pdp12
pdp12
pdp12
pdp12
pdp12
pdp12
pdp12
pdp12
pdp12
pdp12
pdp12
pdp12
pdp12
pdp12
pdp12

proceedings of the neurophysiology psychophysiology session 1969

with linc history discussion

digital equipment corporation

CONTENTS

Page

Forward. 1

Why a LINC 2

Comparison of the LINC, LINC-8 and PDP-12 Computers 3

LINC Computer Application to Research on Behavioral Correlates of Evoked Neuroelectric Potentials of the Brain 5

Computer Automated Procedures For Assessing Some Experiences of Family Membership 7

Computer Techniques for Studying Biologic Rhythms: Quantitative Chronobiology 10

Discussion. 13

PROCEEDINGS OF THE NEUROPHYSIOLOGY-PSYCHOPHYSIOLOGY SESSION AT THE 1969 SPRING DECUS BIOMEDICAL SYMPOSIUM

FOREWORD

This symposium was directed at exploring applications of LINC-class computers (classic LINC, LINC-8, MICRO-LINC, PDP-12) in several diverse areas of neurophysiology and psychophysiology. Users of these small laboratory computers still encounter students and colleagues trained in university centers on large-scale machines who react to the LINC as a "toy" or as a "cute little logic box". That the LINC is much more than this is a tribute to the ingenuity and genius of Wesley Clark and his colleagues who developed the LINC concept at MIT in 1961, to Mary Allen Wilkes who has developed the very approachable, interactive LAP assembly languages, and to manufacturers such as DEC who have maintained most features of the original design while increasing certain capabilities, lowering cost, and providing field maintenance. Papers by Severo Ornstein, a member of the original LINC group, and Richard Clayton, product engineer for the LINC-8 and PDP-12 at DEC, document the birth and evolution of the LINC concept.

The LINC was not intended as a replacement for the large-scale computer, but was designed as a flexible laboratory tool, a sophisticated super-oscilloscope if you like, which would not require the user to become a "computernik". Because its capabilities are clearly limited, it encourages many scientists to think twice about computer approaches to a problem, especially if the problem was conceptualized as requiring the capabilities of a dedicated UNIVAC 1108. It is my own experience that the resulting LINC solutions are significantly more imaginative, economical, and heuristic, emphasizing the use of the computer as an interactive real-time experimental tool, rather than as an off-line analysis device. The generality of this experience is documented in papers discussing varied LINC applications by William R. Goff, William H. Sheriff, and Charles F. Stroebel. The paper on "LINC Applications to Quantitative Analysis of Parkinson's Disease" by Anthony Sances, Jr. was not available for publication. Regretably, time did not permit reports of other LINC applications from the approximately 250 extant LINC installations.

The panel discussion following the formal reports was joined by Mary Allen Wilkes from the Computer Laboratory at Washington University (St. Louis) and by Dr. Frederick Hiltz (Cornell University, Ithaca). Their comments on LINC software and comparisons of the LINC with process-control class computers were especially valuable. Richard Clayton and Edward Kramer of DEC presented a PDP-12 demonstration at the close of the symposium.

As chairman, I express my thanks to the participants, as well as to Susan Bascom of DEC who ably handled arrangements, and to Dorthie McIntyre of the Institute of Living for her excellent work in preparing the manuscript.

Charles F. Stroebel, Ph.D.
Chairman

"WHY A LINC?"

Severo M. Ornstein
Bolt, Beranek & Newman, Inc.
Cambridge, Massachusetts

Although I have not been involved directly with the LINC now for some time, I was nonetheless there when the idea first germinated and helped to bring it into being. So it seems appropriate by way of introduction for me to spend a few minutes reviewing the early history and some of the original objectives of the LINC. I'm particularly interested in hearing peoples reactions to the changes represented by the PDP-12 and later I'll give you some of my own impressions.

It all began somewhere in 1960 and although many people contributed to the final product, the LINC was primarily Wes Clark's invention. At that time we were part of the Digital Computer Group at M.I.T.'s Lincoln Laboratory. We had come together originally through the hope that our work in computer development might be directed to the needs of the medical community. Several people in the group were involved in neurophysiological research and this provided the background for some of the decisions which had to be made about the machine.

During the Spring of 1961 Wesley filled several notebooks with developing ideas for the machine and by the Fall these ideas were being discussed and criticized. The fundamental concept survived this pretty much intact although many details changed as the picture of the machine began to crystallize. We eventually produced a prototype early in 1962; a binocular version of today's cyclops with the two scopes mounted side by side. Early on, Wesley had referred to the machine as the LINC, and most of us assumed that this had something to do with our presence within Lincoln Laboratory. It was only when the group was about to leave Lincoln that Wes announced casually that LINC really stood for Laboratory Instrument Computer. We clearly still had much to learn.

Back in those days the large machines were beginning to enter their heyday. The computer center could surely service everyone if the computer was only big enough. Claustrophobia of the core, stemming from having been shut up in small early memories, had permanently affected the mind. The idea that other features could be of overriding importance was not taken seriously. And yet a large part of the scientific community, the experimental as opposed to the actuarial, was being poorly serviced. The inaccessibility of the large computer center facilities made debugging of programs a chore. If rewriting and reshaping of programs (as an experiment developed) was a continual rather than a one time effort, matters were even worse. If in addition one needed to connect laboratory apparatus to the computer then the case became virtually hopeless. The computer was remote from the laboratory and most likely accepted only punched cards. And even if it took your data in, without a scope you might never know what had become of it.

These then constituted the physical inaccessibilities. There were also the psychological ones having to do with the size, cost and general impressiveness of the computer. Experimental scientists generally thought in terms of intermediaries who would write programs to solve differential equations, compute standard deviations, etc. No one really thought of the computer as a tool to be utilized directly in the course of an experiment. The computer therefore had very little impact on the kinds of research which were undertaken.*

LINC

Accessibility was thus a key consideration throughout the design of the LINC. Separation of the logic/power supply cabinet from the user's units was an attempt to increase convenience in settings where for space or environmental reasons only the minimum of equipment was permissable in the laboratory. The scope was included so that one could observe results in real time rather than somewhere in the midst of a 4 inch thick printout. The A/D conversion system accepted signals which could come in most cases directly from lab apparatus or at most via simple amplifiers. The console is perhaps the only one I have ever seen which, with its debugging features and speed control, is a pleasure rather than a frustration to use. The knobs permitted one to control parameters of all kinds in the most natural way. The I/O system had to be extremely flexible and we preferred to arrange matters so that devices could be connected with minimal hardware combined with clever programming. XLN, OPR

I think the interfaces for the Calcomp plotter, the Teletype and even the more elaborate tape transports speak for themselves. Pointing out the desirability of the LINC tapes seems unnecessary at this stage although we should pause to consider how handicapped the machine would be without them. The ability to walk up to a cold machine (with no bootstrap program in memory) and load a program from any block of tape into the memory with one simple action is enormously convenient.

Despite these added features the price was to be kept down and the machine had to be small enough and reliable enough to be portable. These conflicting requirements of course meant some compromises. The (expensive) memory had to be kept small not only in number of words but also in word length. This complicated the instruction format although by way of compensation I feel we ended up with a powerful repertoire. Speed was sacrificed for savings; we convinced ourselves that 8 bits of accuracy was sufficient for most analog measurements; etc. 12

The advances that have taken place in reducing the cost and size of logic and speeding up memories have permitted some of the improvements represented by the PDP-12. (Although of course prices are always disappointing.) The faster and larger memory is welcome as is the overlapped tape transfer logic. The larger scope would be more impressive if the precision had been increased.

Combining the parts of the machine into a single larger cabinet certainly gets rid of the cables which are something of a nuisance. However, it clearly violates the original goal of trying to keep the bulky and uninteresting part of the machine out of the way. The addition of the Teletype, grim device that it is, is a concession to reality that we also were ultimately forced to make.

All in all I feel ambivalent toward the changes that have taken place from the classical LINC. There are some clear gains but I think there may prove to have been some losses as well.

