentation Research Laboratory within the Department of Genetics has as its purpose the design of special purpose instruments for biological research. This includes electrical, mechanical and optical design. The LINC in our laboratory has been used as a system element in a number of experimental situations and its use has proven to be both education to us and experimentally rewarding.

Headed by Dr. Joshua Lederberg and under the direction of Dr. Elliot Levinthal, the laboratory has as its primary mission the development of life detection systems on a microbial level for remote Martian exploration. In order to accomplish this end, a number of different types of physical measurements have been investigated in great detail. We believe that these studies, a number of which involve LINC, will also result in new instrumentation and techniques of general laboratory utility.

We wish to request that the LINC be permanently assigned to our laboratory.

II. GENERAL USAGE I-O EQUIPMENT AND PROGRAMS

Our LINC has been equipped with a number of peripheral devices. These include a Datamec IBM compatible tape recorder, a Calcomp plotter, and a teletype. In the process of being installed is a 4096 word external memory.

The Datamec is equipped for two speed (45 and 4.5 ips) two density (200 and 566 bpi) operation, with both read and write capability. These speeds and densities give us a wide range of data rates. The upper limit is 25,000 six bit characters per second. The interface is very simple and required only two cards. One of these would be eliminated if the gated accumulator lines were being used for nothing else.

Programs for the Datamec include those to read and write IBM compatible format, generate data tapes from continuous on-line input, and to regroup the input data on LINC tape blocks and then, if desired, to rewrite these blocks into IBM format. All of these combinations form a highly flexible system. Use of the Datamec has completely superseded the IBM 026 keypunch.

The Calcomp plotter has been in operation for some time now and has proven to be extremely useful. Programs for plotting all forms of data have been written. These include both ordinate and abscissa scaling and linear interpolation. A program has also been written for character generation which includes character size scaling and positioning.

The teletype has proven to be a very good means of getting both program and data into LINC and getting hard copy of both out. Its major drawbacks are its low speed, lack of tabs and that it is somewhat noisy; however, we know of no cheaper means of getting printed output. Input and output routines have been written which calculate teletype code from LAP code and viseversa which take about twenty locations each, so memory usage is not excessive.

The 4096 word memory, which should be in operation within the next few weeks, will be used both for program and data handling. There will be three modes of operation which are: 256 word input and output gulps at eight microseconds per word and single word input which indexes the memory address register with each input. This later mode is designed mainly for data handling.

III. UTILITY PROGRAMS

These programs include those for program input, assembly, and debugging, for keyboard data input and computation and for data display. Most of the programs to be mentioned are more completely described in Appendix A.

The LINCT system is our teletype program text input-output system which has a number of useful features. It is tied into a modified LAP which will assemble for the 2K memory.

We have operating on the IBM 7090 a compiler for LINC which uses a modified Balgol language. This system, called "BLINC", and a program operating system which was written in BLINC are described in some detail in the appendix.

Debugging routines include an octal to Mnemonic converter and print-out program, and a program which follows another program through all of its branching to determine which locations contain instructions and which contain constants. This is used with the converter program to get a proper print-out. A print-out of LAP III was obtained in this way.

A Floating point package with two word mantissa has been written. Copies of this program and a usage explanation will be available shortly. This program has been incorporated into a desk calculator with storage routine. This routine has the usual arithmetic operations as well as square root, e^{X} , $\log_{e} X$, sin X , cos X , and 2 X 2 Chi square. It is arranged for the easy addition of other arithmetic subroutines. Teletype input and output and certain manipulations of the stored data are included.

A number of simple algebraic programs have been written, such as those for mean and standard deviation, Chi square and other statistical operations.

Display programs include those for point and bar graph display with X and λ scaling keyboard calling of data sets. These data sets may be manipulated in a number of ways including inversion, addition, multiplication and rotation.

These are the major programs of a general usage nature now in operation. The only major programing effort now being considered in this class is a simple arithmetic compiler based on the two word floating point system. A more complete symbolic compiler is a possibility, but due to the large amount of effort involved will probably not be undertaken for some time.

IV. EXPERIMENT RELATED PROGRAMS AND HARDWARE

Most of the research in our department is involved with experimentation either on a bacterial or molecular level; therefore all of the on-line LINC experiments that have been done have involved physical methods such as mass spectroscopy, radioactive and fluorescent tagging, fluorescent decay times and particle counting. An anticipated experiment involves the interpretation of Raman spectra.

The LINC has been directly connected to the output of the Bendix time-of-flight mass spectrometer. Output from the mass spectrometer is reduced as it comes into LINC into mass amplitude and time of occurrence. The direct determination of mass number is difficult due to instability in the Bendix's scanning ramp. One means of overcoming this, which will be tried, is to allow LINC to generate the scanning ramp by the use of a mechanical D-A converter which has been built in our shop. This consists of a 200 step per revolution stepping motor driving a ten turn pot. This is a very simple system and has proven most useful. This use of LINC ties in with a much larger system which is a computer program for the direct determination of compound composition from mass spectra. This work is being done under a separate grant and the initial program is being run on the IBM 7090 at the Stanford Computation Center.

The LINC has been used in a number of ways in experiments with fluorescent compounds. The first experiment of this type used LINC as modulator, phase locked detector, and integrator in an extremely sensitive fluorometer. With integration times of ten minutes, the detection of 10^{-13}

13-6

molar solutions of fluorescein with a signal to noise of 15 to 1 were obtained using a 400 milliwatt light source. This experiment was performed to determine parameters for a sensitive fluoremeter as part of our effort to design apparatus for the detection of life on Mars. A program is now being written which will determine the best fluorescent system transfer function for a given material by generating all possible combinations of filters, light sources, and phototubes. The data for the components of this system will be stored as sets on LINC tape.

A system has been built for the determination of fluorescent decay times in the low nanosecond region. This consists of a fast flash lamp, photomultiplier tube, sampling scope and LINC as a 512 channel integrator. Calculation shows that we will get about two quanta per channel per flash. Our design goal is to investigate materials with decay times on the order of five nanoseconds. To date, our best results have been in the 10 nanosecond region. The limiting factor is the lamp decay time. This will be improved by the use of a different type of lamp. The LINC has performed most admirably in this application. No external hardware was required except the mechanical D-A convertor for driving the sampling scope sweep. This experiment is being conducted in cooperation with Dr. Lubert Stryer of the Stanford Biochemistry Department. A program will be written to get a best exponential fit to the experimental data so that direct time constant output will be available.

Programs have been written for the keyboard input of data from nuclear counters which determine man and standard deviations as well as sorting data sets according to size distributions and normalizing the data. These programs have been in routine use by a group in the Genetics Department under the direction of Dr. Leonard Herzenberg. This group is studying antibody reactions in mice.

These programs have, by the rapid presentation of results, allowed the experimenters to determine what the next step in their procedure should be with very little delay, and has therefore increased the number of experiments which they are able to perform by a factor of two to three.

V. THE LINC EVALUATION PROGRAM AS A TRAINING TECHNIQUE.

In general, the experience gained with digital techniques has been of great value to all of us here. The instruction initially received on LINC was quite adequate with one exception. It would have been very desirable to spend more time on use and misuse of the various I-O functions. It has been in this area that most of our nonproductive time has been spent. From the overall point of view, LINC has been a most demanding teacher in its own right. It has changed and simplified our approach to many problems. It has also made possible experiments which would otherwise have been too time consuming to perform.

Several undergraduate and medical students have gained proficiency in systems programing on LINC. It is an excellent machine from the standpoint of man-machine interaction but higher level languages would give a more realistic interaction to sophisticated systems.

VI. COMPUTER PERFORMANCE

The performance of LINC in respect to maintenance has far exceeded reasonable expectation. After approximately 3200 hours of operation, the only failures have been one bad cable connection and two output transistors whose failure can be traced to external misuse.

The general performance of LINC in the laboratory has been entirely adequate and most rewarding. Most of the recommendations that come to mind must be admitted to be generated by our own special requirements; however, there are three recommendations which it is felt are of general interest to most users.

The first area is that of multiple word arithmetic. Any instruction changes which would reduce program length and running time would be a great help. These might include clearing the accumulater on a LAM instruction and recovery of both halves of a multiply.

The second suggestion is to make all of the 2K memory programable. This would be very useful when performing complex computations and would reduce the running time of a number of programs which we now operate by minimizing the number of tape transfers involved. A suggested means of achieving this is being transmitted under separate cover to S.M. Orinsten at the Computor Research Laboratory.

Our third point is that a problem-oriented compiler (e.g. artran or Atyol) would be extremely useful. Even if the compilation were somewhat slow, the reduction in programing time should still be very large. Mnemonic print-outs of the compiled program can allow the programer to see exactly what is happening and give him a framework in which to get machine code zonations.

VII. CONCLUSIONS

The concept of what an ideal laboratory computer should be will vary greatly among various investigators. From our point of view, LINC has proven to be a very useful system. The careful attention of the designers to those points which are most important for the on-line use of a computer is obvious and most gratifying.

