c.1

THE DEVELOPMENT OF A MICROCOMPUTER - BASED PERIOPERATIVE PATIENT MONITORING SYSTEM

by

Neil B. Cox

B.Sc.E.E., University of Alberta, 1978

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENT FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Department of Electrical Engineering)

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

December, 1980

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date Dec 19/80

-6

ABSTRACT

A computer-based perioperative patient monitoring system, called the Mobile Operation Monitoring for Anesthetists or MOMA, is discussed. The primary objective of this work was to produce a flexible, mobile prototyping system which could be used to develop a clinically useful tool to aid the anesthetist in the task of patient maintenance during anesthesia. Potential application areas are discussed along with the limitations and necessary design considerations.

The system developed for this thesis uses a PDP-11V03-L microcomputer system with an A/D converter and dual double-density floppy disk drive to acquire, process, and store four channels of EEG along with up to four channels of non-EEG data. The data is displayed on a Tektronix 4025 video graphics terminal in a number of user-selectable display formats. A currently popular EEG display format (Density modulation of the compressed Spectral Array or (DSA)) is modified to simplify the determination of long-term EEG trends. This display format is presented along with non-EEG data in order to facilitate comparisons between parameters. The user can recall any EEG display which appeared earlier in the operation without interfering with data acquisition, processing, and storage. The keyboard is configured in an easy-to-use format with the keys clearly labelled and grouped according to their function. Each command is generated by typing a single key. A facility for the entry of comments through the keyboard was implemented so that the time of occurrence of certain significant events during surgery, such as pharmacological intervention, can be tagged. These tags are then displayed along with the automatically acquired data. An emphasis was placed on maximizing MOMA's flexibility and an attempt was made to anticipate and

simplify the modifications which may be desirable in the future. Programs are written in Fortran, are fully documented, and are run under the RT-11 operating system.

The potential usefulness of the system and the non-technical documentation was assessed by five anesthetists practicing at Vancouver General Hospital (VGH). It was their opinion that MOMA is a potentially useful addition to the operating room and each assessor expressed an interest in being involved in future clinical applications. The adequacy and overall quality of the programs and related technical documentation was assessed by an engineer employed in the EEG department at VGH and experienced in both Fortran and RT-11. It was his opinion that the programs and related documentation are adequate and will be useful for the development of a clinically useful patient monitoring system.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

ACKNOWLEDGEMENT

I would like to express my appreciation to Dr. Grant B. Anderson for his invaluable supervision and encouragement throughout this research and to Drs. Jim A. McEwen and Peter D. Lawrence for their many helpful suggestions and assistance.

I am indebted to Drs. W. Dodds, J. Nixon, K. Turnbull, B. Walmsley and D. Wong of the Department of Anesthesia and Mr. B. Reeves of the Department of Electroencephalography at Vancouver General Hospital for their participation in the assessment of the system produced for this thesis. In addition, grateful acknowledgement is made to Dr. Jenkins of the Department of Anesthesia and Dr. M. Low of the Department of Electroencephalography for their cooperation and support.

I would like to express my appreciation to the following people at U.B.C. for their contributions: Ms. Gail Hrehorka of the Department of Electrical Engineering for her efficient typing of this thesis, Ms. Cristiana Irving of the Biomedical Communications Department for the preparation of numerous diagrams, and to my fellow graduate students for providing an enjoyable and stimulating work environment.

Finally, the financial support received from the Natural Sciences and Engineering Research Council in the form of a postgraduate scholarship is gratefully acknowledged.

1. INTRODUCTION

1.1 The task of the Anesthetist

There are two main purposes of anesthesia: to help minimize the impact of surgery on the patient both during and after an operation, and to maintain the patient in a state which facilitates the surgeon's job. The anesthetist's task is to weigh the relative merits of the above objectives, which may be contrary, and to choose an optimal course of action.

The anesthetist prepares for this job long before entry to the operating room. In addition to a knowledge of the effects of the anesthetic agents used, the anesthetist must have a knowledge of the effects and requirements of the various surgical procedures. Using this knowledge, he can gather relevant case-specific information and plan an optimal approach to anesthetizing the patient. Continuous monitoring of relevant parameters within the operating room allows the anesthetist to assess the effectiveness of the anesthetic procedure and if necessary make appropriate adjustments.

1.2 Complicating factors in Anesthesia

Modern surgical procedures tend to complicate the anesthetist's task. The increased impact some of these procedures have on the patient leaves little room for error. Also, by distorting parameters conventionally used within the operating room, some surgical procedures can interfere with the anesthetist's ability to assess the patient's state. For example, the significance of parameters related to the cardiopulmonary system is drastically reduced during the critical bypass phase of cardiopulmonary bypass procedures.

The current trend in anesthetic practice is towards the graded administration of agent (1). By using a number of low dose anesthetic agents, the anesthetist can tailor the overall effect of anesthesia to the particular patient and surgical procedure. This will also permit the use of a lighter level of general anesthesia, resulting in a reduced anesthetic impact on the patient. However, this graded approach to anesthesia increases the requirement for patient assessment. To safely maintain the patient at light anesthetic levels requires accurate information about the patient's state. Additionally, since the agent is tailored to particular physiologic system, system specific information is necessary. Also, since it is essential that the anesthetist be aware of the effects the drugs have on one another, this approach places a larger burden on the knowledge and experience of the anesthetist. Sources of patient-related information are presented in figure 1.

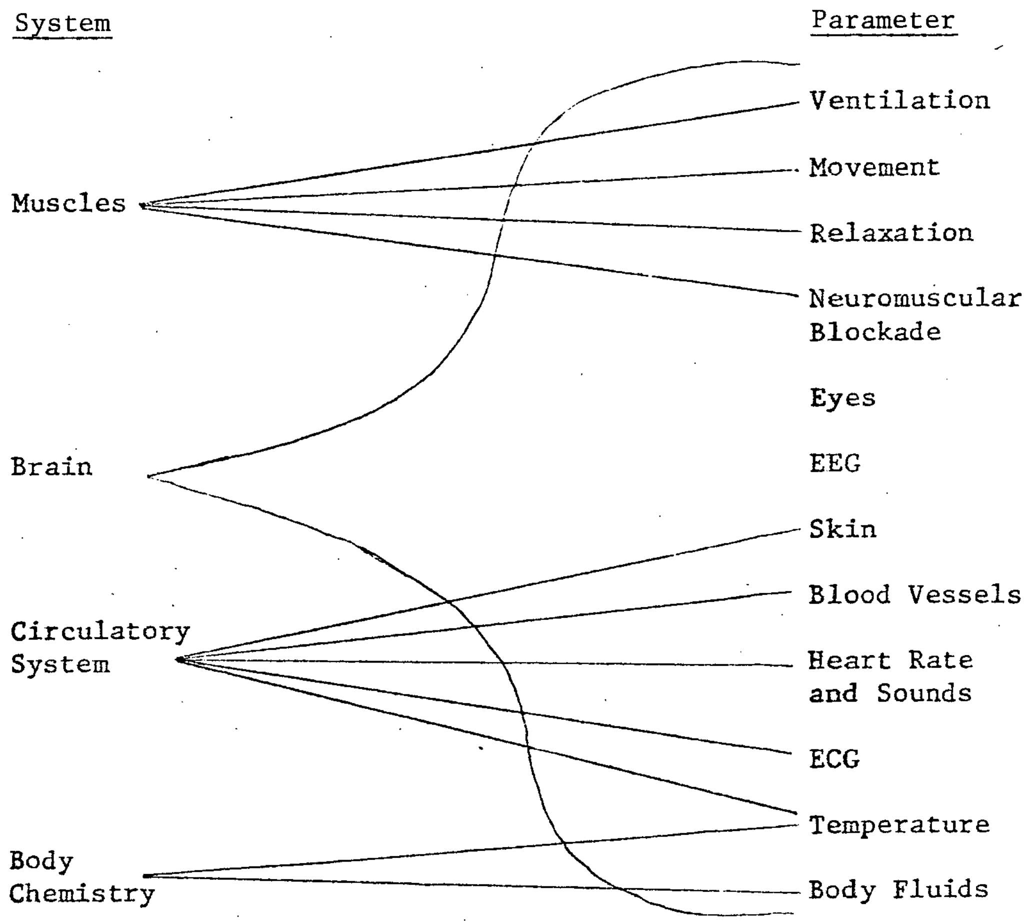


Figure 1: Patient-related monitoring parameters.

Interpatient variability further complicates the anesthetist's job. In addition to the usual physical differences between patients, apparently unrelated pathologies and premedications can cause dramatic changes in the effects of anesthetic agents. A partial solution to these difficulties is to gather relevant information before the operation by interviewing the patient and examining hospital records. However, continuous patient surveillance is necessary in case the collected information is inaccurate or incomplete.

In general, the parameters used for monitoring in the operating room are highly interrelated. Thus, a change in a given parameter may be the result of changes in many systems, both mechanical and physiological. The net result of this is that the anesthetist must monitor many things and consider these interrelationships in all the decisions he makes.

1.3 Scope of the Thesis

The dramatic advancement of computer technology during the past decade has made computer-based perioperative patient monitoring a realistic proposition. The purpose of this thesis is:

1) to discuss the potential application areas of small computer systems to intraoperative monitoring by anesthetists.
2) to define a set of design considerations which should contribute to the acceptance of computer-based systems in the operating room.
3) to produce a prototyping system for the development of an intraoperative monitor.
4) to use the above prototyping system in the design of a prototype monitor.

Chapter 2 is a discussion of the potential applications of computers in the operating room. The present day approach to anesthetic practice within the operating theatre is described. Some of the features of computer systems which may simplify the anesthetist's task are presented. Considerations for the design of a useful device are then discussed. Finally, since a major component of this project is the EEG, the problems of intraoperative EEG use are briefly described.

Chapter 3 is an overview of some of the previous attempts at using computers for patient monitoring. A few typical papers are cited which exemplify the techniques and approaches used in the past.

Chapter 4 describes the development of the system, which includes the hardware and software decisions and the general design philosophy. Specifics of the data acquisition and processing techniques are also described. Finally, some of the system's external features and a description of the display formats are presented.