---

* *We are no doubt still suffering from the scientific equivalent of the "first-sports-car" syndrome. If we don't outgrow it, the computer is hardly to blame.*

COMPARISON OF THE LINC, LINC-8, AND PDP-12 COMPUTERS

Richard J. Clayton
Digital Equipment Corporation
Maynard, Massachusetts

First, there is no one person behind any of this, and in fact looking at the evolution of any of the concepts, which led to the classic LINC, one finds that by no means did the original LINC group immediately wash their hands of the LINC development. In fact, there has been a gradual transition to the point that Digital and other manufacturers have been more intimately involved with the evolution of the basic concepts.

The LINC-8 was done at a technical and political time, leading to the particular configuration that was generated. Since this leads to a bit of history, I'll back up for a moment. The LINC-8 was originally envisioned as a black box, which one could buy, plug onto a standard PDP-8 computer, and thereby achieve something that was program compatible with the Classic LINC. That was the original concept, and as Severo has indicated, sometimes original concepts get molded slightly in the development. We built such a machine, and in fact ran it, and it has been since dismantled. We decided that the machine was not as good as we could accomplish. We could have essentially the same machine, running 70% faster, by removing the constraint that it be a black box plugging onto a PDP-8. This was accomplished by using the same general concepts, but intermingling them a bit more closely with the PDP-8 section; that was what in fact led to the LINC-8.

The economies of production at the time that the LINC-8 was developed suggested that it have a basic PDP-8 within it. That was something we felt an economic necessity to get the price down, and to achieve the kind of price-performance ratio we wanted. That is where the LINC-8 came from. The LINC-8 was designed to be a machine which was either a classic LINC or a PDP-8, and did not address itself in great detail to highly intermixed programming between the two modes, or to some of the details of this intermixing. That was not a point of optimization.

Some of the most significant LINC-8 features of course, have been the cycle time and the memory size. I think Severo had indicated and I'm sure that most of the users concur, 2,000 words of memory is in many cases quite confining, either for data handling or for programming. A number of users have found the time saved in figuring out how to squeeze something into 2K in many ways justifies the price for additional memory, even from 4K to 8K. And in the case of the current configuration of the PDP-12, that price is extremely low. In fact, well over half the PDP-12's on order are 8K machines. I personally feel that this is because most people realize the advantage of having additional memory with which to be rather careless. This can be a real economy when it comes to the actual programming and handling of the machine. I think that is the primary reason.

There are some technical improvements which are just fine tuning of the design which have progressed. Specifically they are the speeding of the display instructions, the increased accuracy of the A to D converter, things of that type.

Further, the larger scope display of the PDP-12 offers many new possibilities. The buffered tape extends the usefulness of the LINC concept to real time situations where continuous processes which require more calculations than can be done on "the fly" come in at rates faster than two cycles/sec. Practical continous asynchronous data transfers up to 1000 numbers/sec. can be achieved to the tape.

The clock offered with the PDP-12 represents an extension of the LINC concept to the variable of time. Because a powerful clock is mass produced the cost is low. When many users have totally compatible systems the interchange of programs is far more practical and probable.

Figures I and II will serve to highlight differences amongst the Classic LINC, LINC-8, and PDP-12.

FIGURE 1: COMPARISON OF LINC, LINC-8 AND PDP-12 COMPUTERS

THESE THREE SYSTEMS OFFER AN EVOLUTION OF THE LINC CONCEPT

MAJOR FEATURES

	CLASSIC LINC	LINC-8	PDP-12
Relative Inst. Time	7	1.3	1
Effective Memory	2K	3K or 4K	4K
Scope Size (Inches)	4 X 5	4 X 5	7 X 9
A-D Accuracy	8 Bits	9 Bits	10 Bits
PDP-8 Programming	No	Yes	Yes
PDP-8 Peripherals	No	Yes	Yes
Memory Expansion	?	32K	32K
Buffered Tape	No	Partially	Completely
Trap Capability	No	Yes	Yes
I/O	Direct	Direct/Interpret	Direct
Teletype	Option	Std	Std
Std Clock	No	No	Yes
Price	$43K	$38.5K	$29.9K
Price of 2nd 4K Memory	?	9K	4K

32,768 Word Disk
$6K

FIGURE II: FEATURES ADDED IN THE EVOLUTION OF THE LINC AND THEIR MAJOR CHARACTERISTICS

1. Speed, memory size, A-D accuracy, teletype and price are all obvious.

2. PDP-8 Mode Programming:

   Compatibility with 4,000 other users and developments associated with the PDP-8.

3. PDP-8 I/O Bus:

   Off the shelf supportable options at a reasonable price.

4. Larger Display:

   When offered with half-size character and faster logic offers more extensive intercommunication.

5. Trap allows device independent software and interchange between different systems.

6. Buffered Tape:

   Allows continuous data collection and storage beyond core levels.

7. Clock:

   Allows compatibility between users and eliminates many specials.

   Offers much better clock at low cost.

LINC COMPUTER APPLICATION TO RESEARCH ON BEHAVIORAL CORRELATES OF EVOKED NEUROELECTRIC POTENTIALS OF THE BRAIN

W. R. Goff and G. D. Goff
Neuropsychology Laboratory
Veterans Administration Hospital
West Haven, Connecticut
and
Yale University School of Medicine
New Haven, Connecticut

In this symposium so far we have heard a brief description of the history of the LINC concept, the modernization of these concepts in newer computers, and an application to clinical problems in the quantitative analysis of Parkinson's disease. I will present a use of the LINC in a more basic research application. Our laboratory is engaged in both basic and clinical research.

The purpose of the experiment that I will use to illustrate our LINC application is to examine the interaction between two kinds of cerebral evoked responses. These evoked responses are a neuroelectric reaction of the brain cells to discrete peripheral stimuli, such as a click for the auditory system or a flash for the visual system, and these potentials are believed to reflect brain activity involved in stimulus detection, identification, association, and possibly even decision and memory.

The two kinds of evoked responses in which we are specifically interested are illustrated in Figure 1. The upper schematic diagram illustrates the sensory evoked response (SER), in this case, the evoked response elicited by a transcutaneous shock stimulus to the median nerve at the wrist. Note that this schematic has a bilinear time base illustrating that the potential changes during the first 70 or 80 msec are occurring more rapidly than the later components of the response. Also note that the total duration of this response is about 400 msec. The second type of evoked response, or perhaps it would be more accurate to use the term "event related potentials", is illustrated in the second schematic. It is the CNV or contingent negative variation. To evoke this kind of response, an initial stimulus, S1, which is a click in this particular experiment, is presented and the subject is told to make a response (e.g., key press) to a second stimulus, S2, when it occurs. S1 thus serves as a "warning" or "ready" signal for a response to be made upon presentation of S2. If we are recording with non-polarizable scalp electrodes and direct-coupled amplifiers, we will see after the sensory evoked response to S1 (seen at the beginning of the CNV trace) a steady negative dc potential which precedes the presentation of S2 and the subject's response. After the response is made, the negative potential gradually returns to the baseline. Since the occurrence of the dc potential is contingent upon the presentation of S2 - - if you do not present S2, you get the sensory evoked response to S1 but no dc shift - - the potential has been termed the "contingent negative variation" or sometimes the "expectancy wave".

Sensory evoked responses to single stimuli are obscured by what we call the "noise" of the on-going EEG or spontaneous cortical activity. The technique of averaging neuroelectric activity following repetitive stimulus presentation to enhance the time-locked potentials of the evoked response at the expense of the randomly occurring EEG is now fairly well known. Response averaging is one of our uses of the computer in the data acquisition phase of the experiment. The CNV can be seen in the unaveraged EEG in many subjects, but it is better resolved by averaging.

In this experiment we place non-polarizable electrodes on the subject's scalp. These are led through biological amplifiers, usually with a gain of 10,000, directly into the LINC's analog-to-digital converter. The problem for the computer is first, to summate from two electrodes, in other words sampling and adding two A to D channels, starting with S1, until sometime after S2. Sampling subsequent to S2 permits us to record the return of the dc potential to the base line. Since the program fills all of the lower half of the LINC's memory (1024 registers), only the remaining half is available for data acquisition. As we need four channels of recording, two for the CNV and two for the SER, we can allocate only 256 memory registers, that is one-quarter of the upper 1024 registers of LINC memory, to each CNV channel. We use a sampling rate of every 15 msec which gives us a total sampling of the CNV of 3.84 seconds. Secondly, in addition to CNV sampling we want to present a shock at three different intervals after S1 and upon presentation of the shock obtain from two additional A to D channels an average of the sensory evoked response. It is known that the occurrence of S1 will have a suppressive effect upon the shock evoked response as a function of the time interval between S1 and the shock. If we were to present the shock at only one fixed interval after S1, we would not know whether any sensory evoked response changes are associated with the suppressive effect of S1, or whether they are indeed correlated with changes in the amplitude of the CNV which is the information in which we are interested. By presenting shocks at three different fixed intervals after S1, the S1-SER suppressive effect will be varied and an analysis of variance can parcel out the effects of the suppression differentiated from effects associated with differences in the amplitude of the CNV.