It has become apparent that in the future we will want to have on-line computer capability even greater than that provided by LINC. Greater word length, higher A-D resolution, larger memory, greater speed and smaller physical size will be the types of improvements that we will be looking for in new machines. A system such as IBM's 1800 is a step in the right direction. This desire for a larger capability has certainly been the result of the use of LINC itself. We feel that future developments must proceed in this direction if full advantage is to be taken of the experience gained from the LINC program.

Appendix: A

Selected Program Discriptions

13-10

Double Precision Floating Point

Winter 1965

t. coburn

13-11

General Information

1. A double precision floating point word consists of three 12-bit words in the following sequence: exponent, high order word, low order word. The last two of these are collectively called the "mantissa".
2. Exponent and mantissa each contain a sign in the leftmost bit, i.e. the 11 bit of the exponent or 23 bit of the mantissa.
3. The mantissa is a fraction between +1 and -1; that is, the decimal point is assumed to be at the left of bit 22.
4. The mantissa is left adjusted. This means that except for zero words, all positive mantissas will contain a 1 in bit 22, and all negative words will contain a zero in bit 22.
5. Integers can and indeed must be used for some of the routines available. These are automatically floated before they are used.
6. A floating point accumulator(FAC) is maintained in locations 112o, 1121, and 1122. It is used in the same way that the regular accumulator is used.
7. The other half of any operation is called the operand or argument.
8. The address of a double precision floating point word is the location of the exponent. Integers are addressed as usual.
9. The floating point routines useindex registers 12-17. These registers are not restored on leaving the floating point package.
10. Entrance to the floating point package is accomplished by jumping to a three instruction routine located some place in core (see next page).
11. After the last operation code the program exits and contiues executing regular Linc instructions.
12. There is no rounding off within the floating point package.

Instructions for using the package

The following sequence of instructions will serve as an example of the necessary format.

```
176   :
177   :
200   Jmp 375 -----------------------------------375      Lda
201   0400    (operand address)                   376       0
202   4001    (operation code)                    377      Jmp 1000
203   0403    (operand address)
204   4002    (operation code)
205   0400    (operand address)
206   0023    (operation code)
207    :
210    :
```

Operand Address

This may be a direct address: 400
or an indirect address: 4002
or it may be zero.

1. In a direct address the location, 400, contains the exponent of the floating point word, or an integer as the case may be.
2. In an indirect address the index register, 2, refers to the corresponding address. Bit 12, the 4000, bit signifies that the address is indirect.
3. A zero operand refers to the floating point accumulator. Hence, to square a number in the FAC, one executes a multiply specifying a zero operand.

Operations

1. Operations available are listed in the following table.
2. The 4000 bit in the operation code is used to indicate whether this operation is the last in a series. In the example above, if the next location following the code 0023 contained 0400, this would be interpreted as "sxl". If the last code had been 4023, then the next location would be the address of an operand.
3. Some operations, fix and sign, are meaningless unless they are the last in a series since the result is left in the regular accumulator.

Table of codes and operations

Code		Operation
1	Cla	Clear and add operand to FAC.
2	Add	Add a floating point word to FAC.
3	Com	Complement operand; leave in FAC.
4.	Mul	Multiply FAC times floating point word.
5	FAC/OP	Divide Fac by floating point word.
6	OP/FAC	Divide operand by FAC, result in FAC.
7	I+FAC	Add an integer to FAC
10	IxFAC	Multiply FAC by an integer.
11	FAC/I	Divide FAC by an integer.
12	I/FAC	Divide integer by FAC, result in FAC.
13	Fix	Convert a floating point word to an integer. Result is left in regular accumulator.
14	Flt	Float an integer, result in FAC.
15	Clr	Zero put in operand and FAC.
16	Max	Compare size of operand with FAC. Larger left in FAC.
17	Min	" " " Smaller " " .
20	SGn	If operand is less than zero, -1 is left in regular acc. If operand equals zero, 0 is left in regular acc. If operand is greater than zero, +1 is left in reg. acc.
21	incr	Increment operand by FAC, leave in Fac as well. This is equivalent to an add to memory.
22	Sub	Subtract operand from FAC. Result left in FAC.
23	Sto	Store FAC in address of operand. Leave in FAC.
24	SSP	Set sign of operand plus; i.e. complement if negative.
25	SSM	Set sign of operand minus; " " " " positive.

```
1000 ADA↑
1001 1776
1002 STC 17
1003 SET↑13
1004 1121
1005 SET↑14
1006 1122
1007 SET↑16
1010 1125
1011 SET↑15
1012 1124
1013 LDA↑17
1014 AZE
1015 JMP 1020
1016 ~~JMP 1013~~  LDA↑
1017 ~~JMP 1037~~  1120
1020 APO↑
1021 JMP 1026
1022 BCO↑
1023 6000
1024 STC 1025
1025 HLT
1026 STA↑
1027 0
1030 STC 12
1031 LDA 12
1032 STC 1123
1033 LDA↑12
1034 STC 1124
1035 LDA↑12
1036 STC 1125
1037 CLR
1040 LDA↑17
1041 BCL↑
1042 4000
1043 ADA↑
1044 1072
1045 STC 12
1046 LDA 12
1047 STC 1050
```

Pick up operand and operation

```
1050 HLT
```

Jump to specified operation

```
1051 STC 1062
1052 LDA 17
1053 APO
1054 JMP 1013
1055 LDA↑
1056 6001
1057 ADD 17
1060 STC 1063
1061 LDA↑
1062 0
1063 HLT
```

Check for return

```
1064 JMP 1450
1065 JMP 1441
1066 JMP 1051
```

Float Arg and return

```
1067 JMP 1450
1070 JMP 1321
1071 JMP 1441
1072 JMP 1051
```

Integer Arg / FAC

```
1073 JMP 1071  1)
1074 JMP 1177  2)
1075 JMP 1543  3)
1076 JMP 1252  4)
1077 JMP 1525  5)
```

Jump table

```
1100 JMP 1321  6)  }
1101 JMP 1531  7)  }
1102 JMP 1534 10)  }
1103 JMP 1536 11)  }
1104 JMP 1067 12)  }
1105 JMP 1135 13)  }
1106 JMP 1064 14)  }  Jump table
1107 JMP 1653 15)  }
1110 JMP 1571 16)  }
1111 JMP 1546 17)  }
1112 JMP 1555 20)  }
1113 JMP 1133 21)  }
1114 JMP 1774 22)  }
1115 JMP 1137 23)  }
1116 JMP 1571 24)  }
1117 JMP 1574 25)  }
1120 0   }
1121 0   }  Location of floating point accumutator (FAC).
1122 0   }
1123 0   }
1124 0   }  Location of register A, or Argument (ARG).
1125 0   }
1126 0   }
1127 0   }  Location of register B.
1130 0   }
1131 0   }  Register Q, used in divide.
1132 0   }
1133 JMP 1177  }  increment
1134 JMP 1137  }
1135 JMP 1441  }  Fix
1136 JMP 1474  }
1137 SET 12    }
1140 1027      }
1141 LDA       }
1142 1120      }
1143 STA ,12   }  Store and return
1144 LDA 13    }
1145 STAi12    }
1146 LDA 14    }
1147 STAi12    }
1150 JMP 1051  }
1151 SET 12    }
1152 0         }
1153 LDA       }
1154 1120      }
1155 COM       }
1156 ADD 1123  }
1157 AZEi      }
1160 JMP 1166  }
1161 APOi      }
1162 XSKi12    }
1163 JMP 12    }  Which is larger? FAC or Arg.
1164 SET 12    }
1165 0         }
1166 LDA 13    }
1167 COM       }
1170 ADA 15    }
1171 AZE       }
1172 JMP 1161  }
1173 LDA 14    }
1174 COM       }
1175 ADA 16    }
1176 JMP 1161  }
1177 LDA       }  Begin Add
```

```
1201 STC 1241
1202 JMP 1424
1203 JMP 1600
1204 STC 1231
1205 ADD 1120
1206 COM
1207 ADD 1123
1210 AZE↑
1211 JMP 1227
1212 APO↑
1213 JMP 1221
1214 STC 12
1215 JMP 1604
1216 JMP 1613
1217 JMP 1622
1220 JMP 1223
1221 COM
1222 STC 12
1223 JMP 1633
1224 XSK↑12
1225 JMP 1223
1226 NOP
1227 JMP 1666
1230 SRO↑
1231 0
1232 JMP 1242
1233 JMP 1600
1234 APO↑
1235 JMP 1241
1236 JMP 1633
1237 JMP 1735
1240 NOP
1241 HLT
1242 JMP 1700
1243 JMP 1246
1244 JMP 1751
1245 JMP 1241
1246 JMP 1712
1247 JMP 1241
1250 JMP 1721
1251 JMP 1246
```

Add Arg to FAC

```
1252 JMP 1600
1253 STC 1316
1254 ADD 1123
1255 ADD 1120
1256 STC 1120
1257 JMP 1756
1260 JMP 1432
1261 JMP 1751
1262 SET↑12
1263 7763
1264 CLR
1265 SRO
1266 1130
1267 JMP 1666
1270 LDA 13
1271 ROR↑1
1272 JMP 1641
1273 XSK↑12
1274 JMP 1264
1275 SET↑12
1276 7764
1277 CLR
```