Chapter 5 presents the results of an assessment of the device by hospital personnel. The potential usefulness of the overall system was judged by a group of anesthetists and the software was judged by a programmer familiar with similar systems.

Chapter 6 presents some conclusions and some suggestions for future research. Application areas both inside and outside the operating room are discussed.

2. COMPUTERS IN THE OPERATING ROOM

2.1 Intraoperative Monitoring at Present

Figure 2 portrays the relationship between the anesthetist and the patient within the operating room. The anesthetist can control the patient indirectly by adjusting the anesthesia machine or directly through the administration of injected drugs and fluids. To gauge the effect of this control, he has at his disposal a number of monitoring parameters. Interpretation of these parameters is affected by the status of the patient, the equipment used, and the operating room environment throughout the operation. In addition, other factors such as the characteristics of the surgical procedure, peculiarities of the patient, and previously administered medications must be considered. As stated in chapter 1, some of these other factors can and should be documented prior to the operation in the form of paper records. Additional paper records are manually maintained by the anesthetist during the operation in order to record important monitoring parameters and surgical events.

A retrospective study of preventable anesthetic mishaps involving human error and/or equipment failure found that 82 percent of these incidents involved human error (2). A conclusion one can draw from this is that significant safety improvements may result from a general simplification of the anesthetist's task. More specifically, simplifying the use of monitoring parameters and equipment should result in an improvement in the quality of anesthetic practice.

As described in Boba (3), the interpretation part of the anesthetist's task is basically a prediction of the likely sequence of events in the immediate future, given that existing conditions remain unaltered. This prediction process relies as much on the trends in the parameters as it does on the

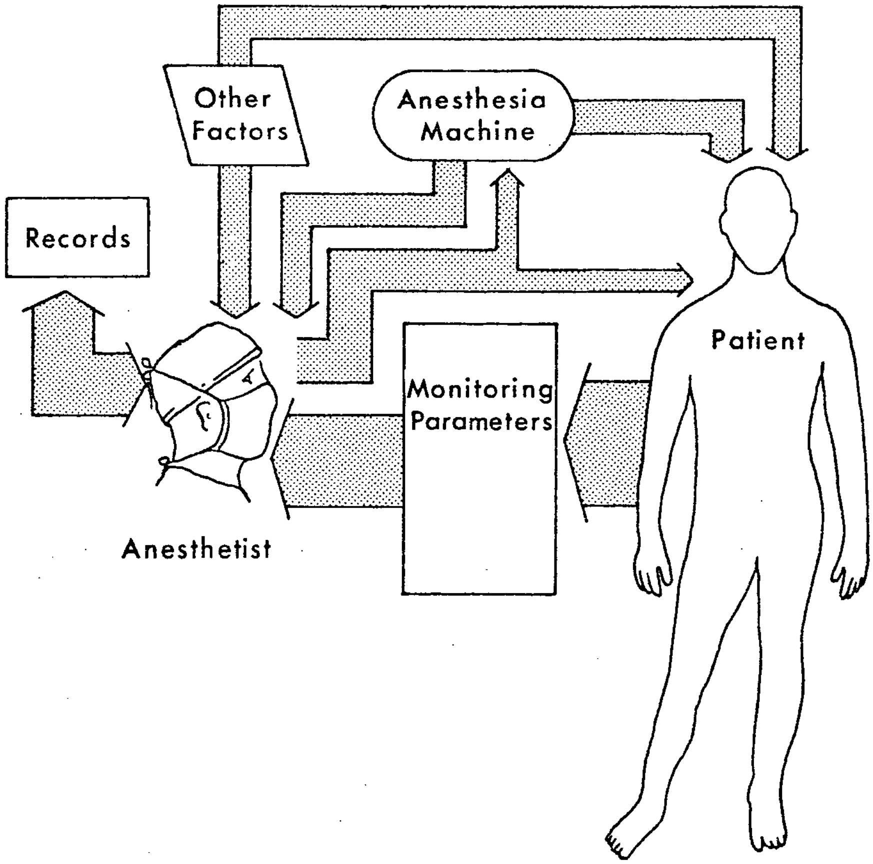


Figure 2: The relationship between the anesthetist and the surgical patient within the operating room.

instantaneous level of those parameters. For example, a low and dropping pulse rate will be more alarming than a low and rising pulse rate. The importance of trending must be considered when display formats are designed.

The need for continuous monitoring within the operating room is generally undisputed. The decision about what and how much to monitor, however, is not so clear. One viewpoint (eg. 4,5) is that conventional parameters which are normally monitored during anesthesia are adequate to control mortality and morbidity and assure an adequate quality of care in the vast majority of cases. In addition, since the introduction of new devices and parameters to the operating room diverts attention from conventional parameters, such changes may increase mortality and morbidity. On the other hand, it has been argued that the increasing complexity of anesthetic practice demands the incorporation of new and sophisticated monitoring techniques and that these new techniques can speed the detection of dangerous situations (eg. 3,6). This argument brings forth a most important point. The simple addition of information to the operating room will not necessarily improve the quality of surgical health care. Significant improvements will only be made if the ease and speed of detection of a crisis situation is improved. This improvement will only be obtained if there is a decrease in the complexity of information interpretation. Thus, the effective presentation of information within the operating room is a prerequisite to high quality anesthetic practice.

Osborn (7) discussed the difficulties in assimilating data from respiratory measurements in intensive care. In this paper, he identified two basic problems in interpreting large amounts of data. Firstly, the accuracy of diagnosis decreases when a large amount of relevant data is presented. Secondly, a major problem in handling data is that of identifying the clinically significant parameters for a given situation. These problems also apply to patient monitoring in the operating room and emphasize the importance of effective information presentation. This viewpoint has also been expressed by others (eg. 8).

2.2 The Potential of Computers in the Operating Room

2.2.1 Data Acquisition and Display

Many patient monitors currently used in the operating theatre acquire and display a single parameter. As a result, the anesthetist must scan a large number of displays to gather information. Also, the angle which he must scan to gather this information can be large. Through the use of a computer system with appropriate data acquisition and display hardware, it is possible to acquire a number of parameters and display them in a compact form. Thus, one can reduce the scanning angle required to gather information and, perhaps, lower the reaction time to significant changes. Additionally, a multiparameter format will permit the generation of displays which simplify the task of parameter comparison.

The general emphasis of the information presented to the anesthetist will have a bearing on its ease of interpretation. For example, it would not be desirable to place an important parameter on a small display. It was suggested earlier in this chapter that the ease of information interpretation directly affects patient safety. Therefore, since a cathode ray tube (CRT) can display information in many diverse formats, a multiparameter monitor with a CRT may be able to increase patient safety by permitting the design of displays with dramatic parameter emphasis.

The relative importance of parameters in the operating room will not necessarily remain the same throughout an operation. Also, as exemplified by the differing opinions about the necessity of electronic patient surveillance, this allocation of importance can vary considerably from anesthetist to anesthetist. If there are a number of display formats accessible through a single display medium, the individual anesthetist can choose the one which he feels best suits the present situation. The resulting increase in the efficiency of information use should produce an increase in patient safety. It is possible to provide this multiple display capability using a computer-based system with a CRT.

2.2.2 Signal Processing

The signal processing capability of computers may prove useful in reducing the parameter interpretation load. Some of the comparisons and correlations which are presently performed subjectively by the physician could be performed by the computer prior to display (eg. 7). The resulting reduction in the parameter interpretation load could perhaps make various critical situations more apparent, thus reducing reaction time and increasing patient safety.

Artefact detection and rejection techniques are not currently well developed. New techniques using computers may overcome some of the artefact problems in noninvasive monitoring, eg. EEG, thus making such monitoring practical for use in a wider variety of situations. In addition, the quality of signals obtained using invasive techniques may also be improved using these techniques. By permitting the use of less invasive monitoring methods and improving the quality of signals within the operating room, there should be a general increase in patient safety.

Many signals obtainable within the operating room have been shown to have clinically significant characteristics. The use of some of these signals has been limited by the time required for the anesthetist to interpret their characteristics. By preprocessing these signals it may be possible to reduce the difficulties preventing their use. For example, the use of signal processing techniques on the EEG can reduce the amount of information presented to the anesthetist and should make it easier for the anesthetist to quickly discern clinically significant EEG characteristics, thus making it more practical as an intraoperative monitoring parameter.

A computer system can directly decrease patient risk by providing warnings about potentially hazardous conditions. By alerting the anesthetist to these conditions, the computer may reduce problem detection time, and could even initiate appropriate compensatory action. Although some instrumentation currently used in the operating room incorporates warning systems, a computer will permit far more elaborate checking procedures by comparing several parameters and parameter trends at once. The resulting system could thus detect a wider variety of conditions and provide earlier warnings.