For recording the SER, again we can allocate only 256 memory registers for each of the two channels. Thus, we use a varying sampling rate program which utilizes the fact that, as I pointed out earlier, the early potentials of the evoked response are changing more rapidly than the later ones. This varying sampling rate samples rapidly (every 500 usec) at the beginning of the response, slows down in the middle (every 1 msec) and then slows down still further (every 3 msec) for the final components. This permits us to sample 490 msec of activity after the shock using only 256 memory registers per channel. Thus we can record all of the CNV and SER information needed into one-half of the LINC's memory.

Two seconds after the presentation of the S1 click stimulus, we want to present a 10 msec tone pip having one of six different frequencies. These frequencies differ by 100 Hz from 1000 to 1500 Hz. The tones are generated by programming the LINC's audio output and are shaped by having the LINC trigger an electronic switch which shapes the tones into the required 10 msec tone pips without switching transients. The subject must identify the tone by pressing one of six keys and this identification is sensed by the LINC over its external level lines. Performing this discrimination helps maintain the subject's attention to the second stimulus which is important both for the development of the CNV and because variations in attention can cause alterations in the SER. In addition, there is some evidence that the amplitude of the CNV is correlated with the accuracy of discriminative performance and we want to gather additional information on this point.

If three seconds have elapsed after the presentation of S2 and the subject has failed to respond, the computer will terminate the pass and record a response failure. The computer also determines the correctness of the subject's discriminative response. It punches a paper tape which records the S1-shock interval, the tone frequency which was delivered as S2, the subject's response, and whether or not it was correct. This punched paper tape can subsequently be entered into another program which gives us a discriminative response profile.

It is obviously not feasible to go through the flow chart of this program in the time available. I will just enumerate the features of the program which I think might be of interest. Please take into consideration that some of this involves maneuvering to get this fairly complex program and a great deal of data into the relatively small capacity of the LINC.

First of all, the program demands via the teletype all of the relevant information needed for a response protocol including such things as subject identification, amount of response amplification, the number of stimulus conditions, the number of summations per stimulus condition, and so forth. Thus, when different people run the experiment you can be sure that they get all the necessary information and that the protocol of the experiment is standardized. The program then searches the digital magnetic tape where these evoked responses will ultimately be stored for subsequent data processing. It finds the first unused tape block and records its location. This avoids re-recording over any previously recorded data. There is an "end data tape" trap: the program computes the number of data blocks that will be needed to store all the data acquired in the experiment and if a sufficient number are not available it halts the computer and tells you that you need a new tape.

The program controls all the timing of stimulus presentation sequences. It samples continuously two A to D channels every 15 msec from the presentation of S1 until approximately two seconds after S2. It presents a shock stimulus at one of three intervals after S1 (there is also a fourth no-shock condition which allows us to evaluate the effect of presenting a shock *per se* on the CNV) and samples the sensory evoked response at the varying rate. It categorizes and stores the CNV and SER summations according to the four stimulus conditions. Two seconds after the presentation of S1, it presents S2 consisting of one of six tone frequencies generated by the LINC's audio output. It senses, determines the correctness of, and records the subject's discriminative responses. If there is no response in three seconds, it records a failure. It creates a punched paper tape which allows you to subsequently evaluate the subject's discriminative response profile. There is also a provision for the subject or the experimenter to interrupt the program at any time. Thus, if any problem develops in the condition of the subject or the equipment the program can be suspended until the problems are solved.

Because of the small memory, we cannot buffer store incoming response; they must be added to the previous summations directly as they are sampled. Thus, between each pass, if the stimulus condition changes, all four channels of previous summations, in other words the entire upper 1024 registers of memory are dumped on a digital magnetic tape working area. The previous summations for the new stimulus condition are then read into memory in preparation for adding responses to the next stimulus as they are digitalized by the A to D converter. Because the number of summations used in each stimulus condition would overflow the LINC's 12 bit word, summations of 16 responses (the maximum number that can be summed without overflow assuming a maximum A to D conversion on each pass) are temporarily stored on magnetic tape in a working area. They are subsequently recovered and entered into the average of the total number of responses using double register arithmetic at the end of each session. Thus, by utilizing the digital magnetic tape and punched paper tape to compensate for the small memory and small word size of the LINC, a rather complex data acquisition program can be run and large amounts of data can be stored. Subsequent to the experiment these data can be retrieved, quantified, analyzed, and plotted using "piggyback" programs, some of which are four times as large as the total 2048 registers available in the LINC's core memory.

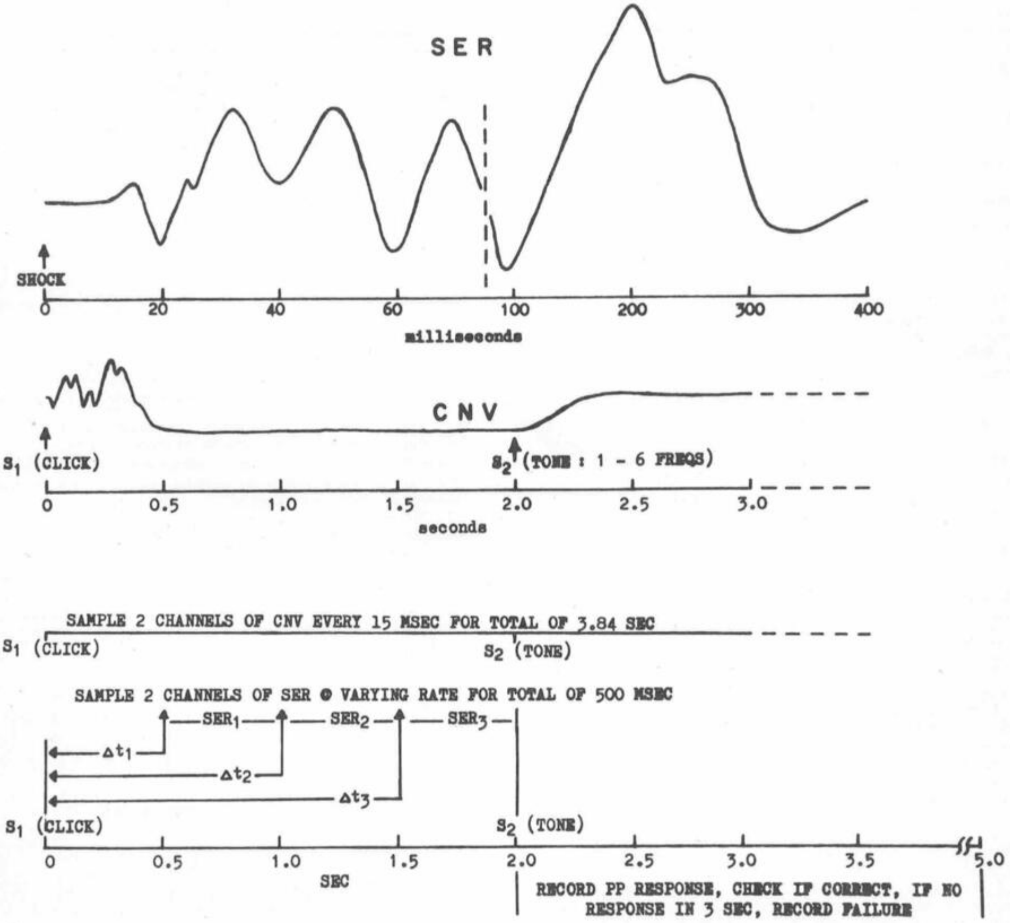


FIGURE 1

COMPUTER AUTOMATED PROCEDURES FOR ASSESSING SOME EXPERIENCES OF FAMILY MEMBERSHIP

William H. Sheriff, Jr.
Neurological Diseases and Stroke
National Institutes of Health
Washington, D.C.