Multiply Arg x FAC

Multiply (continued)

```
1300 SRO
1301 1127
1302 JMP 1666
1303 LDA 13
1304 FOR↑ 1
1305 JMP 1641
1306 XSK↑ 12
1307 JMP 1277
1310 ADD 1126
1311 STC 1120
1312 JMP 1712
1313 JMP 1315
1314 JMP 1721
1315 SRO↑
1316 0
1317 JMP 1742
1320 JMP 1051
```

Divide Arg by FAC

```
1321 JMP 1600
1322 STC 1421
1323 JMP 1756
1324 JMP 1700
1325 JMP 1330
1326 JMP 1751
1327 JMP 1051
1330 JMP 1164
1331 JMP 1340
1332 JMP 1432
1333 JMP 1441
1334 JMP 1633
1335 NOP
1336 JMP 1613
1337 JMP 1341
1340 JMP 1432
1341 ADD 1126
1342 COM
1343 ADD 1123
1344 ADD 1665
1345 STC 1126
1346 STC 1131
1347 STC 1132
1350 SET↑ 12
1351 1131
1352 JMP 1622
1353 JMP 1742
1354 JMP 1356
1355 JMP 1622
1356 JMP 1666
1357 JMP 1666
1360 JMP 1613
1361 LDA 15
1362 APO↑
1363 JMP 1366
1364 JMP 1700
1365 JMP 1376
1366 LDA 12
1367 BCO↑
1370 4000
1371 STA 12
1372 SRO
1373 1370
1374 JMP 1402
1375 JMP 1352
1376 SRO
```

```
1400 JMP 1402
1401 JMP 1355
1402 XSK↑12
1403 SRO↑
1404 2525
1405 JMP 0
1406 CLR
1407 ADD 1132
1410 STC 1122      Divide (continued)
1411 ADD 1131
1412 STA 13
1413 APO↑
1414 JMP 1420
1415 JMP 1633
1416 JMP 1735
1417 NOP
1420 SRO↑
1421 0
1422 JMP 1742
1423 JMP 1051
1424 JMP 1700
1425 JMP 1427
1426 JMP 1071      Check for zero Arg in Add.
1427 JMP 1604
1430 JMP 1613
1431 JMP 1646
1432 LDA 13
1433 STC 1127
1434 LDA 14
1435 STC 1130      Copy FAC into B
1436 ADD 1120
1437 STC 1126
1440 JMP 0
1441 LDA 15
1442 STC 1121
1443 LDA 16        Copy Arg into FAC
1444 STC 1122
1445 ADD 1123
1446 STC 1120
1447 JMP 0
1450 LDA
1451 0
1452 STC 1467
1453 JMP 1432
1454 ADD 1123
1455 STA 13
1456 SCR 13
1457 STC 1122
1460 ADD 1632
1461 STC 1120
1462 JMP 1700
1463 JMP 1470      Float Arg
1464 JMP 1751
1465 JMP 1613
1466 JMP 1622
1467 HLT
1470 JMP 1712
1471 JMP 1465
1472 JMP 1721
1473 JMP 1470
1474 LDA
1475 1120
1476 AZE           Fix FAC
1477 APO
```

6.

```
     JMP 1514
1501 ADA↑
1502 7764
1503 APO↑
1504 JMP 1517
1505 COM
1506 ADA↑
1507 340
1510 STC 1512          Fix FAC (continued)
1511 ADD 1121
1512 HLT
1513 JMP 1051
1514 LDA 13
1515 SCR 13
1516 JMP 1051
1517 LDA 13
1520 SCR 13
1521 APO
1522 ADD 1720
1523 ADD 1664
1524 JMP 1051
1525 JMP 1604
1526 JMP 1613          FAC / Arg
1527 JMP 1622
1530 JMP 1321
1531 JMP 1450
1532 JMP 1177          Integer + FAC
1533 JMP 1051
1534 JMP 1450          Integer x FAC
1535 JMP 1252
1536 JMP 1450
1537 JMP 1604
1540 JMP 1613          FAC / Integer
1541 JMP 1622
1542 JMP 1321
1543 JMP 1441
1544 JMP 1742          Complement
1545 JMP 1051
1546 JMP 1151
1547 JMP 1441          Maximum
1550 JMP 1051
1551 JMP 1151
1552 JMP 1554          Minimum
1553 JMP 1441
1554 JMP 1051
1555 JMP 1441
1556 JMP 1700
1557 JMP 1562
1560 CLR
1561 JMP 1051
1562 LDA 13            Sign
1563 SCR 13
1564 APO
1565 ADD 1570
1566 ADD 1720
1567 JMP 1051
1570 ~~JMP~~ 7775
1571 JMP 1756
1572 JMP 1441          Set sign plus
1573 JMP 1051
1574 JMP 1756
1575 JMP 1441          set sign minus
1576 JMP 1742
1577 JMP 1051
```

13-20

```
1600 LDA  15
1601 RCO  13     Is sign of FAC = Arg?
1602 SCR  13
1603 JMP  0
1604 LDA  16
1605 STC  1130
1606 ADD  1124
1607 STC  1127   Copy Arg into B
1610 ADD  1123
1611 STC  1126
1612 JMP  0
1613 LDA  14
1614 STC  1125
1615 ADD  1121   Copy FAC into Arg
1616 STC  1124
1617 ADD  1120
1620 STC  1123
1621 JMP  0
1622 CLR.
1623 ADD. 1130
1624 STC. 1122   Copy B into FAC.
1625 ADD  1127
1626 STC  1121
1627 ADD. 1126
1630 STC  1120
1631 JMP  0
1632 13
1633 CLR
1634 ADD  1720
1635 ADD  1120
1636 STC  1120
1637 LDA  13
1640 SCR↑1       Scale FAC right
1641 STC  1121
1642 ADD  1122
1643 ROR↑1
1644 STC  1122
1645 JMP  0
1646 JMP  1622
1647 JMP  1700
1650 JMP  1203   Check for zero in Add.
1651 JMP  1441
1652 JMP  1241
1653 JMP  1751   Clear
1654 JMP  1137
1655
1656
1657
1660             Not Used
1661
1662
1663 0
1664 3777
1665 7776
1666 CLR
1667 LDA  16
1670 LAM  14
1671 LDA  15
1672 LAM  13     Add mantissa of Arg to FAC.
1673 STC  1062
1674 LAM  14
1675 STC  1062
1676 LAM  13
1677 JMP  0
```

8.

```
      LDA 13
1701  AZE
1702  JMP 0
1703  SCR↑1
1704  LDA 14
1705  ROR↑1          Does FAC = 0 ?
1706  AZE
1707  JMP 0
1710  XSK↑
1711  JMP 0
1712  LDA 13
1713  ROL 1
1714  BCO .13        Is FAC normalised ?
1715  APO↑
1716  XSK↑
1717  JMP 0
1720  1
1721  CLR
1722  ADD 1665
1723  ADD 1120
1724  STC 1120
1725  LDA 14
1726  ROL↑1          Scale FAC left.
1727  STA 14
1730  LDA 13
1731  ROL↑1
1732  STC 1121
1733  LAM 14
1734  JMP 0
1735  LDA 13
1736  BCO↑
1737  4000           correct sign bit.
1740  STC 1121
1741  JMP 0
1742  LDA 13
1743  COM
1744  STC 1121
1745  ADD 1122       Complement FAC
1746  COM
1747  STC 1122
1750  JMP 0
1751  CLR
1752  STC 1120
1753  STC 1121       Clear FAC
1754  STC 1122
1755  JMP 0
1756  SET 12
1757  0
1760  LDA 13
1761  APO
1762  JMP 1743
1763  LDA 15
1764  APO↑           Complement Arg or FAC if negative.
1765  JMP 12
1766  COM
1767  STC 1124
1770  ADD 1125
1771  COM
1772  STC 1125
1773  JMP 12
1774  JMP 1742
1775  JMP 1177       Subtract
1776  JMP 1742
1777  JMP 1051
```

13-22

A Preliminary Description of an Operating System for the LINC Computer

by

Richard K. Moore
Department of Genetics
Stanford University

March 1965

13-23

The LINC

The LINC is a binary, 12-bit, 2048-word, digital computer. The core is divided into eight "quarters", each quarter being 256 words long. Quarter 0 contains cells 0 through 377*, quarter 1 contains cells 400 through 777, etc. LINC tapes are divided into 1000 blocks, each block being 400 words long. Thus, for example, the tape read instruction replaces the contents of one quarter of LINC memory with the contents of one tape block. There are two tape units on the LINC, units 0 and 1.

The LINC Operating System: LOSS

LOSS is based on a highly structured use of both tape units as well as the various quarters of memory. LOSS strives to provide a framework in which communication among LINC programs is very simple, thus allowing complex operations, such as compiling, to be accomplished through the successive efforts of relatively simple programs. This framework of simple communication is based on three artifacts: (1) the RECURSIVE OVERLAY, (2) the BUFFER, and (3) TEXTS.