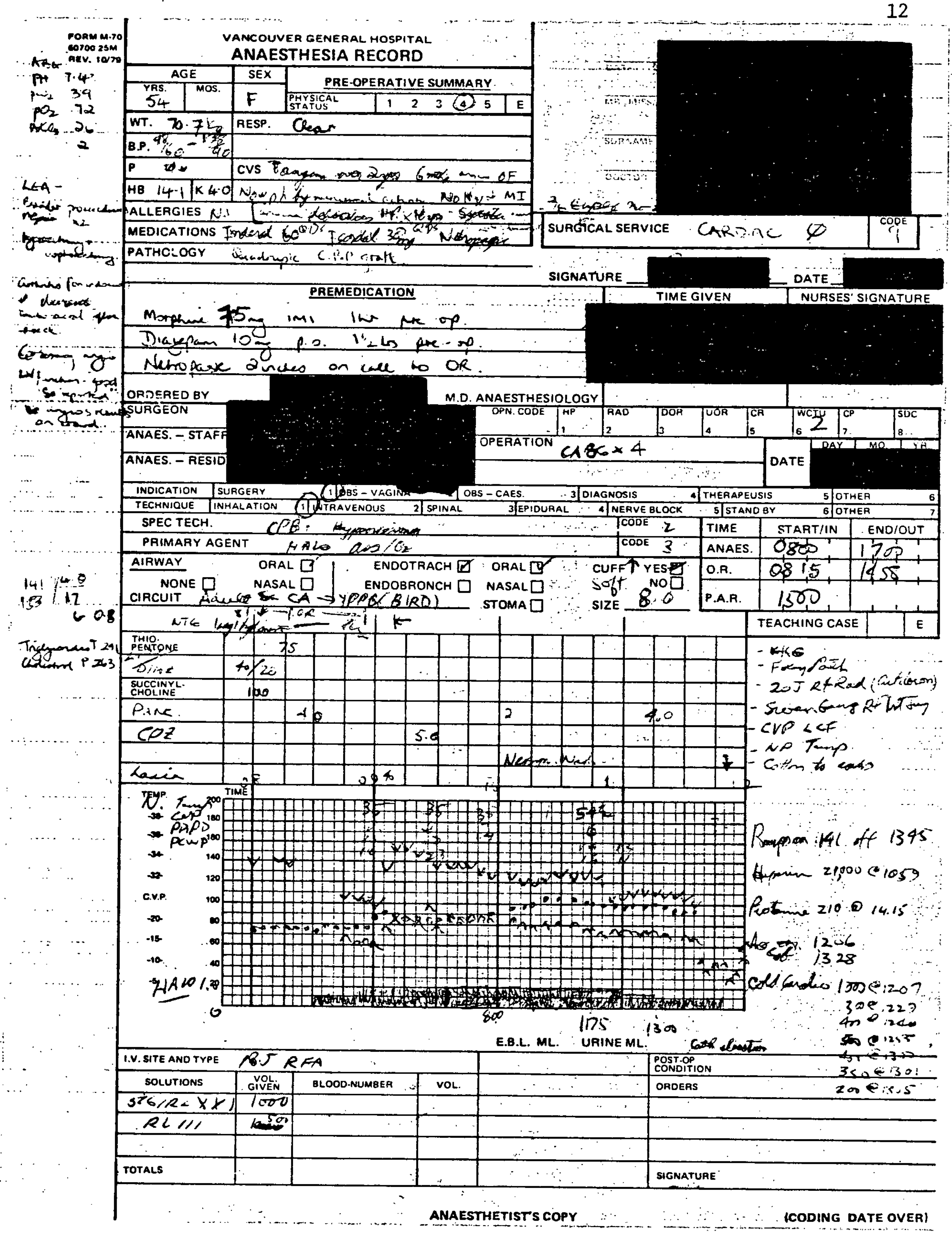
It might be thought that a logical extension of a warning system is automatic patient control. Some such systems have been developed (9-12). However, there are a number of reasons that this, at best, is extremely difficult. Firstly, a number of parameters routinely used in the operating room are not easily acquired by a computer. For example, obtaining the electrical analogue to patient pallor or restlessness is not a simple matter. Secondly, all the parameters to be used by a computer during the operation must be defined and setup prior to the operation. For these two reasons, long and complicated setup procedures would be required to configure a computer system for the task of patient control. Assuming that the computer has been configured for controlling the patient, the program required to interpret the incoming data will be algorithmically and computationally complex. It must be able to account for any situation which may occur in the operating room, to detect any source of false signals, and to "know" as much about the operation, the anesthetic agent, and the patient as does the anesthetist. The anesthetist, on the other hand, has the ability to add and delete parameters during the operation as their significance varies, thus adapting to varying conditions in the operating room. In addition, the anesthetist can draw upon years of experience and training when interpreting data and choosing appropriate courses of action. Therefore, computers are best applied to the task of aiding, rather than supplanting, the anesthetist.

2.2.3 Information Storage

One of the problems of information presentation within the operating room is that there is far more relevant information available than can be usefully displayed at one time. In the past the only solution to this problem has been to work with a fixed subset of this information. A computer-based monitor with a magnetic information storage medium may partially circumvent this limitation. Using such a system, relevant information can be "remembered" for future recall. With an appropriate design, the anesthetist could access various subsets of the stored information through a CRT at any time during the operation.

Storing and accessing information in the manner described above may significantly enhance the anesthetist's ability to monitor trends and maintain records. At present, the only routinely applied means of recording and displaying trend information is through the use of hand maintained paper records. The legibility and the format of these records can vary from author to author. Also, the maintenance of these records takes attention away from the patient and the completeness of the records is limited by the amount of time the anesthetist can allocate without compromising the patient's safety. By automatically sampling and storing measured parameters the anesthetist can be relieved of some of his record maintenance duties while still having access to trend information. The resulting record will be more complete and of consistent legibility and format. An example of a manually maintained intraoperative patient record can be found in figure 3.

The potential benefits of automated record keeping extend beyond the operating room. The consistent legibility and format will make anesthetic records more useful for training purposes. For the same reasons, it will be simpler to discover potential procedural improvements through the analysis of hospital records. Finally, the objective and consistent nature of these records will make the task of event reconstruction far simpler in cases of surgical mishap.


FORM M-70
60700 25M
REV. 10/79

VANCOUVER GENERAL HOSPITAL
ANAESTHESIA RECORD

[redacted]

AGE		SEX	PRE-OPERATIVE SUMMARY
YRS. 54	MOS.	F	PHYSICAL STATUS 1 2 3 4 5 E

WT. 70.7 kg
RESP. Clear
B.P.
P
CVS
HB 14.1 K 4.0
ALLERGIES Nil
MEDICATIONS
PATHOLOGY Quadruple C.P.P. graft

SURGICAL SERVICE CARDIAC
CODE

SIGNATURE [redacted] DATE [redacted]

PREMEDICATION

Morphine 7.5 mg IM 1 hr pre op.
Diazepam 10 mg p.o. 1½ hrs pre-op.
Nitropaste 2 inches on call to OR.

TIME GIVEN
NURSES' SIGNATURE
[redacted]

ORDERED BY [redacted] M.D. ANAESTHESIOLOGY

SURGEON [redacted]
ANAES. – STAFF [redacted]
ANAES. – RESID [redacted]

OPN. CODE | HP 1 | RAD 2 | DOR 3 | UOR 4 | CR 5 | WCTU 6 | CP 7 | SDC 8

OPERATION CABG × 4

DATE DAY MO. YR. [redacted]

INDICATION | SURGERY 1 | OBS – VAGINAL 2 | OBS – CAES. 3 | DIAGNOSIS 4 | THERAPEUTIC 5 | OTHER 6
TECHNIQUE | INHALATION 1 | INTRAVENOUS 2 | SPINAL 3 | EPIDURAL 4 | NERVE BLOCK 5 | STAND BY 6 | OTHER 7

SPEC TECH. CPB: Hypothermia CODE 2
PRIMARY AGENT Halo N₂O/O₂ CODE 3

TIME	START/IN	END/OUT
ANAES.	0800	1700
O.R.	0815	1655
P.A.R.	1500	

AIRWAY ORAL NASAL NONE ENDOTRACH ENDOBRONCH ORAL NASAL STOMA CUFF YES NO
CIRCUIT SIZE 8.0

TEACHING CASE E

THIO-PENTONE 75
SUCCINYL-CHOLINE 100
CO2 5.0
Lasix

- EKG
- Foley Cath
- 20 J Rt Rad
- Swan Ganz Rt Int Jug
- CVP LCF
- NP Temp
- Cotton to ears

TEMP. TIME
200 180 160 140 120 100 80 60 40 20
C.V.P.

Pump on 1141 off 1345
Heparin 21000 @ 1053
Protamine 210 @ 14.15
Ao cx 1206
off 1328
Cold Cardio 1000 @ 1207

E.B.L. ML. URINE ML.

I.V. SITE AND TYPE
POST-OP CONDITION
ORDERS

SOLUTIONS	VOL. GIVEN	BLOOD-NUMBER	VOL.
	1000		
RL			
TOTALS			

SIGNATURE

ANAESTHETIST'S COPY (CODING DATE OVER)

Figure 3: A typical anesthetic record. This form is only part of the record maintained by the anesthetist during open heart surgery.

At present, the only way to supplement the anesthetist's memory of important presurgical information is through the use of paper records. However, the number of these records which can practically be included in the operating room is limited by operating room clutter considerations. A computer-based preoperative information facility would give the anesthetist access to large amounts of information without increasing operating room clutter. Indexing and paging schemes will allow that anesthetist to access various subsets of this information quickly and easily. Although such a computer-based system will not be more convenient than using one or two well structured paper records, it will permit access to a large data base whenever more detailed information is desired.

The topic of computer-based record keeping has been dealt with in the literature and some record keeping systems have been developed (12-20).

2.3 EEG in the Operating Room

The prospect of using the EEG for the determination of anesthetic level was first proposed in 1937 (21). Since that time, consistent correlations between depth of anesthesia and EEG patterns have been shown to exist (22-24). Additionally, certain physiologic abnormalities which can occur in the operating room (eg. hypoxia) consistently elicit clinically useful changes in the EEG (25-27). Since the target organ of many of the drugs used in the operating room is the brain, one would think that a direct measure of the gross electrical activity of the brain such as the EEG would be a useful monitoring parameter. The fact that the EEG is obtained non-invasively makes it clinically appealing. However, it has not seen routine use in any but a few specialized operations.

McEwen (23) suggested that the failure of the EEG to measure up to its promise is largely due to problems contributing to the lack of validity and reliability of related studies reported in literature. In addition, from the anesthetist's standpoint, the trends indicative of gradual changes in the patient's state are not quickly and easily observable using standard EEG recordings. Thus, to make the EEG a useful monitoring parameter, what is needed is a standardized set of significant features and some means of making those features readily apparent.

Recent advances in computer technology and signal processing have renewed interest in intraoperative EEG use. McEwen's thesis went a long way towards defining a useful set of significant EEG features. Additionally, by computing and outputting the EEG power spectra at specific time intervals as suggested by Bickford (58), the difficulties in observing significant trends may be partially overcome. A discussion of the way the EEG was used in the monitor developed for this thesis can be found in chapter 4.