In a way, this was an unusual application for us. The Section on Technical Development services both the National Institute of Neurological Diseases and Stroke and the Institute of Mental Health in the Intramural Research program at the National Institute of Health. We went from 1962 until last year with one Classical LINC, extended to 2K memory after a year of operation. As we operate with the equivalent of one full time programmer, we let the LINC out on a Laissez-faire basis; if you are willing to program it, you can use it, with the Section offering subroutine support and advice. For a three year period we averaged better than 17½ hours per week on a 7 day week using this LINC. If you take a two year Research Associate and give him free rein on a computer, you often come in in the morning and find a bleary-eyed Research Associate. This approach has proven quite productive. The members of our section attempt to only get deeply involved in projects that arouse our interest.

Although most of our work has been in Neurophysiology and Neuropsychology, I happened to get ego-involved in this problem brought to us by Dr. David Reiss of the Adult Psychiatry Branch, National Institute of Mental Health. He has been studying family interactions in Schizophrenia. Schizophrenics have been studied backwards and forwards for many years, but always in terms of the relationship between the schizophrenic and the therapist, or the relationship between a schizophrenic patient and the environment of the mental hospital. Dr. Reiss, along with Dr. Lyman Wynne and others, have decided that it would be much more fruitful to study the relationship between the Schizophrenic and his family.

Dr. Reiss started this line of research at the Massachusetts Mental Center prior to coming to National Institute of Health. His research unit is the family with an adolescent with a behavioral problem studied as a group. The usual grouping is the Father, Mother, Adolescent with a behavioral problem, and a normal adolescent sibling. NIH has a pool of such families living in the Washington, D.C. area that are willing to come in periodically as a group as research subjects. The subject has usually been a patient in the Clinical Center and is undergoing study and therapy by members of the Branch.

In past studies, Dr. Reiss has confronted these families with problem solving tasks, usually on the order of the analogy type problems found on Intelligence Tests, varying the degree of interaction (information sharing) between the members of the family. This procedure involved the experimenter in many mechanical difficulties, moving screens, sorting cards, and in general, acting as a traffic director. He could not adequately observe the procedure, nor, repeat the procedure in a standard, objective manner. He admittedly came to us reluctantly, a Psychiatrist coming to a computer, but he was forced into it.

He approached us with a specific problem for a pilot project, and a testable hypothesis. In previous studies he had found that there seemed to be two basic different modes that families used handling problems. One mode, an environment sensitive mode, the family reacts to cues from the environment and from other members of the family, weighing them with a degree of rational credibility in solving problems. Other families react in a consensus sensitive manner. When these families are working together, they tend to ignore or distort the cues that come to them from the environment if they conflict with the cues they receive from other members of the family. They rely primarily on the consensus of the family. It is as if they are defending their integrity as a family by distorting the rest of the environment.

Dr. Reiss wanted to set up a pilot project to test this hypothesis. The study used eight families, four rated as consensus sensitive families, four rated as environment sensitive families. Donald Uggla and James Bryan of our section designed an interface to remotely control five teletypes by the LINC. The experimental rooms were some distance from the computer room, and on another floor, so LINC was programmed to enable the experimenter to control the experimenter from one of the teletypes. The four experimental teletypes were set up in a sound proofed room. The experimenter's teletype was set up in an observation room that overlooked the experimental room through one way mirrors. The experimenter could see and hear the subjects, they could not see or hear the experimenter. The subjects were seated in a row at their teletypes in the following order: Mother, Subject, Father, Sibling. Movable screens were in the room to block the subjects from view of each other during the experimental phase of the problem.

The interface was designed to allow a character from the teletype to be held by a buffer, one channel for each teletype, and gated into the LINC accumulator with an OPR command, the code in bits 0 – 7 and the teletype number in bits 8 – 10. The output data was written directly by LINC in its serial format with the appropriate channel code. The LINC was interrupted by the control logic every 9.1 milliseconds and the bit was transmitted via an OPR line. A single bit must be transmitted at every interrupt, except at idle conditions at logical 1. The write cycle receives top priority in the program, since the read cycle is significantly faster. We decided to use a message-in, message-out storage space in upper memory, only sending, or acting on, completed messages. Upper memory was also needed for one of the problem conditions, limiting us to a maximum of 32 characters per message. If I had it to do over again, I would act on individual characters received or sent, rather than messages. Even then, there would not have been enough space to control all the contigencies, such as a too-long message inserting an end-of-message flag in someone else's message area. Such things can only be controlled by good instructions and cooperative subjects with a computer this size.

We used an interrupt routine to receive and send individual characters. This routine scanned the message-out area for flags, only sending a message when completed. It placed incoming characters in the appropriate message-in area, putting a message-complete flag in when it received a carriage return code. The main line routine scanned the message-in areas for flags, acted on the messages, evaluating and scoring, then distributed the appropriate messages to the various message-out areas. The procedures depended to a great deal on the cooperation of the subjects and we were very pleased to find that the subjects did cooperate, beyond our expectations. The teletype keys were modified allowing only the letters of the alphabet and three control keys, end of message, error, and one that doesn't really matter for this presentation.

The problem was presented in three modes, explained later. The following procedure was constant for all modes. The family was given a sample trial. First they were given instructions on the procedures. Then they were given an example of a logical sequence, for example, a "C" followed by two "X's" then a "C". They were then allowed to test different sequences on the teletypes, using the particular presentation mode, in the family order mentioned previously. In this case, whenever they typed a "C" followed by any number of "X's" followed by a "C" they received a "+" indicating a correct response. All other responses were graded with a "–". The responses of the individuals were distributed according to the Model format, and all members of the family were requested to communicate freely so they would know the manner in which the responses were distributed.

For the experimental condition, the families were isolated from one another by the screens and instructed not to communicate. They were each given a printed sheet with one correct sequence, and ten test sequences. They were asked to evaluate the ten test sequences with paper and pencil, turn them in, and then proceed with the teletype routine, exactly like they did in the sample problem. After 22 trials on the teletype, they were again given ten examples to score by paper and pencil, to give some indication of learning during the experimental procedure.

In the sample problem, they were all given the same example. In the experimental procedure, two members were given one example, and two members another, each set involving a different logical sequence. Success in the problem was determining the fact that there are two correct sequences and solving the sequence not given.

The three modes were modes of distribution of the responses of the family controlling the amount of information the family shared. The first mode with maximal sharing is shown in Figure 1.

All teletypes line up under the column of the subject whose turn it is to type. He types his solution, it is immediately graded, and distributed to all the other members of the family. Thus all members of the family share all of the information transmitted by the individuals. All of the printouts are identical in this mode. This mode is called the Public Mode.

The second mode, in order of information transmitted is called the Anonymous Mode, Figure 2.

On this mode, the subject is informed of his turn by having "your turn" typed on his teletype. He types his response, it is graded and shown to him, but is stored in memory and not distributed this time to the others. Thus, on the first turn, no one gets a response from any of the other members. On the next cycle through the family, as a member types his response, it is again stored in memory, but a different response from the previous cycle is distributed to each other member of the family, so that every member of the family gets a different response from the previous cycle's responses of the family. After the end of any cycle, every member of the family has received all of the responses made by the family on the previous trial, but has no way of knowing whose response he is looking at. Thus, they are all sharing the same information, but do not know who transmitted the information.