The RECURSIVE OVERLAY

Under LOSS, tape unit 0 is reserved for a program stack and an overlay stack. The program stack consists of those programs which are available to be run on the LINC under LOSS. Each of the programs in the stack is identified by a number. Program 1 occupies blocks 11 through 15,

---

* From now on, unless otherwise noted, numbers are in the octal system.

program 2 occupies blocks 21 through 25, etc. The overlay stack begins following the top of the program stack. During execution, a program occupies only quarters 1 through 5. Quarter 0 contains a number of routines which are used by LINC programs. One of these routines is an overlay procedure which allows any program in the program stack to be called like a subroutine. As an example, suppose that there are four programs on the program stack and that program 3, currently under execution, wishes to call program 2. In this case program 3 merely places the number 2 in the accumulator and transfers control to the overlay routine. The overlay routine (1) saves the address from which it was called, (2) writes quarters 1 through 5 on tape (at the top of the overlay stack, say blocks 61 through 65), (3) reads program 2 (blocks 21 through 25) into core and (4) begins execution at cell 400. When program 2 has completed its execution (which may have included overlays of other programs, in which case program 2 would have been written on blocks 71 through 75) it merely returns control to the overlay routine which reads back in program 3 from the overlay stack and returns control to the cell saved at step (1) above.

The power of this system is shown by the fact that in the preceding example program 2 was able to perform its function without knowing what program had called it nor in what depth of recursion the overlay process was currently involved.

The BUFFER

While the elements of mathematical or logical operations are variables or arrays (single cells or blocks of consecutive cells), the elements of such operations as input-output or of inter-program communications are

LISTS, where the elements of a LIST are either variables or alphanumeric STRINGS. In order to allow for the manipulation of such LISTS, LOSS includes a general purpose BUFFER (beginning at cell 3000) together with two procedures, PUT and GET, located in quarter 0. The BUFFER is a pushdown stack whose unit of storage is a RECORD. The arguments to the PUT procedure are one or more LISTS; these LISTS are combined by PUT to form a RECORD which is placed on top of the BUFFER. Similarly GET($$L) causes the variables of the LIST L to assume the values found in the top RECORD of the BUFFER , which RECORD is then erased from the BUFFER. Since an OVERLAY affects only quarters 1 through 5, the canonical way for programs to transmit parameters to each other is by means of the BUFFER.

TEXTS

LOSS reserves tape unit 1 for the storage of TEXTS. A TEXT is a group of consecutive tape blocks preceded by a few special code words and a five character name. TEXTS are grouped together to form BOOKS, each BOOK being 100 tape blocks long. Thus BOOK 0 comprises blocks 0 through 77, BOOK 1 comprises blocks 100 through 177, etc. The 0'th block of each BOOK is an index which contains the names of all the TEXTS (in alphabetical order) in its BOOK together with their size and initial blocks. This formalism is not meant to restrict the kind of information which can be stored on tape, but rather makes it possible for allocation of tape storage space to become automatic and somewhat resistant to destructive over-writes. At the same time a block number together with the first two letters of its name become a concise, as well as a securely redundant, way to refer to TEXTS.

The LINC MONITOR

The LINC MONITOR is, by convention, program 1 on the program stack. Its only capabilities are to accept instructions from the typewriter, to perform simple operations upon the buffer, and to overlay any of the programs on the program stack. Its specific operations are:

DISPLAY n	n an octal integer; the n'th RECORD of the BUFFER is displayed on the scope.
TYPE n	The n'th RECORD is typed.
EXECUTE n	The n'th program on the stack is overlayed.
ERASE n	n RECORDS are erased from the top of the BUFFER.
PUT(L)	The LIST L is placed on the BUFFER. L consists of octal integers and alphanumeric STRINGS. A STRING, in this sense, begins with the character " and is terminated by % .

Since LOSS itself includes a method for referring to and loading programs, i.e., the OVERLAY, the MONITOR does not require a "loader" or a list of available "systems".

Richard Moore
March 8, 1965

BLINK

General Description

BLINK is a version of Subalgol designed for use with the LINC computer. Programs very similar to Subalgol programs are translated on the 7090 by the BLINK compiler (which is written in Subalgol), into relocatable LINC code.

Reserved Word Changes with Semantics

BLINK has no "library procedures", though it retains all of Subalgol's "intrinsic functions". The following Subalgol reserved words are without special meaning in BLINK:

STOP, SHLT, SHRT, EXTR, STATEMENT, WHILE, SEGMENT, MONITOR, STEP, INPUT, OUTPUT, TRACE, DPRECISION, LIBRARY, CARDREAD, PRINTOUT, COMPLEX, RE, IM, WRITE, READ, SQRT, LOG, EXP, SIN, COS, TAN, ENTIRE, SINH, COSH, TANH, ARCTAN, ROMXX, ARCSIN, ARCCOS, RCARD, READM, WRITEM, CHECKM, MOVEM, MOVEFILE, ENDFILE, REWIND, UNLOAD, FLAGM, etc.

The following reserved words are introduced or redefined with BLINK:

I.

A. ROTL, ROTR, SCLR

B. BTCLR, STCOM, STSET

C. LDA

D. STA

E. DO

F. REPEAT

G. RDC, RCG, MTB, WRC,
WCG, WRI, CHK, RDE,

H. INCR.

I. OVERLAY

J. STRING

K. LIST

L. PUT, GET

M. RESTART

N. *GETCOR

O. EXTERNAL

II.

A. I1, I2, ... I7

B. M, MF, MI

C. I1I, I2I, ... I7I

D. POINTER

Corresponding Semantics

I.

A. ROTL(N,OPERAND) : Intrinsic function; arguments type integer; result type integer; corresponds to ROL instruction; as in later intrinsic functions, the effect of the i-bit is obtained by using a value of N > 17.

B. BTCLR(MASK,OPERAND) : Intrinsic function; types integer; corresponds to BCL, etc.; as with ROTL class function, a constant first argument naturally reduces length of resulting code.

C. LDA(<arbitrary arithmetic expression>) : Expression is calculated and placed in accumulator; if of type floating, it is truncated to an integer; useful in connection with DO and STA as mentioned below.

D. STA(<simple variable>) : Contents of the accumulator are placed in the simple variable.

E. DO(<integer arithmetic expression>) : Expression is evaluated, treated as a LINC instruction and executed, e.g.,*

```
LDA(I)$
DO("470")$ COMMENT AZEi$
GO TO L$
STA(J)$
```

---

* Within BLINK examples, numbers are decimal unless placed in double quotes.

is identical in effect to:

```
EITHER IF I EQL 0$ GO TO L$
OTHERWISE$ J=I$
```

The DO function, however, is of only dubious value as a tool to create tight code; its real purpose is to allow the use of external device communication instructions in BLINK programs, e.g., OPR, SNS, etc.

F. REPEAT(<integer expression>)$ <statement>$: Identical in effect to:

```
DMY1=<integer expression>$
FOR DMY2=(1,1,DMY1)$ <statement>$
```

except that the REPEAT loop is more efficient and does not change the value of any variable in its indexing.

G. RDC(i,u,QNMBR,BNMBR) : Identical to LAP, except that i and u are represented by a 0 or 1.

H. INCR(<expression>,<variable>) : Identical in effect to:

```
<variable>=<variable>+<expression>
```

except that if the variable is subscripted, INCR calculates the subscript only once and INCR is a function having the new value of <variable> as its value.

I. OVERLAY<integer expression> : The OVERLAY* routine is entered with the integer expression in the accumulator.

J. STRING<identifier>(<integer>)=(<alpha string>) : The STRING declaration is identical to the ARRAY declaration, except that only a single dimension, and no irregular subscript ranges, are allowed. The effect of the STRING declaration is different in

---

* The reader should be familiar with LOSS at this point.

that the zero'th position of the STRING (even though not requested in the declaration) is reserved and filled with the size of the STRING, this information being necessary to the PUT and GET routines. STRINGS may be manipulated word by word, as are ARRAYS, through subscription. Thus S(1) refers to the first and second characters of the STRING S.

K. LIST : The LIST declaration is identical to the Subalgol OUTPUT declaration except that a STRING name (followed by empty parenthesis) is allowed as a LIST element, and fulfills the role served by the alphanumeric insertion phrase in Subalgol. A LIST, however, is somewhat more elegant than a Subalgol INPUT or OUTPUT list since a LIST can be used for either input or output (i.e., as argument of either PUT or GET), and includes the types of its elements, therefore needing no accompanying FORMAT (which concept therefore fails to exist in BLINK).

L. PUT,GET : These are simply procedures (always in core) which can have any number of LISTS as program reference parameters.

M. RESTART : When several BLINK programs are to be compiled during the same 7090 run, the non-last programs use RESTART to terminate compilation rather than FINISH. RESTART reloads the BLINK compiler instead of returning control to the monitor.