2.4 Intraoperative Monitoring in the Future

Figure 4 portrays the revised relationship between the patient and the anesthetist which can be expected when computers are introduced to the operating room. The general flow of events and interpretation considerations are the same as described at the start of this chapter. However, through the addition of computers, the anesthetist can access more information without a concommitant increase in the difficulty of data interpretation. Detailed preoperative information about a variety of factors can be stored and accessed only when needed. Parameters presently not used can be acquired and displayed along with some of the currently used parameters in a format designed to simplify interpretation. The necessary surveillance of critical equipment such as the anesthesia machine can be in part taken up by computers, thus freeing the doctor for other tasks. Finally, some of the record keeping tasks of the anesthetist can be taken over by computers and, in the process, the anesthetist can be provided with a more complete record of the operation.

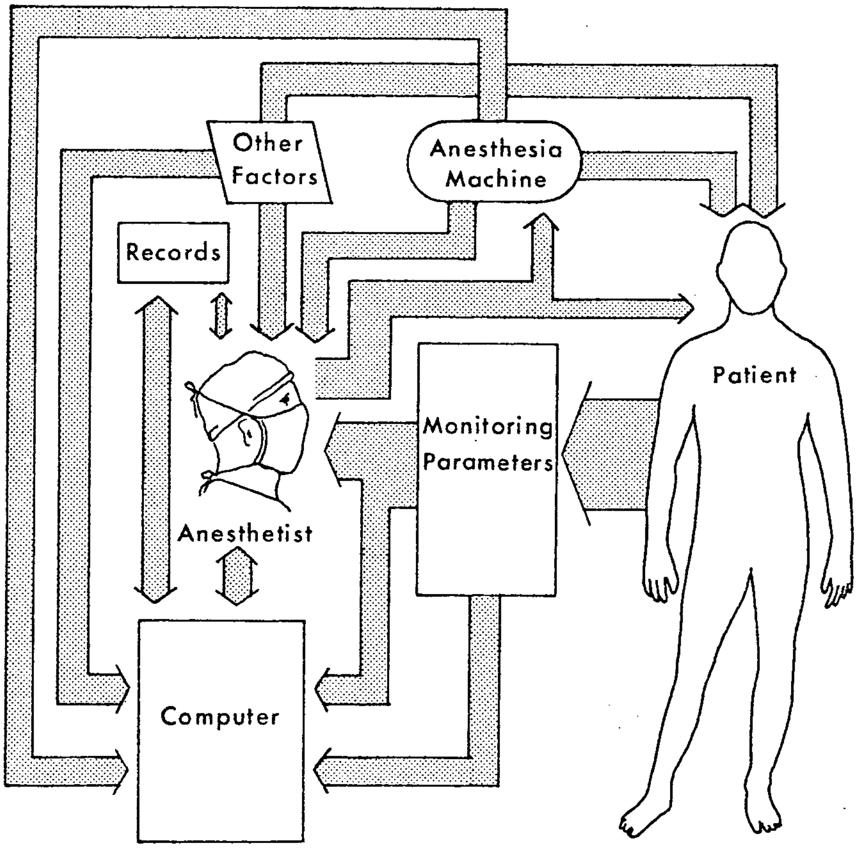


Figure 4: The relationship between the anesthetist and the surgical patient when a computer is introduced to the operating room.

2.5 Design Considerations

Many of the signals obtainable within the operating room may be contaminated with noise from a number of sources, resulting in what are referred to as artefacts, Thus, in the design of surgical patient monitors, an attempt has to be made to identify and, if possible, compensate for artefact sources. Once these sources have been compensated for, the signals acquired by the system will be more useful and the effectiveness of a warning system based on those signals will not be as easily compromised by false alarms. Examples of artefact sources are power lines, electrosurgical devices, defibrilators, patient movement, and interference from physiological systems other than the one being monitored.

As the monitoring power of a device increases so does the detrimental effect of a breakdown of that device. It is clear, then, that if computers are to play an important role in the operating room, they must be extremely reliable. To achieve this reliability, one can incorporate battery backup systems and duplicate certain critical hardware components. However, these measures increase the size and cost of the monitor. In order to minimize the effect of a breakdown, there should be some means of restarting the system during the operation once the reasons for that breakdown has been rectified.

There are a number of documented incidents where faults and design defects in monitoring equipment have directly contributed to injury to the patient (28,29). In order to minimize the number of these incidents, safety and performance standards for electromedical equipment have been set (30) and techniques for the design of safe equipment have been developed (31,32). Every effort must be made to limit the possible danger to both the patient and the user due to device failure or misuse.

Within the operating room, the prime concern of the anesthetist should be patient maintenance. Consequently, the operation of a monitoring device should require little or none of the anesthetist's time. Also, the anesthetist cannot be expected to be either a computer programmer or an expert typist. For these reasons, a computer-based monitor must essentially run itself. User interaction should be optional and machine-initiated wherever possible. Requests for interaction should not require immediate attention. Finally, interaction should be simple, easily understood and easily remembered. It is clear that the likelihood of general acceptance of any monitoring device will also decrease as setup time increases. As a result, the setup time must be minimized.

For the following reasons, it is not desirable to have components of a monitoring system outside the operating room. Firstly, it would then be necessary to "wire" every operating room which may use the system, thus reducing the system's portability. Secondly, the malfunction of any external device(s) may be difficult to detect and quickly rectify. Thirdly, system startup and operation procedures are likely to be complicated, requiring travel in and out of the operating room and/or more than one person. Finally, the likelihood of accidental interference with the proper functioning of the system by external sources is greater. For these reasons, the components of an intraoperative monitoring system should all be in the operating theatre.

A number of additional practical considerations must be taken into account if a computer-based monitor is to be inserted into the operating room. Firstly, in order not to irritate members of the surgical team and interfere with communication, it must be reasonably quiet. Secondly, it must be small enough to fit into the operating theatre without impeding the movement of people. Finally, in order to allow reorganization of the operating theatre to suit the needs of specific procedure, it should be moveable.

In addition to monitoring parameters related to the patient, the anesthetist must insure that proper functioning of various patient maintenance devices such as the anesthesia machine. Through the redesign of these devices (33-36), a concommitant easing of the anesthetist's task may be expected in the same way as has been described in connection with the computer-based monitoring of physiological parameters. Although developments in computer-based anesthesia machines, intraoperative patient monitors, and surgical support systems have proceeded separately, there is no reason that the information output from these different systems could not be displayed on the same display medium (eg. a CRT). The optimum organization of all the various types of relevant information will only be determined through clinical experience.

As with any medical device, the potential benefits of a computer-based monitoring system must be weighed against expense. Although the price of computer systems with sufficient computational power has been and should continue to be on the decline, the monitor must significantly increase the ease and accuracy of anesthetic practice to be justified. The cost effectiveness of computer-based systems can only be assessed through extensive clinical evaluation of prototype systems. Until the results of such evaluations are available, this issue will remain in doubt.

In the preceding discussion, potential application areas of computer-based perioperative monitoring systems were given. In order to develop such a system, it can be expected that a program of design and re-design with interspersed clinical assessment will be required (see figure 5). In this way, it is possible for the skills and opinions of experts from different fields to be more fully reflected in the design. It should be clear, also, that from the beginning an evolutionary structure should be sought so as to allow for easy modification and extension of the system as its purposes and function are better understood.

The potential applications of computer-based data acquisition and monitoring systems extend far beyond the operating room. The ability to sample, process, store, and display electrical signals is useful in a wide variety of other areas, including many clinical departments and intensive care units. By redisplaying previously acquired data and/or mimicking

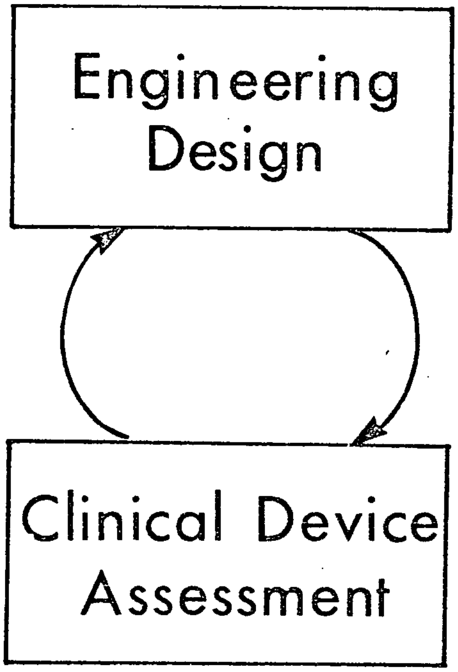


Figure 5: The medical device development cycle.

procedures, such systems can be used as training tools. Computers in general have been found to be useful for the storage of large volumes of information. This feature, along with proper data entry tools can be used to assist in the generation of letters, technical papers, and hospital documents. These potential uses must be considered when the overall cost effectiveness of any computer-based system in the hospital is assessed.

3. PREVIOUS PERIOPERATIVE PATIENT MONITORING SYSTEMS

3.1 EEG Only Systems

It was stated in chapter 2 that a major reason the EEG has seen limited use in the operating room is the difficulty in quickly obtaining useful information from conventional EEG recordings. Consequently, if the EEG is to be effectively used by the anesthetist within the operating room, some means of presenting the clinically significant EEG characteristics in a quickly and easily interpreted format is necessary. A number of devices have been designed for this purpose.

A device called the cerebral function monitor (CFM) (37) has seen use in operating rooms. This device band-pass filtered (2-5 hz), amplified, rectified and logarithmically compressed incoming EEG signals from two scalp electrodes. The impedance of the electrodes was also monitored, thus allowing artefacts due to diathermy or faulty electrodes to be detected. The CFM has been found useful for the detection of abnormal neurological conditions during surgical operations (38-40).

Volgyesi (41) presented the hypothesis that the depth of anesthesia can be monitored by observing the relative amplitude of delta waves and the shift in frequency of the dominant rhythm of the EEG. Using this hypothesis, a device was designed which determined the ratio between the mean amplitude of the "augmented" delta frequencies and the mean amplitude of the entire EEG signal, called the augmented delta quotient (ADQ).

Fleming and Smith (42) described another EEG monitoring system. This system computed the frequency spectrum of the EEG signal from 0 to 16 Hz and periodically outputted it using a DSA format (see section 4.5.1). This format provided the user with both amplitude and frequency information while at the same time considerably reducing the amount of paper required to present the EEG throughout the operation.

3.1.1 Discussion

All of the above systems suffered in that they serve only to add information to the operating room. Little was done to improve the overall presentation of information within the operating room or to make relationships with other parameters readily apparent.