The third mode, the Standard mode, has minimal information shared by the family. (Figure 3).

```
                PUBLIC MODE
       Printout from family in study

MOTHER    SUBJECT   FATHER    SIBLING       ┌──────────────────────────────────┐
                                            │             EXAMPLE              │
                                            │   PARENTS - VDMDMDMDMDMV         │
                                            │   CHILDREN - VMMMMMMMVV          │
                                            └──────────────────────────────────┘
VDMDMDMV+
          FCCCHH-
                    VFMFMFMVV-
                              VJCJCJCVV-
VDMV+
          BHBHCC-
                    ABCD-
                              VDMMDDMDV-
VMDMDMDMV-
          ERERERE-
                    VDMV+
                              VDMDMDMDMDMV+
```

(Figure 1)

```
              ANONYMOUS MODE
   printout from family in study - member A (MOTHER)

                                      ┌──────────────────────────────┐
                                      │          EXAMPLE             │
YOUR TURN                             │ PARENTS - CXXXXXXXC          │
XCCCX+                                │ CHILDREN - XCCCCCCCCX        │
                                      └──────────────────────────────┘
YOUR TURN
CXXXXC+

+XCCCCX
-VMMMMV
+CXXXXXC
YOUR TURN
CXXXXXXXXXXXXXXC+

+CXC
-VBBBBV
-WFFFFW
YOUR TURN
XCCX+

+XCX
-ZXXXZ
-CX
YOUR TURN
XCCCCCCCCCCCX+
```

(Figure 2)

In this mode, "your turn" is typed on the teletype of the appropriate member of the family. He types his response and is given his grade. Then, responses stored in memory are distributed to the other members of the family, no member getting the same response. These responses were predetermined by the experimenter from previous frequencies of correct and incorrect responses on this type of problem. All families were given the same sequence. At no time does any member of the family have access to any information from another member.

It was predicted that the consensus sensitive families would perform poorly on the Public Mode, in which they would have maximal conflicting, shared evidence, fair on the Anonymous Mode, and best on the Standard Mode. Environment sensitive families should perform equally well in all Modes. The predicted direction is illustrated in Figure 4.

Analysis of the pre and post test inventories tended to confirm the hypothesis.

To conclude, I would like to read three observations made by Dr. Reiss as major advantages over non-computer methods:

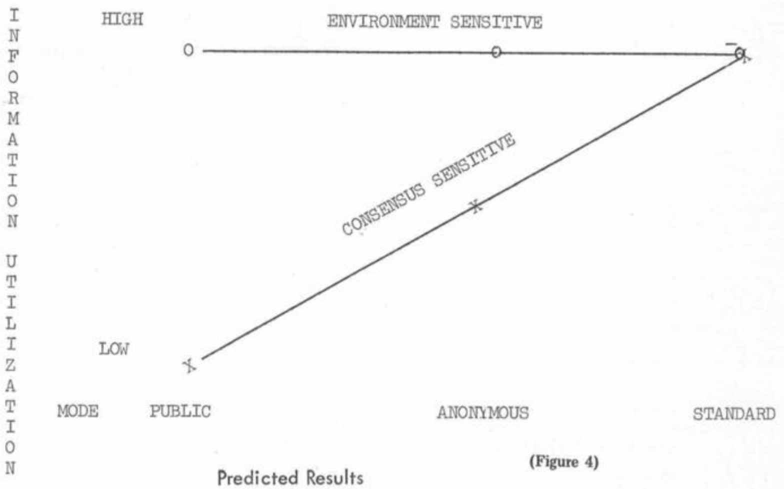
1. Observer rating scales, long bedeviled by problems of reliability and validity, can be eliminated since all of the family' performance is recorded automatically.
2. The experimental procedure is precisely standardized and can be exactly replicated using a copy of our program tape and the same standard teletypes and computer equipment.
3. The presence of the human experimenter has been partially eliminated and the prospect is good for his total unemployment.

We hope to soon eliminate the human element in this procedure, but, as it stands, it severely strains a 2K memory, even with microtapes. We expect to get about three years good usage out of these procedures before transferring to an anticipated larger computer system.

```
          STANDARD MODE
     printout from family in study - member A (MOTHER)

YOUR TURN
CTC+

LUV-
CTTTT-
CSC-
YOUR TURN
TCT-

CSTTTTTTTTTTS+
CSSC-
CCTTTTTTCC-
YOUR TURN
CTTTTTC+

CSTTTS+
CTSSSST-
CSTTTTTTTTTTTTS+
YOUR TURN
CSTTC+
```

```
            EXAMPLE
PARENTS - CTTTTTTTC
CHILDREN - CSTTTTTTTTS
```

(Figure 3)



Predicted Results

(Figure 4)

COMPUTER TECHNIQUES FOR STUDYING BIOLOGIC RHYTHMS: QUANTITATIVE CHRONOBIOLOGY

Charles F. Stroebel, Ph.D.
Director
Laboratories for Experimental Psychophysiology
Institute of Living
Hartford, Connecticut

Chronobiology is the quantitative study of biologic rhythms — those biologic phenomena which fluctuate periodically over time. These rhythms, sometimes called biologic clocks, can vary in their period of oscillation from milliseconds (e.g., biochemical reactions or unit nerve activity) through the entire time domain to hours, days, years, or longer. By specifying the periods of individual rhythms along a continuum, a spectrum of bio-periodicities may be envisioned as shown schematically in Figure 1. Some regions of the spectrum have been more fully studied than others, either because of prominence (the sex cycles) or ease of study and demonstrable clinical relevance (EKG or EEG).

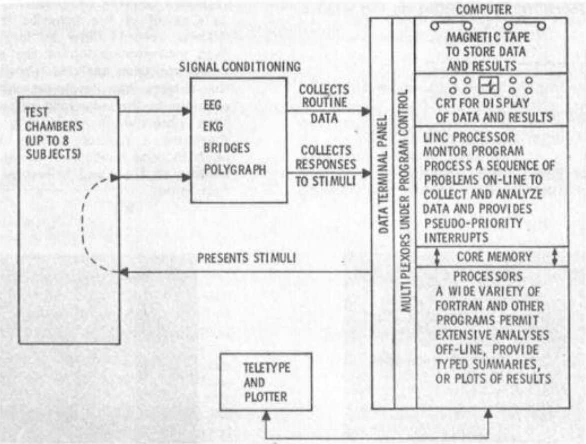


Figure 1

Comparatively recent demonstration of the importance of other domains in the bio-periodicity spectrum (Halberg, 1969), argues strongly for the development of practicable measurement and computer techniques to permit large scale quantitative analysis of their relevance in normal and abnormal human functioning. Examples are the activity-sleep cycle, the dream cycles, the times of maximum and minimum susceptibility to pharmacologic and infectious agents, rhythm adjustment to intercontinental jet and space flights, and various abnormal rhythms associated with emotional illnesses. (Stroebel, 1969).

This paper will discuss techniques for studying long and short period rhythms with a LINC-8 class computer. Several of the applications should be of general interest because of their flexibility in permitting this relatively small computer to operate 'simultaneously in the following modes in a relatively automatic fashion: batch, process control, and interactive in a dedicated experiment.

PROBLEM

Because of inherent or superimposed noise which is common in most biologic signals, several or many cycles must be observed to permit quantitative estimation of a rhythm's period, amplitude, and variability. While this sampling requirement poses relatively few problems in studying ultradian rhythms where many cycles occur over a short period of time, it may require full dedication of a LINC-8's capability for the period of data collection and analysis (e.g., EEG Spectrum analyses, average evoked response studies, analysis of operant behavior for periodicities) (Weiss, 1967). Measurement of slower circadian (circa, about; dies, day) or infradian rhythms requires less frequent sampling over longer time intervals — an uneconomical application for a dedicated LINC-8; a more economical approach would collect data off-line to be batch processed when the computer is not dedicated to more demanding tasks. Our specific problem, however, stemmed from a desire to use the computer as an interactive component in the longer rhythm experiments, collecting and analyzing data on-line to make decisions as to when a behavioral trial should be presented or when and how much of a medication should be infused based upon the subject's own biologic clock time.

To summarize our requirements, we sought a means to accomplish the following objectives concommitantly in a relatively automatic, flexible fashion using the LINC-8:

1. To commit the computer to "full-time" interactive experiments (capable of brief interruptions) such as average evoked response studies.
2. To sample a varying number of human and monkey subjects for a varying number of telemetered physiological variables four to five times each hour, testing incoming data for artifacts and against limits established from previous days of data for a given subject; to appropriately store the data, and to make decisions as to when to deliver behavioral trials or medications.
3. To service interrupt device alarms using the external sense lines.
4. To automatically que and process various lengthy batch jobs (e.g., CalComp plotting of data when computer time becomes available).

METHOD

The foregoing objectives were met by writing a software monitor (called MONTOR (Tuttle, Pavel, Stroebel, 1968)) for the LINC-8 which executes GUIDE filed programs automatically in a sequence specified by typing job programs and control programs into the monitor index via the teletype. The control programs permit flexibility by providing the following generalized functions:

1. Repeat the above program.
2. Jump the next program on a (specifiable) external line level change.
3. Jump the next program after previous programs on the list have been executed XX times.
4. Interrupt this program and save it for restart after the interrupt is processed.
5. Halt.