N. *GETCOR<integer><identifier> : This is a control card recognized by BLINK. *GETCOR is similar to RESTART, except that after completing the compiler output, the indicated disc file (rather than the compiler) is loaded.

O. EXTERNAL PROCEDURE, EXTERNAL SUBROUTINE : These declarations, identical to those in Subalgol, allow linkages to be created on the LINC between BLINK subprograms and subprograms created by means other than the BLINK compiler.

II.

A. Unlike Subalgol, BLINK has reserved variables. I1 through I7 (index registers), are simple variables of type integer with absolute address 1 through 7, respectively. These variables are GLOBAL and their values are not restored after an overlay.

B. M, MF, and MH are GLOBAL arrays with absolute base address of 0. Their types are integer, floating, and half-word, respectively. These are used to great advantage together with I1 through I7:

```
        M(I1) = M(I2)$ results in the elegant code
        LDA 2, STA 1
```

C. I1I, ... , I7I are used in conjunction with the M() and MH() arrays in order to reference consecutive words, or half-words, of core. As an example, the following statements replace the contents of quarter 5 by the contents of quarter 4:

```
        I1 = "3777" ; I2 = "2377" ;
        REPEAT  "400" ; M(I21) = M(I1I) ;
        COMMENT  LDAi1 , STAi2 ;
```

Thus the value of the indicated index register is incremented <u>before</u> it is used as a subscript. When the above program is completed, I1 will contain "2377" and I2 will contain "2777" . (It is a quirk of the LINC that the core is logically divided into halves; thus 3777 is the predecessor of 2000 and 1777 is the predecessor of 0.)

In the case of half-words, index registers are incremented by 4000 rather than by 1. Thus if an index register were stepping through the characters of quarter 4, it would assume successively the values 2000, 6000, 2001, 6001, 2002, etc. The 4000-bit indicates the right-half of the word.

An index register can be made to point at a variable by a statement of the form InI=<variable> . The code generated is:

```
        SET i n
        <variable location>
```

The following program places the characters of the STRING S() into quarter 7, putting one character (right justified) into each word.

```
        STRING S(20)=( alpha string );

        I7I="3377";

        COMMENT:  There is canonical correspondnece between
                  registers and quarters.

        I2I=S(0); I2=I2+"4000";

        REPEAT 40; M(I7I)=MH(I2I);

        COMMENT:  LDHi2 , STAi7 $
```

D. POINTER is that cell ("155") in QUARTER 0 which points to the top of the BUFFER. The statement POINTER=M(POINTER) would erase one record from the BUFFER. If we assume that the top record of the BUFFER begins with an alphabetic item, then the statement:

```
        I2=M(POINTER)+"4001";
```

would allow MN(I2I) to reference successive characters of that first item.

APPENDIX 1: QUARTER 0

"/" indicates data location

```
        0000 16
        0001 CLR
        0002 STA
        0003 3140
        0004 STA
        0005 3000
        0006 RCG
        0007 4011
        0010 JMP 400
        0011 HLT
        0012 HLT
        0013 HLT
        0014 HLT
        0015 HLT
        0016 HLT
        0017 HLT
SAVE    0020 SET↑15
        0021 3000
        0022 LDA↑
        0023 1776
        0024 ADD 0
        0025 STA↑15
        0026 STC 16
        0027 LDA↑16
        0030 ROL↑1
        0031 SCR 1
        0032 STC 14
        0033 LDA 14
        0034 STA↑15
        0035 LZE↑
        0036 JMP 27
        0037 LDA
        0040 21
        0041 STA↑15
        0042 LDA
        0043 15
        0044 STC 21
        0045 JMP 116
RESTORE 0046 LDA
        0047 0
        0050 STC 73
        0051 SET 15
        0052 21
        0053 LDA 15
        0054 AZE↑
EMPTY   0055 HLT
        0056 STA
        0057 21
        0060 STC 15
        0061 LDA↑15
        0062 STC 14
        0063 LDA↑14
        0064 ROL↑1
        0065 SCR 1
        0066 STC 16
        0067 LDA↑15
        0070 STA 16
        0071 LZE↑
        0072 JMP 63
        0073 JMP 0
PARAM   0074 ADD 23
        0075 STC 17
        0076 ADD 0

          0100 STC 16
          0101 LDA↑16
          0102 SCR 6
          0103 STC 15
          0104 LDA 16
          0105 BCL↑
          0106 7700
          0107 ADD 117
          0110 ADD 17
          0111 JMP 113
          0112 LDA↑17
          0113 STA↑16
          0114 XSK↑15
          0115 JMP 112
          0116 LDA↑
          0117 6001
          0120 ADD 16
          0121 STC 122
          0122 JMP 0
REPEAT    0123 LDA
          0124 0
          0125 STC 210
          0126 JMP 20
       ?/ 0127 SAM↑15
        / 0130 HLT 10
        / 0131 STC 136
          0132 ADD 210
          0133 JMP 74
        / 0134 JMP 1502
        / 0135 HLT
        / 0136 HLT
          0137 LDA↑17
          0140 AZE
          0141 APO
          0142 JMP 150
          0143 COM
          0144 STC 10
          0145 JMP 135
STEP      0146 XSK↑10
          0147 JMP 135
EXIT      0150 LDA
          0151 136
          0152 STC 243
          0153 JMP 241
          0154 SET↑11
POINTER   0155 3140
          0156 LDA 11
          0157 JMP 0
          0160 LDA
          0161 11
          0162 STC 155
          0163 JMP 0
PUT       0164 LDA
          0165 0
          0166 STC 176
          0167 JMP 154
          0170 LDA
          0171 11
          0172 STA↑11
          0173 JMP 160
          0174 COM
          0175 STC 276
          0176 JMP 0

          0200 STC 210
          0201 ADD 17
          0202 STA 11
          0203 LDA↑13
          0204 STA↑11
          0205 XSK↑17
          0206 JMP 203
          0207 LDA↑
          0210 0
          0211 STA↑11
          0212 JMP 160
          0213 JMP 231
GET       0214 LDA
          0215 0
          0216 STC 176
          0217 JMP 20
        / 0220 STC 301
          0221 JMP 154
          0222 AZE↑
EMPTY     0223 HLT
          0224 STC 301
          0225 JMP 175
LST.OP    0226 STC 243
          0227 ADD 0
          0230 STC 271
          0231 JMP 20
        / 0232 ROL 3
        / 0233 STC 276
          0234 JMP 271
GET.CL    0235 ADD 0
          0236 STC 243
          0237 JMP 154
          0240 STC 155
LST.CL    0241 JMP 46
          0242 CLR
          0243 JMP 0  OVERLAY
STRING    0244 ADD 0
          0245 JMP 74
        / 0246 JMP 1601
        / 0247 HLT
          0250 LDA↑17
          0251 STC 13
          0252 ADD 247
          0253 STC 271
          0254 JMP 46
          0255 LDA 13
          0256 COM
          0257 SRO
          0260 276
          0261 ADD 264
          0262 JMP 274
LST.EL    0263 ADA↑
          0264 7776
          0265 STC 13
          0266 ADD 0
          0267 JMP 74
        / 0270 JMP 1601
        / 0271 HLT
        / 0272 JMP 46
          0273 LDA↑17
          0274 STC 17
          0275 SRO↑
          0276 0

          0300 SET↑11
          0301 0
          0302 LDA↑11
          0303 BSE↑
          0304 6000
          0305 COM
          0306 ADD 11
          0307 STC 301
          0310 LDA↑11
          0311 STA↑13-
          0312 XSK↑17
          0313 JMP 310
          0314 JMP 231
          0315 HLT
          0316 HLT
          0317 HLT
          0320 HLT
          0321 HLT
          0322 HLT
          0323 HLT
          0324 HLT
          0325 HLT
          0326 HLT
          0327 HLT
          0330 HLT
          0331 HLT
          0332 HLT
          0333 HLT
          0334 HLT
          0335 HLT
RETURN    0336 JMP 46
          0337 STC 341
          0340 RCG
          0341 0
          0342 JMP 0
OVERLAY   0343 AZE↑
          0344 JMP 336
          0345 APO
          0346 JMP 363-
          0347 STC 17
          0350 ADD 0
          0351 STC 342
          0352 JMP 20
        / 0353 SCR 2
        / 0354 STC 356
          0355 WCG↑
          0356 4101
          0357 ADD 130
          0360 ADD 356
          0361 STC 356
          0362 LDA
          0363 17
          0364 ADA↑
          0365 7770
          0366 APO↑
          0367 CLR
          0370 ADA↑
          0371 7
          0372 ROL 3
          0373 ADA↑
          0374 4001
          0375 STC 377-
          0376 RCG
```

APPENDIX II: LOSS Character Codes.