In general, EEG signals are described using their time and/or frequency characteristics. However, the display formats of the CFM and the ADQ monitors do not independently present either of these characteristics. The inability to analyse these displays using commonly encountered feature descriptors makes it more difficult to use past experience to detect and reject artefacts. Consequently, while these devices do little to make parameter interrelationships readily apparent, the nature of their displays increases the need for deriving these relationships. Also, each of these systems acquires and processes only one EEG channel. Thus if information from a number of EEG channels is wanted, a number of devices must be used. The extra time required to monitor and operate these devices combined with difficulties in directly comparing their outputs make this arrangement infeasible.

It was suggested by Frost et al (43) that one of the major limiting factors preventing the development of EEG automation is the configuration of commonly available computing devices, i.e., the centralization of the control of data acquisition, processing, and display duties in a computer-based monitoring system limits that system's effectiveness. A multiprocessor approach to EEG acquisition and interpretation was then described. For the same reasons as Frost presented for automatic EEG interpretation, this approach may also prove useful in the design of perioperative patient monitors. Using such an approach, systems similar to those described at the start of this chapter may prove to be useful as preprocessing modules for a multifunction monitoring system.

3.2 Multiparameter Monitors

In 1972, J.S. Brodkey et al (44) described a computer-based patient monitoring system for neurosurgical patients. This system was capable of sampling, storing, and displaying EEG, evoked potentials, ICP, and blood pressure. The data display medium was a laboratory oscilloscope.

A World Health Organization (WHO) travelling fellowship report in 1978 overviewed work in computer-assisted patient monitoring at a number of centers across North America (45). Four of these centers were involved with the application of computers in the operating room. The following five paragraphs describe work at these centers.

A system developed in Cleveland can simultaneously monitor heart-rate, temperature, systolic and diastolic blood pressure, inspired oxygen concentration, and blood oxygen saturation from 4 patients (45 pp. 6-9). The computer was located outside the operating room and data was displayed on a video monitor within the operating room in the form of trend graphs. Information about various surgical events could be entered via a small, specially designed keyboard. It was planned to expand this system to include the acquisition and processing of four channels of EEG.

A system for the continuous monitoring of brain activity during anesthesia and surgery was developed in San Diego (45 pp. 30-33). This system monitored EEG, arterial blood pressure, heart rate, and user entered anesthetic events. This information was displayed on a video terminal which was updated every eight seconds. The EEG was displayed using either the compressed spectral array (see section 4.5.1), or the total integrated micro-voltage value (0-16 Hz) and median power frequency (6-16 Hz). The user can access data acquired 30 minutes, one hour, or five hours previously through the video terminal. Although the computer was located outside the operating room, there were plans to implement a system with similar features on a micro-computer which could be located within the operating room.

Researchers in La Jolla produced a prototype system for research (45 pp. 34-36). This microprocessor-based system monitored arterial blood pressure and one EEG channel. EEG was displayed using the DSA (see section 4.5.1) on a thermal printer. The entire system could be located in the operating room and there were plans to expand the system to incorporate more physiological parameters.

An EEG-based system to aid the anesthetist in patient state assessment was developed in Vancouver (45 pp. 40-42, 46). This system was capable of displaying 4 channels of EEG in either the CSA or DSA format along with respiratory and haemodynamic parameters. Preoperative and perioperative information could be entered via the video terminal's keyboard. This information could be displayed along with the automatically acquired data on the video terminal within the operating room. The computer, however, was located outside the operating room. This project was the impetus behind the work described in this thesis.

A system using the EEG for patient monitoring during open heart surgery was developed in the Netherlands (45 pp. 48-49, 47). Two channels of EEG were acquired and processed. Non-EEG parameters were typed into the computer, the computer was outside the operating room, and up to two patients could be monitored at one time. This system used zero crossing histograms for EEG processing and display and updated an X-Y recorder every 60 seconds.

At the 1980 AAMI meeting, Paulsen (20) described a monitoring system for slowly varying signals. A major design criterion for this project was to reduce the amount of the anesthetist's time required to perform routine chores such as record keeping. Using the system developed, the anesthetist could enter information to the system via a hand-held 16 key keypad. In addition, up to 16 analogue signals could be automatically sampled and stored every 30 seconds. This information was output to an XY plotter on top of the anesthesia machine. The format of this record was the same as the presently maintained records so it could be continued in the event of computer failure. The computer was located outside the operating room and the system was connected to 6 operating rooms and 2 recovery room beds.

3.2.1 Discussion

All but one of the systems described above (La Jolla) used computer systems that were too large to be practically located within the operating room. It was mentioned in chapter 2 that this configuration leads to complicated startup precedures, higher probability of interference from outside sources, lack of mobility, and necessitates the use of more personnel than a self contained system. Due to size limitations and deficiencies in available hardware, the computational power of the computer used in the La Jolla system was limited. Consequently, the number and sophistication of the features which could practically be implemented on this system were also limited.

Three of the above systems (Cleveland, Netherlands, and Paulsen) used the computational power of a large external computer system to simultaneously monitor a number of patients. While this feature should reduce the cost per patient of using the system, it also magnifies the negative effect of computer breakdown. Thus, a multipatient monitor both increases the need for reliability and, due to the external locations of the computer, complicates the achievement of that reliability.

A number of the systems used a cathode ray tube (CRT) for information display. An erasable display medium such as this permits a wide variety of display formats and provides the potential to access a large amount of information without increasing operating room clutter. However, depending on the systems design, accessing information which has passed from the screen requires user interaction and can be difficult. In addition, in the event of a computer failure, access to information stored in the computer is lost.

With a paper record output medium, all the information is always displayed. In addition, paper records will not become inaccessible when the monitor malfunctions. However, paper records increase the clutter of the operating room and do not have the format flexibility of CRT's. Given the above tradeoffs, it is likely that the "optimum" perioperative patient monitor will incorporate both erasable and nonerasable display media.

In an attempt to relieve the anesthetist of some of his record keeping duties, Paulsen devised a 16 key keyboard for the entry of timed comments during the operation. A special purpose keyboard was also used in the Cleveland system. Another alternative is to provide for the entry of these comments through the terminal's keyboard, as in VGH's EEGAL system. While Paulsen's system bypassed the need for typing skills to enter comments, it also limited the anesthetist to a finite number of comments thus leading to less complete or less detailed surgical records. If automated surgical records are to be accepted within the operating room, some means of entering any comment without the need for advanced typing skills will be necessary. Perhaps a combination of written and computer maintained records with some means of linking them together would be appropriate.

In order to present time trends of the EEG, the compressed spectral array (CSA) (San Diego, Vancouver), the density modulated compressed spectral array (DSA) (La Jolla, Vancouver), and zero crossing histograms (Netherlands) have been used. These formats for presentation have been found useful for the detection of relatively rapid changes in the EEG's characteristics which are indicative of certain physiological abnormalities, such as inadequate cerebral perfusion after occlusion of the carotid artery during carotid endarterectomies. In these applications, however, EEG changes related to depth of anesthesia are treated more as a source of confusing signals rather than a tool for patient state assessment. It is clear, therefore, that improvements must be made in the presentation of long term EEG trends before the EEG will be used for routine patient monitoring. A discussion of the relative merits of the above displays can be found in section 4.4.1 and section 4.5.1.

4. IMPLEMENTATION

The general design philosophy for MOMA has been aimed towards maximizing the overall flexibility of the system so as to allow later adjustment of many of its design parameters. Although this approach compromises the initial efficiency, size, and cost effectiveness of the system, these considerations can be more appropriately handled when the "optimum" design parameters have been more precisely determined.

4.1 The Hardware

Figure 6 shows the general hardware configuration of MOMA. In keeping with the design philosophy, the hardware purchased for this project has been of a general purpose nature. In the future, hardware modification and additions tailored to specific tasks may produce significant performance improvements.

The main component of MOMA is a Digital Equipment Corporation (DEC) PDP-11V03-L microcomputer system. This system includes:

1) an LSI-11 processor (KD11-HA).
2) a double density floppy disc system with total storage capacity of 1,025,024 bytes (RX02).
3) a 12 bit, 16 channel analogue to digital converter (ADV11-A).
4) a programmable real-time clock (KWV11-A).
5) four independent serial line interfaces compatible with EIA RS-422, EIA RS-423, and EIA RS-232C, and with selectable baud rates between 150 and 38400 bits per second inclusive (DLV11-J).
6) 64 K bytes of random access memory (MSV11-DD).
7) software support and documentation for the above hardware.
8) a fully documented operating system (RT-11V03-B).
9) diagnostic programs for system maintenance.

This system provides software compatibility with other DEC systems at VGH and came with much of the hardware and software required for monitor development and device maintenance.