Variables to be sampled with slow rhythm changes (e.g., core, temperature, heart rate) are sampled and [redacted] evaluated once every twelve minutes by an interrupt presented by an external clock. All programs likely to be interrupted by the external clock contain skips at logical points in the program flow where the program, counters and appropriate registers are saved in a storage area on tape, the sampling interrupt serviced, and the interrupted program restarted.

Figure 1 shows a diagram of the general system, whose control components are illustrated photographically in Figure 2. During a typical 24-hour period, core temperature rhythm data is collected, analyzed, and stored requiring 30 seconds once every twelve minutes, day and night. For the remaining of the eleven and one-half minute intervals during the day, the computer tests and measures behavioral performance of the subjects or is used for program development; the eleven and one-half minute intervals at night are used to accumulate average evoked response data by stage of vigilance and circadian phase for a single subject and/or to analyze and plot data previously collected on the GalComp plotter. Once the monitor has been initialized, no further console interaction is required. Redundant error detection programs both internal and external to the computer provide emergency notification by telephone to a laboratory staff member on call when a failure occurs.

RESULTS

Our LINC-8 has been operating 24-hours daily under MONTOR control for over a year. Failures attributable to minor "anomalous" computer malfunction have occurred on the average of once every two and one-half weeks. Hardware failures requiring field maintenance are less frequent. Because disc storage has not been installed, MONTOR performance can be optimized by giving careful attention to location of programs and data on LINC-tape; otherwise, time spent in tape searching can be considerable.

During the year considerable data has been collected to document two kinds of biologic rhythm abnormality associated with disturbed behavior in the Rhesus monkey. Figure 3 is a vertically isometric LINC-8 scope display showing desynchronization of the brain temperature rhythm in a monkey developing psychosomatic symptoms (asthmatic breathing, neurogenic skin lesions, gastrointestional disturbances) during the period of desynchronization. The figure shows 48-hour brain temperature records with samples every twelve minutes; the vertical dotted lines indicate the onset of a twelve-hour light period each day. The desynchronization of the rhythm becomes apparent in the seventh curve from the bottom; resynchronization with the light cycle is evident in the fifth curve from the top. During the interim, the subject's body slowly becomes prepared for night functioning during the day and day functioning during the night, a phenomena which would manifest itself clinically as insomnia.

Figure 4 shows a vertically isometric LINC-8 scope display of the brain temperature rhythm in a monkey developing more disturbed, regressed behavior (despondency, movement stereotypy, loss of appetite) as a result of the behavior frustration-stress. The vertically displaced 48-hour records show an increased amplitude of the first 24-hour peak with suppression of the peak on the second day of each record. Power spectrum analyses (Figure 5, panel 2) of these data suggest that the subject has developed 48 and 96-hour rhythm components in response to the behavioral stress. Comparable rhythm abnormalities have been observed in psychiatric patients. The remainder of Figure 5 illustrates a further finding, that six weeks of treatment with a phenothiazine medication serves to suppress the "abnormal" longer than 24-hour rhythms and to restore relatively normal behavior and circadian functioning.

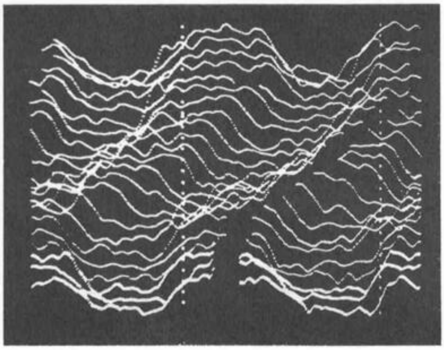


Figure 3

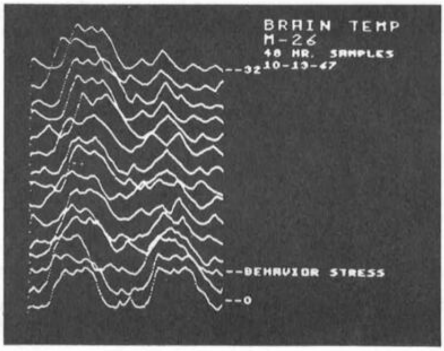


Figure 4

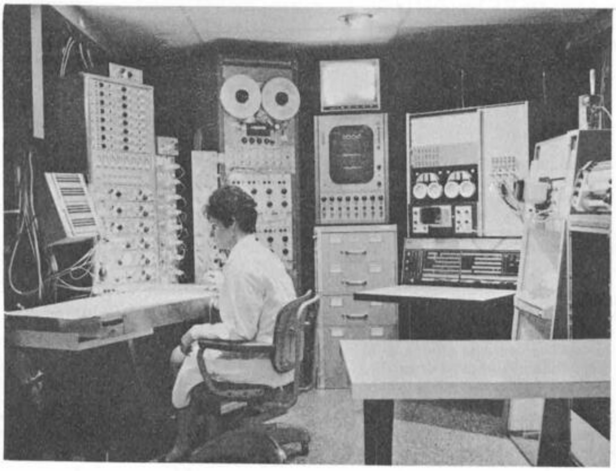


Figure 2

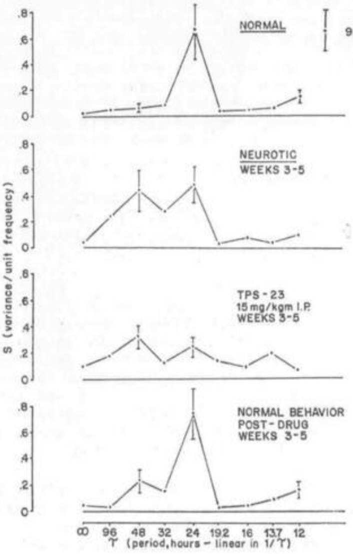


Figure 5

Other experiments suggest that the recovery process may be accelerated by phasing drug administration with the subject's own biologic clock time (as determined by the LINC) rather than by administering the drug in the usual clinical fashion at the convenience of staff. The evidence further suggests that appropriate timeing of such medication can maximize the active drug principle and minimize undesirable side effects which are common with psychiatric medication. We are currently impressed that correction of the biologic rhythm disturbance using the interactive computer approach inevitably seems to precede behavioral improvement, suggesting a potentially important new role for computers in the psychiatric treatment process. (Stroebel, 1967)

The techniques discussed here for using a LINC-8 in a chronobiology research environment represent only the initial steps of a complex data analysis sequence (Stroebel, 1969a). Similar approaches for data collection using a different computer have been developed with considerable ingenuity by Dr. W. Runge (1969) at the University of Minnesota, where Dr. Franz Halberg has developed sophisticated statistical programs for subsequent analysis (1968) referenced to international biologic rhythm standards.

BIBLIOGRAPHY

Halberg, Franz. "Chronobiology." In *Annual Review of Physiology 31,* 1961, pp. 675-725.

Halberg, Franz and Hamburger, C. "Electronic Computer Techniques for the Study of Endocrine Rhythms in Psychoendocrinology" *Acta Endocrin. Cong. Acta Endocrin., Kbh. Suppl.* 100:170 (1965).

Runge, W. "Some Instrumentation for Data Collection in Chronobiology." In *Proceedings of the Second International Symposium on Experimental and Clinical Chronobiology*. In press.

Stroebel, C. F. "Procedures for Data Transfer in Chronobiology." In *Proceedings of the Second International Symposium on Experimental and Clinical Chronobiology*. In press.

Stroebel, C. F. "Biologic Rhythm Correlates of Disturbed Behavior in the Rhesus Monkey." In Rohles, F., ed. *Circadian Rhythms in Nonhuman Primates Bibl. Primat.*, Number 9, pp. 91-105. Karger, Basel/New York (1969).

Stroebel, C. F. "Biologic Rhythm Approach to Psychiatric Treatment." *Proceedings of the Seventh IBM Symposium* IBM Corp. Yorktown Heights, New York (1967).

Tuttle, R., Pavel, K., and Stroebel, C. "A Monitor for Automatic Sequential Operation of Programs on the LINC-8." Decus Library.