CHARACTER	CODE
	00
!	01
"	02
#	03
$	04
%	05
&	06
'	07
(	10
)	11
*	12
+	13
,	14
-	15
.	16
/	17
0	20
1	21
2	22
3	23
4	24
5	25
6	26
7	27
8	30
9	31
:	32
;	33
<	34
=	35
>	36
?	37
@	40

CHARACTER	CODE
A	41
B	42
C	43
D	44
E	45
F	46
G	47
H	50
I	51
J	52
K	53
L	54
M	55
N	56
O	57
P	60
Q	61
R	62
S	63
T	64
U	65
V	66
W	67
X	70
Y	71
Z	72
(carr. ret.)	73
(end of text)	74

code derivation:

```
LDA '
(teletype code)
COM
ADA i
0277
SCR 1
```

Appendix III: A Detailed Description of LOSS, Especially QUARTER 0.

SAVE AND RESTORE:

These routines govern a push-down stack whose presence allows the other routines of QUARTER 0 to be recursive.

JMP SAVE
LOCATION$_1$
LOCATION$_2$
.
.
.
LOCATION$_n$+4000

causes the n locations together with their contents to be saved on the top of the push-down stack. The call JMP RESTORE causes the topmost list of locations on the stack to be restored to their former contents. This push-down stack occupies cells 3000 to 3140 and is called the SAVE-BUFFER, as opposed to the PUT-BUFFER which begins at 3140.

PARAMS:

Consider the BALGOL statement:

P(3,YZL1,L2) . . . (a procedure call)

which would be equivalent to the following LINC code:

```
LDA
Y
STC*+3
JMP P
0003
0000
Z
JMP L1
JMP L2
```

Value parameters are thus represented by their values in the calling sequence, name parameters by their addresses, and program reference parameters are preceded by the JMP prefix.

The heading of procedure P() might appear as follows:

```
P:  LDA
    0000
    JMP PARAMS
    7405
R:  0000
    0000
    0000
    LDAiI7
                 STA list for Z
             }
    LDAiI7
                 STA list for L1
             }
    etc.
```

Where 7405 derives from the formula 100(76 - no. of value params) + (no. of params), R will be assigned the return address for each call of P(), and R+1 and R+2 will be assigned the values of the two value parameters. Index register 17 is left by the PARAMS routine so that the non-value parameters can be conveniently obtained and stored where required in the body of P().

REPEAT:

The program

```
        JMP PRPEAT
        JMP PAST
        0005

                  }  code

        JMP STEP
PAST:   etc.
```

causes "code" to be executed 5 times. REPEAT is completely recursive (i.e., many REPEAT loops may be nested), and is the sole user of index register 10. A REPEAT loop can be terminated only by completing the full number of iterations, or alternatively, by executing the instruction JMP EXIT within the loop. If the count parameter (5 in the above example) is zero or negative, the loop is not executed at all.

PUT and GET:

These can best be explained through an example. The following program will replace each item in the LIST L2 by the corresponding item in the LIST L1. A,B,X,Y, are variables and S1() and S2() are strings. First, the Balgol statements:

```
STRING S1(5)=('ALPHAS'),S2(5);
LIST L1(A,B,S1()),L2(X,Y,S2());
PUT(;;L1);GET(;;L2);
etc.
```

Next, the corresponding LINC code:

```
   JMP PUT
   JMP L1
   JMP GET
   JMP L2
   JMP GET.CL
   etc.
L1:ADD 0
   JMP LST.OP
   LDAi
   A
   JMP LST.EL
   3776 . . .  (control word; the first digit is type: 3 is
   LDAi        integer, 7 alphabetic, and 1 floating.  The
   B           next 3 digits contain the complement of the
   JMP LST.EL  size of the item.)
   3776
   JMP STRING
   S1
   JMP LST.CL
L2:ADD 0
   etc....
   JMP LST.CL
S1:0005        (string size)
   4154
   6050
   4163
   7474        (end code)
   0000
   7474        ('extra margin' end code)
S2:0005
   0000
   0000
   0000
   0000
   0000
   7474
```

The BUFFER:

The BUFFER has a linked-list structure, the top of which is POINTER (cell 155). If in the previous example we assume that A and B have the values 7 and 24, respectively, then after the statement PUT(;;L1) , the BUFFER would have the following appearance:

```
LOCATION/CONTENTS

 0155     3154
 3140     0000 . . . null contents denote BUFFER bottom
 3141     3776
 3142     0007
 3143     3776
 3144     0024
 3145     7771
 3146     4154
 3147     6050
 3150     4163
 3151     7474
 3152     0000
 3153     7474
 3154     3140
```

If GET is ever entered when the BUFFER is empty, i.e., when location 155 contains 3140, then a halt occurs at location 223; if RESTORE is called when the SAVE-BUFFER is empty, a halt occurs at location 55.

OVERLAY:

When it is desired to OVERLAY a program from the stack, the program number is loaded into the accumulator, and JMP OVERLAY is executed. When a program has completed its function and wishes to return to its caller, JMP RETURN is executed. OVERLAY 0 is equivalent to RETURN. The sequence

```
(case 1) LDAi
         0005
         JMP OVERLAY
         JMP RETURN
```

is much more efficiently accomplished by:

```
(case II)  LDAi
           7772  (i.e., -5)
           JMP OVERLAY
```

for in the second case, the present contents of core are neither written on tape nor read back in when program 5 is finished; rather program 5's RETURN is placed on the same level with the RETURN appearing in case 1.

If an argument of OVERLAY is greater than the size of the program stack, then the last program on the stack is loaded; thus a copy of the MONITOR is usually in both the first and last positions. If RETURN is called when the OVERLAY stack is empty, a halt occurs at location 55 (since RESTORE will be spuriously called).

TEXTS:

The first two words of the 0'th block (the index) of each BOOK are

block no.	(0,100, or 200, etc.)
4253	('BK')

Each succeeding group of four words are TEXT entries

char$_1$	/	char$_2$
char$_3$	/	char$_4$
char$_5$	/	size (i.e., length in blocks)
initial block		

The end of the index is denoted by a zero where the first two characters of the next name would be.

Each text is headed by the following four word code:

char$_1$	/	char$_2$
char$_3$	/	char$_4$
char$_5$	/	max. size
current size/ type		

The purpose of having types is for file protection; when some program is written which will create a special kind of TEXTS - it can, of course, use

use any of the 100 available types. Thus far type 0 denotes a standard alphabetic TEXT (using the half word codes in Appendix II), and type 41 ('A') denotes an 'absolute' TEXT, i.e., a TEXT whose first block consist of a header alone and whose next 5 blocks are an absolute program ready to be placed on the program stack.

Appendix IV: How to Run a BLINK Program.

The BLINK3 compiler is stored in the disc files of Stanford's 7090 computer. In order to use this compiler, it is necessary to prepare a card deck as follows:

No. 1 Card:	SYSTEM	F-INFO
	TAPES	Mount on A3, at low density, a tape which can be removed from the Computation Center.
No. 2 Card:	SYSTEM	F-INFO
Control Card:	Cols. 1-6	BLINK3 file number (changes periodically), right justified.
	Cols. 7-11	BLINK

There should follow a BLINK source deck. This deck should be terminated by a RESTART, FINISH, or GETCOR card. If a RESTART terminator is used, it should be followed by another BLINK program.

The output produced by BLINK3 will be on the tape which was mounted on unit A3. This output is quite similar to that produced by the SUBALGOL compiler, i.e., listings of the source decks, diagnostic messages, symbol tables of the compiled programs (these being especially useful for console debugging). In addition, however, the tape will contain the actual LINC code produced by the compiler.

If no compiler error messages are produced, the tape is brought over to the LINC, mounted on the LINC's tape unit, and read by an appropriate LINC program. One such program merely searches the tape, ignoring all that it sees, until it comes to compiled code. That code is then transferred to the LINC tape, unit 1, in the form of a TEXT.

One of the principle drawbacks of the BLINK3 system is the means by which information is transfered from the 7090 to the LINC. Not only is it inconvenient to have to physically travel to the Computation Center, but jobs requiring special tape handling are not given top scheduling priority on the 7090.

The BLINK4 system will employ an electrical connection between the LINC and the Computation Center's PDP. The BLINK programmer, instead of preparing a card deck at the Computation Center, will prepare a TEXT on the LINC. When completed, this TEXT will be sent to the 7090 via the PDP. The first line of the TEXT will actually be the No. 2 card expected by the 7090 monitor. Instead of using tape A3, the BLINK4 compiler will send its output directly to the PDP, which will relay it to the LINC. The LINC in turn will write the output on IBM tape. The tape can be examined on the LINC's scope and never need be listed. If the tape contains error messages, then it is only necessary to alter the original TEXT and re-send it to the 7090. If there are no error messages, then the procedure becomes the same as under BLINK3.

THE LINC TELETYPE MONITOR SYSTEM

LINCT

The System consists of a monitor which accepts Macro-instructions and associated octal parameters from the teletype and separately coded programs which are executed to acheive the desired result and return to the monitor. The requirements to use LINCT are a standard LINC and a Teletype Corporation series 33 teletype attached to relay #0 and External Line #0. Tape requirements are Blocks 200 and forward on unit 0, as the system is followed by an indefinite scratch area a practical upper limit of 277 is satisfactory. !