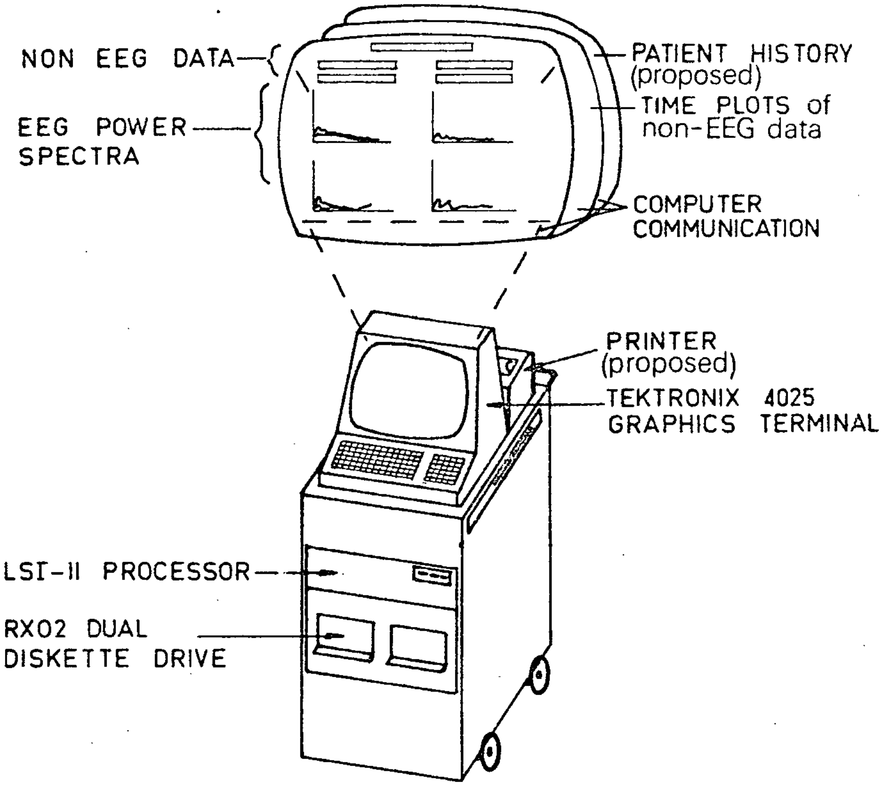


Figure 6: The General Hardware Configuration

For display and user interaction, a Tektronix 4025 graphics terminal (TX4025) was purchased. This raster-scanned microprocessor-based terminal came with 32 K bytes of graphics memory and 8 K bytes of text memory. The instruction set of the TX4025 is compatible with the Tektronix 4027 color graphics terminal. Thus, the software developed for this version of MOMA will not require major revisions if a color capability is desired in the future. The scrolling capabilities, the relatively high level graphics capabilities, and the upward compatibility with a color graphics terminal make the TX4025 well suited to this application.

A Digital Pathways rechargeable battery-supported computer clock (TCU-50D) was purchased. This device avoids the necessity of manually initializing the date and time at system startup.

4.2 The Software

For this and the remaining sections of this thesis, the following definitions will be used:

1) MOMA CYCLE = a single data acquisition, processing and storage loop.
2) MOMA CYCLE PERIOD = the time consumed by a single MOMA CYCLE.
3) SIGNAL = the analog data signal to be sampled and processed.
4) RECORD = the sampled portion of an EEG signal to be used for processing within a given MOMA CYCLE.
5) SEGMENT = a subset of a RECORD.

4.2.1 The Software Approach

The general design philosophy of maximizing flexibility was discussed at the start of this chapter. Towards this end, the following approaches were adopted:

1) A high level and frequently used language (FORTRAN) was selected for the majority of the programming tasks. Although this sacrifices execution speed and program size, it should result in a more readable and understandable software system.
2) Subprograms were used extensively so as to divide the software into logical modules and to improve readability.
3) Each module was heavily documented and contains a caption which briefly describes its intent.

4) The intent and features of each software package were documented.
5) Program flow diagrams were drawn to graphically portray the flow of program control.

4.2.2 Software Components

Unless otherwise specified, program documentation for each of the following components can be found in appendix 2.

4.2.2.1 DEC Supplied Components

All of the programs written make use of the RT-11V03-B operating system. The foreground/background monitor is required for the main MOMA programs but the single job monitor is sufficient for the rest.

A number of the sampling, processing, and I/O subroutines, which were supplied with the system, were used. The following three manuals, supplied by DEC, contain documentation for these subroutines; the Advanced Programmers Guide, the FORTRAN IV Extensions Users Guide, and the Laboratory Subroutines Manual.

4.2.2.2 Visual Editing Package

A visual editing package was written to interface the visual editing capabilities of the TX4025 with a text editor supplied by DEC (TECO). TECO was written by members of the DEC Users Society (DECUS) and is not supported by DEC. The visual editing package was written using TECO instructions and has proven invaluable as a program development and documentation tool. Comments were used sparingly in these routines because the inclusion of comments unnecessarily slows them down.

4.2.2.3 Graphics Routines

A general purpose graphics package was written to take advantage of the TX4025's graphics capabilities. This package is comprised of a group of FORTRAN subroutines for vector generation, figure generation, and data plotting. These routines were put into an object library (GPHLIB) so as to simplify the program linking procedure.

4.2.2.4 MOMA Routines

Figure 7 shows the relationship between the MOMA hardware and software components. The foreground/background operating system supplied with the computer provides a two priority level mode of operation. Routines involved with data sampling, data processing, and data storage were assigned the higher priority (foreground) and routines involved with data display and user interaction were assigned lower priority (background). Using this scheme, data acquisition and processing routines are transparent to the user and cannot be interfered with unless the program is terminated.

The absolute minimum cycle time is simply the time required for data acquisition, data processing, and data storage, which was found experimentally to be approximately 12 seconds. This, however, leaves no time for the display and interaction routines. In addition, the amount of disk storage required per unit time is inversely proportional to the MOMA cycle period. Thus, for long continuous operation of the system and efficient display and interaction, a large cycle period should be chosen. On the other hand, the longer the computer takes to update the display, the longer it takes for a crisis situation to be brought to the attention of the anesthetist. This is extremely critical in situations such as cerebral ischemia resulting from inadequate cerebral perfusion, where irreversible neurological damage can occur in a

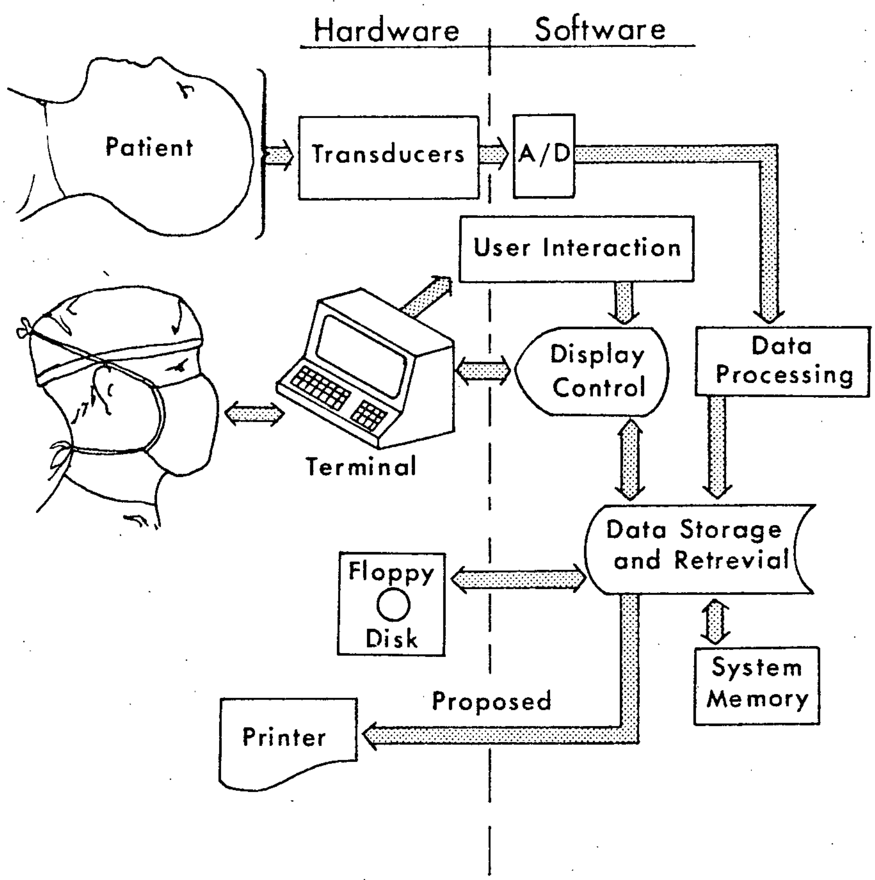


Figure 7: Information Flow Block Diagram

matter of minutes. This clearly calls for a short MOMA cycle period. The MOMA cycle period was chosen to be 30 seconds, thus leaving approximately 18 seconds per cycle for display processing and user interaction. Operations of up to 7 hours and 40 minutes may be recorded upon a single disk under these circumstances. More importantly, updating of the display is frequent enough to prevent a significant delay in the recognition of a crisis situation.

In order to simplify the future modification of the MOMA routines a large number of global parameters were used. A network of common blocks passed these parameters to various modules of the program. By appropriately setting these global parameters, it is possible to generate a wide variety of display formats and configurations.

The display and interaction routines (background routines) can run in the absence of a foreground job. This enables the user to use all the display and interaction features of MOMA when viewing data acquired from previous operations.

The RT-11 FORTRAN debug (conditional compilation) statement was used extensively in the foreground routines. These routines can be recompiled using the /ONDEBUG option and linked to run in the background. The resulting program allows the user to trace the program flow, plot the acquired data, plot the power spectral estimate, and store the data. Although this facility was designed for program testing purposes, it may also prove useful for the interactive acquisition of data. Its inability to access previously stored data and its interactive nature, however, make it impractical for use by the anesthetist within the operating room.

When the standard data acquisition, processing and storage routines are run along with the display and user interaction routines, the resulting program exceeds the available memory capacity of the system (32K words). Consequently, it was necessary to overlay parts of the routines from the disk. However, since each disk access takes, on the average, one quarter of a second, this scheme significantly slowed down the program.

4.2.2.5 Testing Routines

The PDP-11V03-L came with diagnostic programs. These programs can be used to verify the proper functioning of the various hardware components of the system.

Test programs which call various subsets of the graphics package and the MOMA package were written. This allows program bugs to be traced and program alterations to be tried. Through the use of RT-11 FORTRAN's debug feature mentioned previously, it is possible to insert statements which further aid in the task of program debugging. A number of the graphics and MOMA routines contain these statements.

Foreground and background simulation routines were written for the testing of the MOMA routines. The foreground simulator sends predefined test patterns to the background job in the same manner as does the main foreground job and is useful for the testing of the display and user interaction (background) routines. The background simulator receives and types out the message sent the foreground job and is useful for the testing of the data acquisition and processing (foreground) routines.

Finally, a routine was written to aid in the calibration of the A/D converter.

4.2.2.6 Startup Routines

A number of startup programs were written. These programs are called along with various text files by startup command files. Using these command files, the system directs it own startup, checks the integrity of various system components, and sets the modes of the terminal and the operating system.

4.2.2.7 Command Files

In addition to the startup command files mentioned in section 4.2.2.6, a number of support command files were generated. These were used to simplify the compiling and linking of packages containing many modules.