Weiss, B. "Digital Computers in the Behavioral Laboratory." *Proceedings of the 1967 DECUS. Biomedical Symposium.* Digital Equipment Corp., Maynard, Mass. 1967. pp. 31-44.

DISCUSSION

DR. GOFF: Can you comment on new software for the PDP-12?

DIAL RICHARD CLAYTON: The most significant initial aspects of the software of the 12 is a new operating system including assembler, editor, and various peripheral programs to handle filing, and program conversion, etc. That is currently dubbed the DIAL System, or Display Interacting Assembly Language. The program is patterned in the interactive sense, very much after the LAP6 program which Mary Allen Wilkes distributed some two years ago, and the user characteristics, at least in an editing sense, are very similar. The assembly characteristics are slanted very much toward the PDP-12. Specifically it is an assembler which handles either of the two order codes, and the format for assembly, and henceforth the format with which one writes source programs - the format for the assembler is very similar to the one commonly used throughout the DEC machines. Specifically, 6 character alphanumerics plus slash for the comment delimiter, large character i as the indirect bit, and different symbol characters. This departure from the Classic LINC format we felt obligated to go to on trying to bring the two machines together in a single assembly language which would allow rapid convenient assembly of programs in either or both modes. In detail, the program allows within core some 180 user defined symbols, outside of the instructions that are part of the symbol set. Beyond 180 symbols, the tape is used as the symbol area yielding, of course, a substantial decrease in the assembly rate. As soon as one goes to mass storage for the symbol table (even if one does very clever things for symbol searches if you have to go out to some mass storage in the form of LINC tapes or DECtape) the assembly slows down fast. To that end, an additional 4K of core or disc will be useable for the symbol table. The 8K version is in the program, but not at the moment running. We will be running DIAL at Spring Joint for those of you who are there. The machine that is there is in effect a production type machine. There will be workable versions of DIAL with the first PDP-12's that go out, which are still a couple of weeks away. That will be an evolving program and I think we'll see quite a few changes to that in the form of additions in the next year and a half.

SEVERO ORNSTEIN: Dick, I think it was well pointed out that one part of the laboratory instrument meaning of the LINC was the terminal frame. What have you done to it? (data terminal panel assembly).

IO RICHARD CLAYTON: It existed in the form of two large plug-in units in the Classic LINC. In the case of the LINC-8 it consisted of about 50 unused slots mounted behind the door in the upper right hand side of the front of the LINC-8. What happened to that on the PDP-12? To some degree, it has gone "bye-bye" a little bit anyway. What we have done in that is as follows: we have put a large number of options prewired into the machine itself, so a number of some of the things that went into the data terminal panel assembly, such as plotters, additional teletypes, certain things with display and A to D, these have been prewired within the machine. This is something that essentially gives an economy of manufacturing and allows the user to request of the manufacturer that these be supplied. Further, it makes it practical to supply them at a reasonable cost, and with a minimum of labor for their installation and expansion of the machine. For other kinds of options, again available literally "off the shelf," are for example, IBM compatible tape, things of this sort; these are also offered as standard options and plug onto an I/O bus.

old IO The data terminal panel concept as originally implemented in the Classic LINC allowed up to two boxes to be plugged into the machine, and used some predecoded operate lines. What one ran into was 1) a limitation in the total number of device codes, if you like, available for easy interfacing, and the interface structure was extremely simple, but was not a bus structure, we would in technical terms now call it a radial structure in that it had several sets of ports into the machine. Once those were used up in the simple straight forward way, you were all over, and that was in general the way one implemented the first interfaces one designed. Beyond that, one had to make a bus structure out of one of these radial ports, so that several devices could be put into the SN and TN lines of the Classic LINC. This then meant that at some point the interface had to be redesigned if one wanted many interfaces and a number of Classic LINC's have gone that route. The basic concept of the PDP-8, LINC-8, or PDP-12 I/O bus is that of a LINC I/O bus structure which is independent of however many devices are connected to it, and the cables for that interface connection go from device to device. And in this case, a device consists of a mounting panel of logic with or without its own power, and as one adds or takes off devices, they have any one of 64 device codes which gives you potentially up to 64 devices, and the machine structure becomes rather independent of what combination of peripherals are on the machine. It is not quite as convenient to plug and unplug these devices as it would the data terminal panels of the Classic LINC, but the independence in intermixing of options is much more practical, as in the implementation of these in a production sense, specifically the ability to buy these options independent of what you currently have and our ability as a manufacturer to again, supply them, again independent of the current status of your machine. And it is with that philosophy that we do not have the mounting panels per se.

What happened with the LINC-8 was that a number of people put nice little interfaces in the data terminal panel, for one or two things, but then if they wanted to make an elaborate interface they eventually went to separate mounting panels anyway, and the door and panel were rather cumbersome. Now the plug-in option becomes even more cumbersome and more expensive in many cases, and so what we've done is, there is still some room inside the machine itself, and the mounting panels for creating this kind of an interface are readily available from our standard catalog of logic parts, and the interface is again described as a bus-type interface. Does that answer the question?

DR. STROEBEL: I would like to ask Mary Allen Wilkes a question. Possibly one of the great successes of the LINC concept today has been the software which has been written by quite bright users in the field. It has become available through one or another various channels. Is there any way you can see of furthering encouraging this flow of software so that new application programs become more readily available? You have certainly been concerned about this for a long time; for example, your Washington University Group under the direction of Wes Clark maintains a bibliography of all papers published as scientific literature which involve some kind of LINC processing. Do you have any thoughts on this?

M. A. WILKES: Well, I think one thing that we've learned is that it is not easy to keep the program flow, as you refer to it, going successfully between users. I should perhaps clarify one thing - when we started the bibliography that you referred to, we were not primarily interested in acting as a program information exchange group. DECUS does this already, and I still encourage people to submit programs to DECUS. We were interested in keeping track of the documented research articles that described finished work, or work in progress that would be of interest to other scientists in the same field, or to scientists working in different fields, but who are interested in using computers in the same way. This has been valuable because most of the journal articles listed on the bibliography do not in fact mention the computer itself at all, and that's fine. That means it has properly been integrated into the laboratory and is being used as a research tool. But it is very difficult for new LINC users to track the information down without a bibliography.

As to the program flow business, I'm sure someone from DECUS is better able to answer to this. We do not attempt to exchange programs at all between people. If they see something on the bibliography that looks as though it is something they can use, then, fine, they are encouraged to write to the author or, if it is available through DECUS, to write to DECUS. We cannot maintain tapes, or other people's documentation.

I would certainly say this about distributing programs. And that is, there is no point in trying to distribute a program which has nothing written down about it. And there are a number of people who write programs who seem to think that because a program is useful to them, and they know how it works because they wrote it, or the guy who sits next to them wrote it, they seem to think this is a tremendous benefit to someone in California. There is someone way back in the literature in the field, who said that 'if it is not written down, it doesn't exist.' and I'm afraid that's how I feel about computer programs. The

---

† *The name DIAL has been changed to LAP 6-DIAL, because of the similarity of many of the operating features of the editor portion of the system to those of LAP-6. Although the present name is LAP 6-DIAL, the responsibility for all development, corrections, and distribution will rest with Digital Equipment Corporation.*

documentation has to be plain, it has to be complete, and it has to be such that someone can run the program on his own, by using only the information he can get through the mail. So, if you want to exchange programs, I would first encourage this kind of attitude toward documenting them.

DR. STROEBEL: Would publication of programs in some more formal kind of scientific publication to permit legitimate publication credit encourage this kind of documentation?

M. A. WILKES: I'm not sure. I think the programs go with the instrument itself, with the computer as part of the tool. The two work together to hopefully do some useful work. If there are programming techniques that are learned in writing the program, these are perhaps properly documented in some of the computer journals. I don't think that the detailed workings of a program should be documented in the scientific journals, although you see it occasionally, but, I don't think this is too useful. A lot of teaching of some of the "higher level languages," for example, is based on information that is available only in journal articles, and this information is only descriptive of the language. It does not attempt to tell you exactly how to use the language or to write programs in the language, and these descriptions are almost impossible to use as programming guides. So from that standpoint, I wouldn't encourage it, not when it encourages an author to be just a bit more general than he should be about the way the program, or the language, actually works.