The monitor is started by an 0700 0200 in the switches and a START 20. A return, line feed is the signal that the monitor is ready to accept input. The operator then types a 2 letter Macro code followed by the appropriate octal parameters and unit numbers (binary only). Commas separate fields and blanks are ignored. All parameters need not be explicitly specified as they are initially defined as zero; however, as unit 1 is desireable as a library tape, unit numbers are assumed as 1 unless a 0 is typed in the field (in some cases a , is necessary to "open" the field, but once a field is opened 1 is assumed) unless zerois typed, ,, means 1). An error in a calling sequence (e.g. illegal macro code, something besides 0 or 1 in a unit field, a non-octal digit in a octal field, or too many parameters in some cases) will result in a NO being typed back followed by a carriage return, line feed signifying ready status for a new line. The RUB OUT key is interpreted as an illegal charachter resulting in the NO and may be used to delete the line. The RETURN key effects execution of the Macro.

The following pages contain write ups of the Macros with descriptions and calling sequences. Also a page of actual teletype operation.

1. Input Type: IN n,u,x

This program receives alphanumeric text from the teletype and record it on tape in succesive blocks beginning at Block n, Unit u (initially assumed 1).

Input Description: All alphabetic, numeric, and special charachters are valid except ? which is ignored. The RUB OUT key will delete only the line currently being typed (multiple depressions have no effect on the text). Upon depressing it the program will do a carriage return, type 7 X's over the junk that is deleted and proceed to a new line. To end a line, press RETURN. The program gives the line feed when ready to accept a new line (usually immediate, but delayed when writing tape.) To terminate input, press EOT (CTRL & D keys) immediately following a RETURN, at any other place a NO is typed back and the EOT is ignored. The program after an EOT will write out the remaining text and type the message: LAST BLOCK USED IS ??? for tape logging purposes. Control is then returned to the monitor.

Line Numbering: If $x \neq 0$ numbering is suppressed. If $x=0$ (normal) an octal line number for the preceeding line will be typed every eighth line beginning after line zero. The number is preceededdby < to avoid confusion with the numerous 4digit numbers that appear.

Output Format: The first two words of every block are 7575_8, the third word is negative if the block is last in the text (never looked at in the system but might be of value). The text begins in the left half of the fourth word and continues by LINC half word indexing through the entire block. The end code is signaled by a 13_8 followingba 12_8 (EOL code). This places the restriction that the charachter \ cannot be first in the line.

2. Type (list)   Type: TY n,u,L_1,L_2,N_c,x

Type will list the text beginning at Block n, Unit u (initally assumed 1) under control of the remaining parameters.

Normal Format: If L_1=L_2=N_c=x=0 the printed output of the entire text has the same format as the input listing except for deleted lines.

Unusual Formats, Control Parameters:

L_1 : Begins printing at line number equal to L_1

L_2 : Stops printing at line number equal to L_2 ;however if L_2=0 the entire text after L_1 is printed. Either L may be greater than the last line number meaning equality to it.

N_c : Prints the first N charachters per line. Useful for quick and dirty listings to get line numbers after alterations.

x ; x≠0 suppresses line numbering.

Notes: If a block read does not have the 7575 text code a NO is typed and immediate return to the monitor is made. If the line is longer than 65_{10} charachters, the program will start a new line on the teletype and continue printing the same line of text.

3. Group Type: GR N1,U1,N2,U2,N3,U3,.......Nn,Un

Group will group the n texts at Block N_i, Unit U_i into one text and store the result at the System scratch area (around 240 depending on the edition being used) on unit 0. The main purpose of this program is to prepare multiple texts for assembly. From 0 to 174_8 texts may be called for. Grouping in equivalent to catenation, i.e. the first of text I follows the last line of text I-1 and the last line of text I is followed by the first line of text I+1. Text 1 follows nothing and text n is followed by the text end code.

Operating Notes: A NO is typed if any block read lacks the text code. At the end of the grouping the message m BLOCKS GROUPED AT ??? is written before return to the monitor. m is the number of blocks written at the beginning of the scratch area which is given in the ???.

Caution: Only a one block buffer is used so nothing may be inserted in front of the scratch area text but may be appended.

4. Copy Type: CP m,n_1,u_1,n_2,u_2

The program will copy m blocks from block n_1, unit u_1 to block n_2, unit u_2. If m>1, successive blocks are copied. The information may be of any type and is not limited to texts.

Caution: A three block buffer is used so if moving more than three blocks forward on the same unit care must be taken to avoid clobbering blocks which have yet to be read. The range of m is 0 to 1000.

5. Alter Type: AL n,u,x

This program will perform a group of insertions and deletions to the text at block n, unit u. It makes use of the scratch area and programs Input and Copy. A brief description of its operation is in order. First the macro is typed, then the monitor instructs Input to place the alteration text at the beginning of the scratch area. The Alteration Text is then typed in (Format described below.).ended with an EOT (no last block message is typed). Input then enter Alter. Alter reads the Alteration text and the text to be altered (n,u). It writes the altered text in the scratch area but immediately following the Alteration Text (Note: The altered text is never at the beginning of the scratch area). It then types a message and conditionally renters Copy to return the altered text to block n,u. In any case the monitor is re-entered.

Alteration Text: Alteration instruction lines and lines to be inserted in the text make up the Alteration Text. The alteration instructions refer to line numbers in the text to be altered and references must be in sequence from line 0 forward. The format of alteration instructions is; $/m,n_{\text{return}}$ no blanks are permitted on the line.
The program will remove lines from the beginning of line m to the beginning of line n and will insert any lines of text that follow it until another alteration instruction is encountered. (or an end of text)
Note: A slash cannot be the first charachter in the Alteration text unless the line is an alteration instruction (no restriction on original text).
The message ILLEGAL PROCEDURE is typed if: a block is read which does not contain the 7575 code, an alteration instruction of an illegal format, or an alteration instruction of legal format but where m>n, m<(the previous instructions n), or m>(last line number of the original text).
As the process is a merge, line numbering of the old text is preserved.

throughout the one pass, after completion however, the line numbering is dependent on the alterations made and a partial print may be used to determine line numbering in places of interest.

If x=0 the text will be returned in place of the original text if the length of the altered text is less than or equal to the original, otherwise it will be left in the scratch area and the message n BLOCKS REMAIN AT x will be typed, where n is the number of blocks and x is the location of the first block. If the altered text is returned the message n BLOCKS RETURNED will appear. The purpose of this criterion is to prevent clobbering a block of text immediately following the original text. If x=1 the text is unconditionally returned, and if x=2 the text is not returned.

An example is in order:

```
AL356    means alter block 356, unit 1, x=0
/1,1     says "remove nothing and insert the following lines in front of line 1"
ALPHA
BETA
GAMMA       these three lines are inserted
/5,10    says "remove lines 5,6,7, and insert" but there is nothing to insert.
/12,13   says "remove lines 12 and insert before 13"
DELTA    this line is substituted for 12
/17,100   presuming a text of say 62 lines, this will remove line 17 through
EPSILON   the end of the text and append whatever follows it to  an end of text
ZETA      (another alteration instruction is illegal as m must be less than
ETA        or equal to 62 and greater than or equal to 100)
THETA
(eot)    whereupon the alteration is performed the text is found to be smaller
         and blocks are returned,  the message is then typed.
   1 BLOCKS RETURNED.
```

A note on Grouping: Grouping is an easier way to insert information before an existing text or appending to it. As an altered text is never left at the beginning of a scratch block it is necessary to group it to place it at the beginning (one block of text may be grouped in front of it safely).

6. Assemble. Type: AS

LINCT has been tied into the LAP Convert Metacommand, which works quite satisfactorily, through a program which transforms LINCT text at the beginning of the system scratch area into LAP text and places the result in blocks 336 forward (LAP input area) and enters the LAP converter which assembles to blocks 330 333 (270 to 277 for 2K version). The rules of line structure given in the LAP III Manual must be adhered to, obviously a new special charachter set is necessary as LAP uses a somewhat unusual set. The changed charachters were chosen for typing convenience and resemblance was a secondary situation. The following is the list of changes

LAP	LINCT
i	; (semi colon)
p	* (asterisk)
u	! (exclamation mark)
I	/ (slash)
Origin	$ (dollar sign)
Tag	: (colon)

```
OPERATIONAL SAMPLE FROM TELETYPE
IN523,1
$220
LDA;
1777
ROL 3
:5H JMP 7B
WRC;10  (note:  unit 1 if from cards)
2/240
RDC ;!
4A
4A=230
JMP *-5
LAST BLOCK USED IS 523
GR523
AS
```

7. Execute Type: XE n,n,n,....,n

This is not a program but a means of loading and executing a program. The monitor will place the first parameter in location 1375 and successive locations thereafter, when RETURN is pressed the monitor jumps to 1375 and the octal commands typed in will be Executed. The reason 1375 was chosen was that one may do an RDC and a JMP and it will leave parameters in 1400 forward or if a program is to be read into quarter 2, three 16's (NOP) may be given and instructions are in quarter 3. Overlaying is a hit and miss proposition as a Missed tape check on the first attempt to read will cause error.