4.3 Monitoring Parameters

The A/D converter has 8 independent channels (quasidifferential mode) available for the acquisition of analog data. It was decided that the same four EEG channels that were used in the EEGAL project (F3-C3, C3-01, F4-C4, and C4-02 as defined by the international 10-20 system for electrode placement (48)) should be used in MOMA. These four channels represent the four quadrants of the head. MOMA's EEG acquisition process is shown in figure 8.

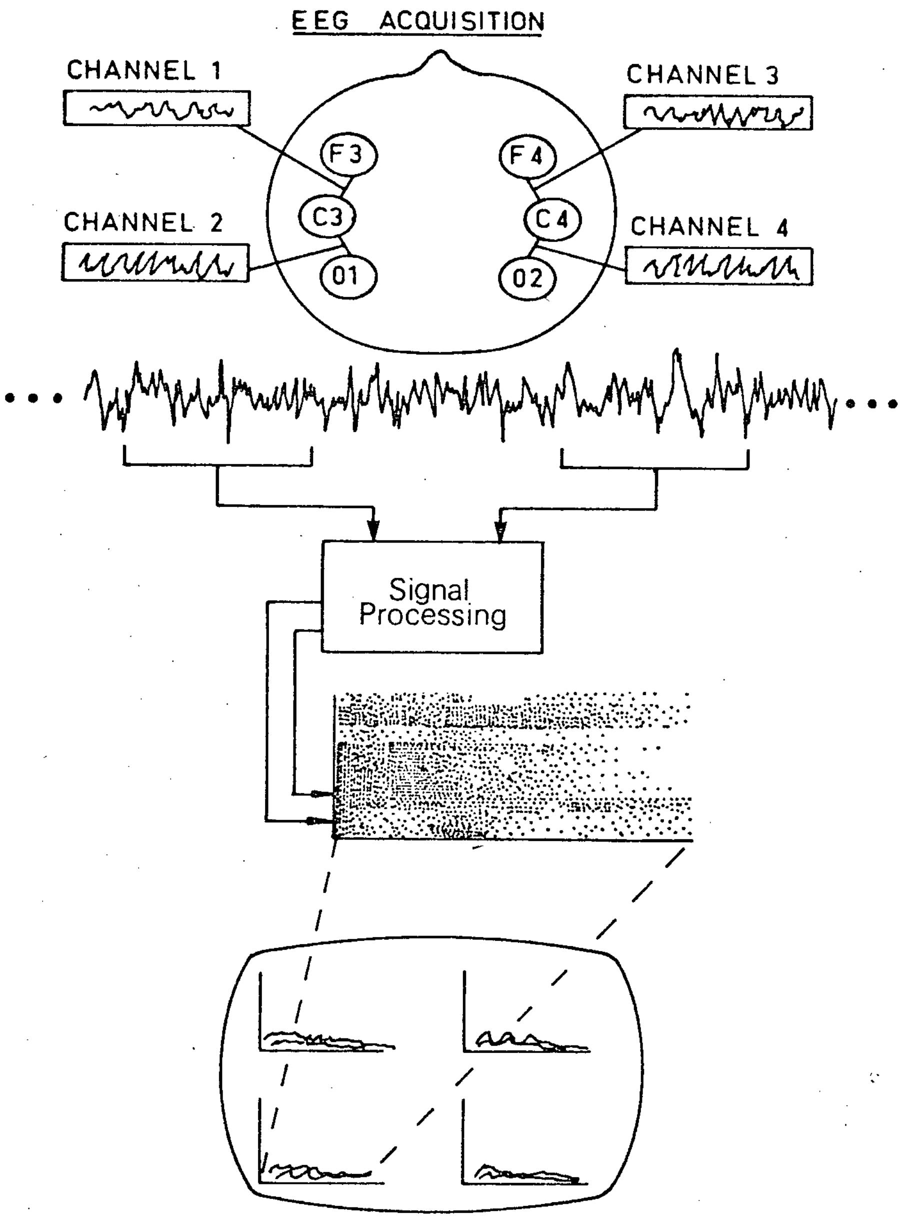


Figure 8: The EEG acquisition Process

The existing software can be set to sample each of the remaining four unallocated A/D channels once every MOMA cycle. Consequently, only slowly varying signals, such as mean blood pressure, pulse rate, or temperature, can be meaningfully sampled. It is possible, however, to insert program modules which sample these channels at a faster rate.

The only processing of data acquired through the unallocated channels is that of multiplicative and additive scaling. However, as with the sampling rate, it is not difficult to insert program modules which perform signal processing appropriate for a particular parameter.

Some of the monitoring devices presently used in the operating room provide outputs appropriate for sampling by a computer. Thus, for initial clinical trials, likely candidates for sampling will be those parameters routinely acquired and displayed using these devices. For example, pulserate, temperature, left atrial blood pressure (LAP), central venous pressure (CVP), and mean arterial pressure (MAP), are routinely monitored during open heart surgery at VGH by devices with computer compatible outputs. Although the ECG is also monitored by such a device it would be necessary to sample it more than once every MOMA cycle to obtain meaningful data, thus some software modification would be required. As dictated by the results of clinical trials and the ease of data acquisition, other parameters will likely be added to this list. In addition, each surgical procedure has different monitoring requirements and, consequently, different parameters will be sampled.

4.4 EEG Processing

4.4.1 Discussion of Techniques for EEG Processing

Figure 9 shows samples of multichannel recordings of baseline EEG activity before and during anesthesia. This figure clearly shows that

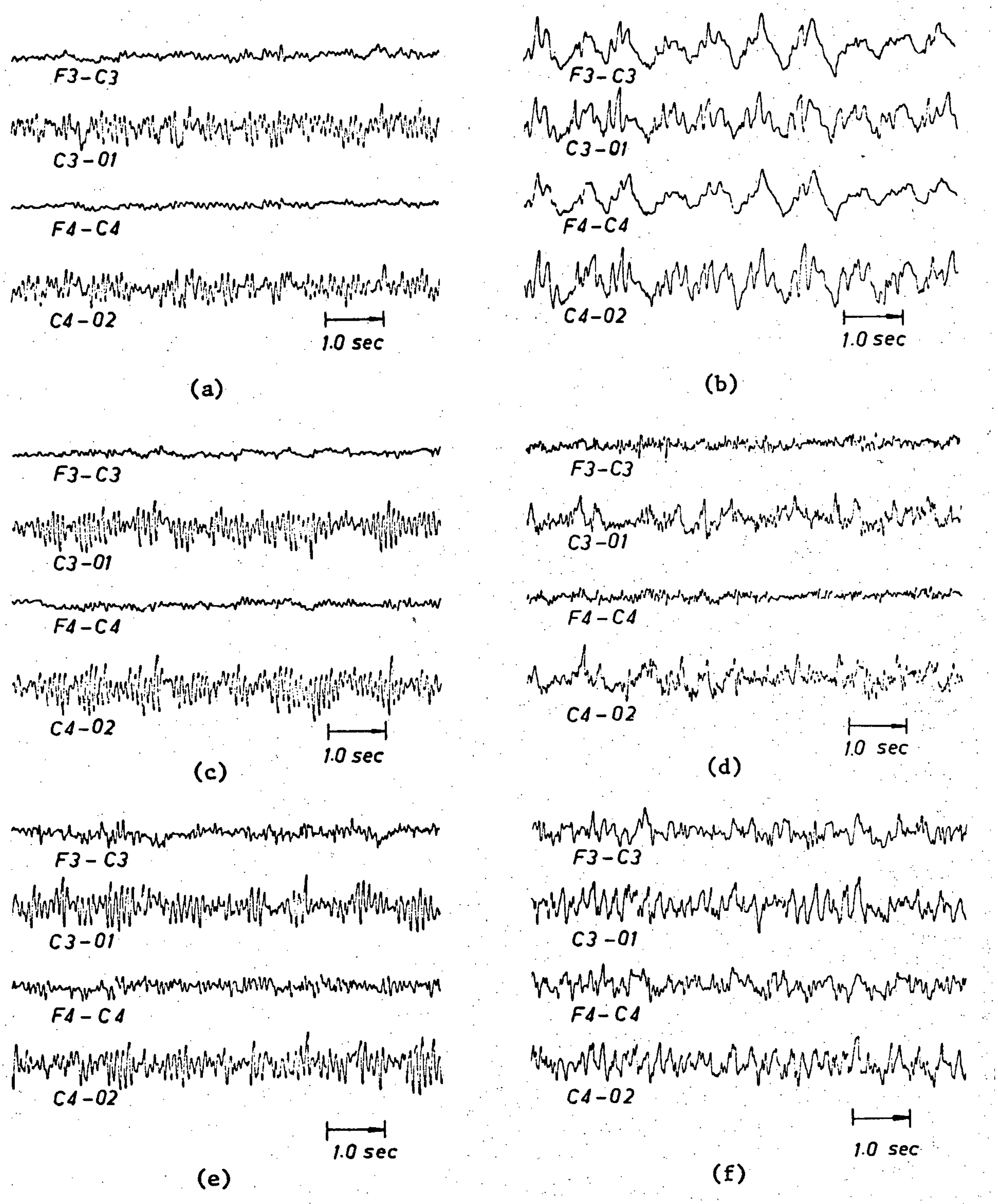


Figure 9: Sample Segments of Multichannel EEG Activity. Samples of EEG activity before and during anesthesia for three subjects having similar baseline EEG characteristics are shown in (a)-(b), (c)-(d), and (e)-(f). Segments (b), (d) and (f) were recorded during halothane anesthesia, narcotic anesthesia and enflurane anesthesia, respectively. (after McEwen (23))

processing and display techniques must be developed to minimize the effect of interpatient variability and to reduce the amount of information presented.

The Augmented Delta Quotient (ADQ) used by Volhyesi (41) and the technique used in the Cerebral Function Monitor (37) both reduce each channel of the EEG to a slowly varying signal which changes according to certain EEG characteristics. The result is a considerable reduction in the amount of data generated when compared to conventional EEG recordings. A problem with these techniques, however, is that a given change in the ADQ or the CFM can correspond to a number of physiological changes. For example, changes corresponding to anesthetic depth can mimic changes corresponding to cerebral ischemia. In addition, changes in one characteristic of the EEG can balance out changes in another characteristic. Thus changes which may be significant will not be reflected. Finally, EEG signals have generally been described in terms of their frequency and/or time characteristics. Since the above techniques do not independently present either of these characteristics, the interpreter is not able to analyze the data in terms of commonly encountered feature descriptors.