S. ORNSTEIN: I have a comment. I can't let Dick [Richard Clayton] get away quite as easily as he did when he was talking about the data terminal panels, because I think that one of the things that we're seeing here, as a matter of fact, that worries me a little bit, is the standardization that we very carefully built out of the first machine. And I see it creeping in and I don't like it very well, because the whole theory underlying our data terminal box was that we didn't know, as I say, what we were going to connect to, and it was put out there specifically to make it easy, not for the manufacturer of the machine, but for the user who wanted to use it for his special clear kind of thing.

It will hook up to his special apparatus that nobody else in the whole world could conceivably understand, and I understand that it's easier now to get the parts to plug-in so that we interface to this system or the IBM tape system, and that's fine if that's what half of them want to do. But, I think that this whole basic concept was that we were in fact putting a computational device in the laboratory with a whole lot of other apparatus that we didn't understand, and so make it easy for people to adapt the LINC to their own unique requirements, we left the data terminal as flexible as possible. I would also like to ask really why you did choose to eliminate it? Was it just packaging convenience, or why did you choose to put the machine all into one piece as you have. Because I have heard people say that's not quite so handy. Would you like to answer that?

Io

R. CLAYTON: Before I do, I understand Severo's concerns here on the data terminal panel question, and they are very valid. I think I indicated our attempt there is to a form where readily available components to make up whatever type of interface is appropriate for the special equipment within the laboratory - and on the context of the I/O bus, with readily available cables, and panels to do it from. We also don't feel that we can specify how big that is, and that unless we force everybody to buy a long, large one, we then get into some other types of limiting problems. In addition, I would like to make one point. We have now proceeded quite a ways in the last six years, to a point where a number of users are very interested in the machine as a tool, and much less interested in the interfaces per se. At least for a large number of peripherals, many of which are quite common. And that's what leads to the desirability of having a number of peripherals readily available at low cost. On the question of the packaging, there are some very obvious economies involved in not interconnecting machines together with elaborate cable structures, basically. The cable structure of the Classic LINC was a rather expensive one, and added quite a bit to the manufacturing costs, and to some of the subtleties and difficulties of the check out and operation of the machine. We also felt that it imposed very severe restrictions on moving the machine.

DR. STROEBEL: Dr. Hiltz has agreed to share some of his experiences with the use of process control computers in neurophysiology applications.

DR. HILTZ: Thank you. As active users of small computers in real-time biological work, you might find some interest in similar applications of the larger breed of real-time computers usually known as process-control machines. Not long after I wrote the DATAVG on-line signal averaging program for the LINC-8 I went to Cornell, where an IBM 1800 had recently been installed in a laboratory for radiation biology and neurophysiology. My first interest there was to write a similar signal averaging program for the 1800. A comparison of the two programs is interesting in itself and should bring out some more general differences between the two classes of computer. Perhaps a word about the process control computer per se is in order first.

These machines look rather attractive for real-time lab work. They are designed to run subroutines in response to "outside" events, rather than under direct control of an "inside" program. To this end, they make heavy use of a priority interrupt structure and I/O devices which run on data channels, that is, independent of program control. Analog and digital input and output are readily available.

However, the design of these computers to control processes such as steel production or oil refining leads to a philosophy that is not particularly attractive in the research laboratory. I speak of the 1800 now, but it is typical of the process control computers. The idea of protection is important in process control work. This shows up in two important ways. First, the several "users" of the computer will include different parts of the controlled process as well as programmers debugging new routines and perhaps something like accounting going on in the background. All these users must be assumed to run completely independently of one another and therefore must be completely protected from one another. This is accomplished by a sophisticated monitor, or executive system which assigns sections of core storage and different I/O devices among the users and prevents each from using what belongs to the others. The monitor also provides graceful recovery from errors by the users. All these goodies naturally require a lot of monitor, in the case of our 1800, about 6500 words out of the 16000 available.

The second aspect of protection is protection of the process. Once the computer is in control, it had better stay in control, or else we are likely to find ladles of steel running off the ends of their tracks, etc. Thus, for every interrupt there must be a service subroutine. Of course, we use a dummy subroutine for all these in our lab, and attach our working programs only to the interrupts which they use. This is fine for the first attempt at a program, which is of course full of bugs. When we have a new version, we can't simply delete the old one and put in the new. As far as the monitor is concerned, that would leave some part of the process uncovered for a time, so it won't permit deletion of a real-time program. We must instead replace the old version with a dummy, then compile the new version and replace the dummy with the new version. This takes only a small handful of control cards. Protection of this sort is just not applicable in the lab environment; we have spent a great deal of time working around the many ways in which the system protects the user both from himself and from others, of which the above is just one example.

Let's return to the signal averaging programs as examples of how the two kinds of computers can handle a lab problem. The LINC-8 program DATAVG runs a loop which displays the accumulating sum of signals while waiting for a trigger pulse on a sense line. Then it samples an analog input voltage, which places the value in the accumulator. From there it is added to the accumulating sum in the core, which is then scaled and displayed right from the accumulator. Simple. Direct. All this action except the display may be inhibited by the occurrence of any of several contingencies, made known via analog inputs or sense lines, so (for example) only responses evoked during positive EEG voltages will be included in the average.

The 1800 program is of necessity somewhat more complex. The trigger signal activates an interrupt. The monitor executes 169 instructions in delivering control to my program from that trigger pulse. Naturally we can't afford to wait that long, so I have taken over the hardware interrupts. These must remain in core, however, to handle interrupts from the system I/O devices such as the disk, which I don't care to program myself. The trigger interrupt service routine starts the A-D converter. Because the converter operates on a data channel independent of programs, it of course cannot put its sample value in the accumulator. Rather it leaves the value in a table previously established in core and sets a flag when done. The program waits for this flag, during which time it can't do much else, then brings the analog value from core into the accumulator, from which it is added to the accumulating sum. Our CRT display operates on another data channel, taking its data from a table in core. For display, the program would have to scale the accumulating sum, set some bits which give microcoded commands to the display and deposit the result in the display table. It turned out that there was just not enough core for yet another table, so the live display was dropped. Inhibition of averaging based on external contingencies was discarded as too awkward and time-consuming to be worthwhile, basically because the program runs so automatically.

In short, for this type of program at least, the LINC-8 runs rings around the 1800. The comparison looks like this:

	LINC-8	1800
Core for storage of accumulating sums	2048	2560
Maximum number of analog input channels	8	16
Maximum input rate, samples per second	11000	8000
Live Display	Yes	No
Display of a reference signal waveform	Yes	No
Display after averaging ...	Yes	Yes
Adjustable display format .	Yes	No
Inhibition of averaging on contingencies	Yes	No
Computation of standard deviation of the mean ...	Yes	No*
Back averaging (signal precedes trigger)	Yes	No

** Standard deviation computation later added to 1800 program.*

continuous sampling

Both programs use a keyboard-interactive set-up routine, and both make extensive use of program overlays from mass storage.

In answer to the question about the future of the LINC-8 program DATAVG, I have been asking people about whether this should be improved for the PDP-12. I don't know yet whether we should work out a PDP-12 version. The only real differences would be to take advantage of the faster A-D converter and the built-in clock. The LINC-8 version requires an external clock or some sort of pulse generator in the laboratory to connect to one of the sense lines.

QUESTION: Are there any positive things you can say about the 1800 as a typical process control computer?

DR. HILTZ: Yes. It's nice to have all that core and it's nice to have FORTRAN. Even in our on-line lab work, at least half of our programming is of the sort that can be done very comfortably in FORTRAN. The inefficiency of assembler coding is, of course, the amount of effort between the perception of the problem and a workable solution in the computer. Her FORTRAN is much better. I would want a PDP-12 with 8K of core so I can run such programs in FORTRAN, and I would hope that the assembler in DIAL would offer the ability to link subroutines with FORTRAN calling programs.

FORTRAN

DR. STROEBEL: Most LINC-8 users may not realize that a carefully documented procedure is already available to combine LAP programs with FORTRAN routines which are processed by the PDP-8 part of the computer. The procedure, called "LINC-TRAN" has already been published as program L-21 in the DECUS Library. The general strategy used by LINC-TRAN on the LINC-8 should be considerably simplified for the PDP-12.