A Floating Point Subroutine Package for the LINC

Jeremy Pool

This package was written for programs compiled by "Blink", an IBM 7090 Balgol compiler for the Linc, written by Richard Moore; however, it is completely compatible with any machine language program.

The arithmetic routines - add,multiply, divide - and the float subroutine are, with minor alterations, those written by J.C. Dill, W. M. Stauffer, and R. W. Stacy of the University of North Carolina.

In this package the format for floating point numbers is a one word exponent followed by a one word mantissa. Both words are signed, one's complement numbers (standard form for the Linc). Zero is designated by a zero exponent and a zero mantissa. Floating point numbers must be in standard form, so that the mantissa has an absolute value between 010 000 000 000 and 011 111 111 111. The decimal point is understood to be between bits 11 and 10 of the mantissa.

In addressing it is always the first word, the exponent, which is specified.

The calling sequence is as follows:

```
JMP    400
  A1
  O1
  A2
  O2
  .
  .
  .
  An
  On
  Next instruction
```

A1 is the address of the first operand. Three possible formats for this address are possible:

A1 > 0 ; A1 = absolute address of operand

A1 = 0 ; The operand is the floating accumulator

A1 < 0 ; A1 = indirect address of operand

For indirect addressing, the address is not complemented; only the 11 bit must be set to 1. Thus with A1 = 4063, location 63 contains the address of the operand, not location 3714.

01 is the desired operation. Two forms are possible:

01 < 0 ;	Execute the specified subroutine, and then continue to execute the next specified subroutine.
01 > 0	Execute the specified subroutine, which is the last in the series of subroutines, and return and execute the next instruction in location p + 1.

Here again, when 01 < 0, this is specified by setting to one the 11 bit, not by complementing the entire number. Thus 4002 means add and continue to execute floating point instructions while 0002 means add and return from the floating point package.

Some of the routines are "integer" subroutines and assume one of the numbers involved to be an integer. In this case the actual address of the integer is specified by the operand, directly or indirectly.

The subroutine codes, their mnemonics, and their explanations are as follows:
(op = operand; FAC = floating accumulator; ac = Linc's accumulator)

1.	CLA	Clear and add	c(op)——> c(FAC)
2.	ADD	Add	c(op) + c(FAC) ——> c(FAC)
3.	COM	Complement	complement of c(op) ——> c(FAC)
4.	MUL	Multiply	c(op) x c(FAC) ——> c(FAC)
5.	DFA	Divide (a)	c(FAC) / c(op) ——>c(FAC)
6.	DAF	Divide (b)	c(op) / c(FAC) ——> c(FAC)
7.	IAD	Integer Add	c(op) + c(FAC) ——> c(FAC) ; c(op) = I

In the previous subroutine and in some of the following, the operand is assumed to be an integer (I).

10.	IML	Integer multiply	c(op) x c(FAC) ——> c(FAC) ; c(op) = I
11.	DFI	Integer divide (a)	c(FAC) / c(op) ——> c(FAC) ; c(op) = I
12.	DIF	Integer divide (b)	c(op) / c(FAC) ——> c(FAC) ; c(op) = I
13.	FIX	Fix	c(op) is convert4d to a fixed point number (an integer), and is stored in the regular, Linc, accumulator. Numbers are not rounded; all fractional parts are lost. Any number less than one is stored as zero. Any number greater than $3777_{(8)}$ or less than $-3777_{(8)}$ is converted to 3777 or -3777 respectively.
14.	FIT	Float	c(op) is assumed to be an integer. It is converted to a floating point number and replaces c(FAC).
15.	CLR	Clear Storage	0 ——> c(op)
16.	MAX	Maximum	The c(op) is compared with c(FAC). The larger value replaces c(FAC).
17.	MIN	Minimum	The C(op) is compared with c(FAC). The smaller value replaces c(FAC).
20.	SGN	Sign	If c(op) < 0, then -1 ——> c(ac) If c(op) = 0, then 0 ——> c(ac) If c(op) > 0, then 1 ——> c(ac).

21. INC Increment c(op) + c(FAC) ——> c(FAC) and ——>c(op). This is the floating point counterpart of Linc's add to memory instruction.

22. IIN Integer Increment c(op) + c(FAC) ——> c(FAC) and c(op); c(FAC) = I Note that in this instruction it is the FAC, not the operand, which is assumed to be an integer.

23. STO Store c(FAC) ——> c(op)
24. SSP Set Sign Plus |c(op)| ——> c(FAC)
25. SSM Set Sign Minus -|c(op)| ——> c(FAC)

26. SQT Square root ; $\sqrt{|op|}$ ——> FAC

27. IPT Input ; the number inputted on the keyboard ——> op
The number is inputted in decimal and is terminated by a space. The number may be preceded by a minus sign. Any of the following inputs are allowable:

27.345 △
-.0001 △
996 △
-101 △
-62. △
△ (=0)

There is no limit to the number of digits inputted. Pressing "del" at any time during an input deletes what has been entered and the entire number must be retyped.

30. OPT Output ; the operand is outputted on the teletype in the following format:

X.XXX, XXX return and line feed

The first four digits are the decimal mantissa and the last three the characteristic as a power of ten.

Also PKG = JMP400. The mnemonics are used in an assembly program to be described.

If locations 1472 and 3742 are altered so that they both hold "4276", the teletype does not return after it has outputted a number; it spaces once. (Normally these locations hold "6570")

The actual teletype output routine is included as a subroutine within the package, so that it can be jumped to from outside the subroutine package. To type a character, load the accumulator with that character's teletype code and jump to location 1742. Control will automatically be returned to p + 1. Index registers 12 and 15 are used by this subroutine and are not restored if one jumps to 1742. A modification is included for scope output of the same format.

The package occupies all of quarters one, two, and three. Quarter 7 is used from location 3700 to 3756. All index registers are restored to their previous value except in the case mentioned above.

The floating accumulator is locations 1120 and 1121.

No error detection is provided. Overflow of exponents in arithmetic subroutines will yield incorrect answers, not error messages. The same is true of invalid operation codes, etc.

A sample program which calculates $\frac{3}{x^2-3x}$, which stores the result in the floating accumulator, and which leaves +1, -1, or 0 in Linc's accumulator, depending upon the value of the result, follows:

x is in 1G
3 (integer) is in 1T
locations 24 and following contain LDA
0
JMP 1000

JMP 24

1T
4014 — Loads -3, floating point, into the FAC
0
4003

1G
4002 — $x-3 = c(FAC)$

1G
4004 — $x(x-3) = x^2-3x = c(FAC)$

1T
4012 — $\frac{3}{x^2-3x} = c(FAC)$

0
0020 — Determines sign of answer Exit because the 11 bit = 0

Next instruction

Program follower

To use:
Read in program to be tested and execute it once.
Then read it out temporarily on tape and read it back into memory, in executed form, into blocks of upper core corresponding to those blocks of lower core where the program normally operates, i.e., quarter 0 into quarter 4, 1 into 5, etc.

On sense switches set the quarters which the program uses (actual <u>lower</u> core quarters).

On the right switches set the address of the first executed instruction.

Read in the Program follower and start 20.

When the program halts locations 20 and following will contain the locations of your program which are instructions. The following is an example of the final output:

loc.	contents
20	20
21	176
22	405
23	760
24	0
25	0
26	0

This means instructions were contained in locations 20 through 176 and 405 through 760 of your program.

The program follower is not perfect. It will not catch returns from subroutines where the return address is manipulated to be anything besides p+1 or a constant return address. It will not catch jumps executed by pulling addresses out of a jump table. It will assume that all XSK instructions can proceed to both p+1 and p+2, while this is not always true. Therefore, the results may contain a few locations which are data and may omit locations which contain instructions.

The program follower is just that; it does not tell you what parts of your program were meant to be instructions, it tells you which locations can be reached, as instructions, by the various jumps and branches of your program. Thus it provides a good method for troubleshooting a program by showing you where your program actually can go.

Mnemonic dump

To use:

Read program to be typed into upper core in quarters corresponding to the lower core location of the program, i.e., a program which runs in quarters 0 and 2 should be read into quarters 4 and 6.

Have tape JP on unit 1.

Read Bl 310 into quarter 0 and start 20.

Type on the kbd one-digit numbers corresponding to the quarters used by the program. For the case mentioned above, type 0 space 2. Separate these digits by spaces.

Then type in the locations which are instructions in the form specified below.

If the program postulated above ran from 20 to 360 and from 1000 to 1377 the entire input should be as follows:

$$\underbrace{0\triangle\ 2\triangle}_{\text{Quarters}}\quad \underbrace{\underbrace{0020\triangle\ 0360\triangle}\underbrace{1000\triangle\ \ 1377\triangle}}_{\text{Instruction Locations}}\quad \text{EOL}$$

Location 0 may not be specified as an instruction location

Sense switches control the output format as follows:

All set to 0 = single column output
SW 1 at 1 = 2-column output, numbering spaced by 200
SW 1, 3 at 1 = 2-column output, numbering spaced by 100
SW 1, 2, 3 up = 4-column output