Two major techniques have been used for eliciting frequency information from the EEG; period analysis and spectral analysis. Period analysis is based on counting the number of times the EEG crosses the isoelectric (0 voltage) line. However, it is necessary to collect a large number of data samples to obtain reliable frequency estimates using period analysis. Consequently, long time intervals must be used. These long intervals will delay the response to abrupt changes in the EEG. Also, EEGs with differing characteristics can produce the same zero crossing frequency. Spectral analysis, on the other hand, reliably provides frequency information by analysing short time intervals. Since each frequency band is independently presented, changes in one part of the spectrum will not obscure changes in another. For this project, spectral analysis was used. A discussion of the display formats used to present the time trends in the frequency information obtained from spectral analysis can be found in section 4.5.1.

Work has been done in automatic patient state assessment using pattern recognition techniques on the EEG (eg. 23,49). If a monitor could be designed incorporating these techniques, the anesthetist might be relieved of the task of EEG interpretation. However, the algorithms for performing these computations with a high degree of success are computationally demanding and are difficult to perform in real-time using existing computer hardware. If and when such a system is developed for real-time use, its acceptance will rely on the ease of recognition and rectification of any sources of erroneous estimates and on the system's ease of use.

4.4.2 Analog Preprocessing

In order to reduce the effect of low frequency movement artefacts, higher frequency EMG artefact and power line pickup (60 Hz), and to prevent aliasing during power spectral estimation, the EEG signals should be high-pass and low-pass filtered. In the EEGAL system, the EEG was high-pass filtered at .54 Hz (first-order high-pass built into the Beckmann Accutrace EEG machines) and low-pass filtered at 30 Hz (third-order butterworth low-pass Krohn-Hite 3342-R). Recordings from the EEGAL project were used in the demonstration of MOMA (see chapter 5). Since aliasing occurs at 32 Hz in MOMA (64 Hz sampling frequency), the low-pass filter of future versions will likely roll off at a lower frequency, ie. about 25 Hz. In addition to being filtered, the EEG must be amplified for compatibility with the A/D converter. The EEG is characteristically about 50 microvolts in amplitude and the A/D operates between plus and minus 5.12 volts, thus an amplification of about 10^5 is required.

4.4.3 Digital Preprocessing

Two digital preprocessing techniques were used in order to optimize the output of the power spectral estimator, which is based on the Fast Fourier Transform (FFT). Firstly, if a DC bias is present in the data segment to be processed, the FFT will produce a large DC element and small higher frequency elements. In order to prevent this, the mean of each data segment was removed prior to processing. The means are available to routines entered later in the same MOMA cycle but are not stored on disk. Secondly, in order to limit spectral leakage, a cosine taper was applied to the first and last ten percent of each data segment. Both of these operations are performed immediately on receipt of a data segment (ie. while the next segment is being acquired), so as to spread the computation related to signal processing more evenly within the MOMA cycle. This should result in smoother display processing and user interaction.

4.4.4 EEG Power Spectral Estimation Technique

Since its rediscovery in 1965 (50), the fast fourier transform (FFT) has become a fast and economical means for obtaining power spectral estimates (PSEs) of digitally sampled signals. Techniques using the FFT implicitly assume that the underlying analog signal is a stationary random process.

There are a number of considerations for choosing a PSE technique for this application. Firstly, in order to operate in real-time and to allow for display processing and user interaction, the technique must be reasonably fast. Secondly, since the spectral characteristics of the spontaneous EEG are generally described in terms of broad frequency bands, the PSE does not need to have high resolution. Thirdly, since the computer has a limited amount of random access memory (RAM), it is important to minimize the amount of memory required for spectral analysis. Finally, the amount of data that must be stored every MOMA cycle will affect the maximum operation length which may be recorded and the time required to recall each record from the disk.

Using a record segmentation technique presented by Welch in 1967 (51) it is possible to obtain a decrease in the variance of the spectral estimate by sacrificing spectral resolution. For example, if K equals the number of segments, there will be a PSE variance reduction of 1/K for Gaussian processes. A study comparing this and 4 other spectral estimation techniques (52) concluded that although some of the other techniques produce better results, the Welch method is a reasonable choice for time limited applications. Dumermuth (53) suggested that while this technique has some leakage problems, it may be useful for EEG processing in systems with limited memory capacity. Although other PSE techniques exceed the performance of the Welch scheme (eg. 54-57), they generally take more computation time and consume more memory.

4.4.5 Segment Size

An appropriate choice of segment length during power spectral estimation will result in convenient frequency increments in the derived power spectra. With this in mind, a segment length of one second was chosen. This produces a power spectrum with one Hz increments.

4.4.6 Record Size

The variance of the PSE produced by the Welch method decreases as the number of segments increases. However, as determined by McEwen (23), the validity of the assumption of stationarity of the EEG signal decreases as record length increases. Additionally, with the addition of every new segment, the required processing time and the minimum MOMA cycle increases. With these considerations in mind, a record size of eight seconds was chosen and was divided into eight one-second segments for processing.

4.4.7 Frequency Domain Smoothing

In order to improve the statistical characteristics of the PSE, a three point moving average window was used.

If $P(k)$ = the unsmoothed PSE $\quad k=0,1,\ldots,N-1$

and $S(k)$ = the smoothed PSE $\quad k=0,1,\ldots,N-1$

then
$$S(0) = .5P(0)+.5P(1) \tag{4.1}$$
$$S(N-1)=.5P(N-1)+.5P(N-2) \tag{4.2}$$
$$S(j)=.25P(j-1)+.5P(j)+.25P(j+1) \quad j=1,2,\ldots,N-2 \tag{4.3}$$

4.4.8 Sampling Rate

Studies relating the EEG spectral characteristics with patient state have suggested that the frequencies of interest are below 30 Hz (22). Since the signals must be sampled above the Nyquist rate, ie. twice the maximum frequency desired, the minimum sampling frequency is 60 Hz. Also, for good computational performance of the FFT, the number of samples per data segment should be a power of two. Using a segment size of one second, three sampling rates which satisfy the above two constraints are 64 Hz, 128 Hz, and 256 Hz. In order to minimize memory requirements and processing time, the lowest of these sampling rates was used (64 Hz).

4.4.9 Summary of EEG Processing

For a given 30 second MOMA cycle, each EEG channel is processed in the following manner:

1) Each signal is amplified, high-pass filtered at .54 Hz, and low-pass filtered at 30 Hz.

2) An eight second record is acquired by the computer through the A/D converter which samples at 64 Hz.

3) Each record is partitioned into eight one-second nonoverlapping segments.

4) The mean is removed from each segment.

5) The first and last 10 percent of each segment is tapered using a cosine taper.

6) A power spectral estimate is computed for each EEG channel using a data segmentation technique suggested by Welch (51).

7) Each estimate is smoothed using a three point moving average window.

8) The smoothed estimate is stored on disk and sent to the display processing routines along with various other program parameters.

4.4.10 Specifics of the Spectral Estimation Technique

Let:

$X(j)$ = the EEG data record to be processed. $j=0,1,\cdots,N-1$

N = the length of each record.

L = the length of each segment.

K = the number of segments.

D = the distance between the first sample of adjacent segments.

$X_i(j) = X(j+(i-1)D)$ = that i'th segment. $j=0,1,\ldots,L-1$

$W(j)$ = the time domain window. $j=0,1,\ldots,L-1$

$A_k(n)$ = the fourier transform of $X_k(j)*W(j)$. $n=0,1,\ldots,L/2$

$I_k(f_n)$ = the modified periodogram of $X_k(j)*W(j)$ $f_n=0,1/L,\ldots,1/2$

$P(f_n)$ = the unsmoothed power spectral estimate $f_n=0,1/L,...,1/2$

$S(f_n)$ = the smoothed power spectral estimate $f_n=0,1/L,...,1/2$

Assume that X(j) is a stationary second order stochastic process and that E(X)=0. In order to decrease the computational load and to simplify data buffering, nonoverlapping data segments were chosen. In addition, the entire record must be covered by the data segments to insure that all acquired data is used. Thus,

$$D = L \text{ and } N = (K-1)*D+L = K*L. \tag{4.4}$$

For this project, N=5.2, L=64, K=8, and D=64.

The processing technique procedes as follows. After applying the data window W(j) to each data segment, the fourier transform of each data segment $A_k(n)$ is computed.

$$A_k(n) = \frac{1}{L}\sum_{j=0}^{L-1} X_k(j)W(j)e^{-2kijn} \tag{4.5}$$

where $i=(-1)^{\frac{1}{2}}$ and n=0.1.,,,.L/2.

The modified periodograms $I_k(f_n)$ are then derived from the fourier transforms.

$$I_k(f_n) = \frac{L}{U}\left| A_k(n)^2 \right| \tag{4.6}$$

where $f_n=\frac{n}{L}$, n=0,1,...,L/2, and $U=\frac{1}{L}\sum_{j=0}^{L-1} W(j)^2$.

The unsmoothed power spectral estimate $P(f_n)$ is the average of the modified periodograms. Thus,

$$P(f_n) = \frac{1}{K}\sum_{k=1}^{K} I_k(f_n) \tag{4.7}$$

Finally, $P(f_n)$ is smoothed using the three point average window described in section 4.4.7 to produce the smoothed spectral estimate $S(f_n)$.

A more detailed description of the above processing technique can be found elsewhere (51).

4.5 EEG Display

4.5.1 Discussion of Techniques for EEG Presentation

In any attempt to portray the time trends of the frequency spectrum of EEG signals, three dimensions must be portrayed; frequency, time and magnitude. Since the display mediums presently used have only two spatial dimensions, that data must either be projected onto two dimensions or a non-spatial dimension such as color or intensity must be used.

Representation by projection is the technique devised by Bickford (58) and has been used in a number of clinical areas (eg. 59,60). It has been called the Compressed Spectral Array (CSA). This technique simply maps both the time and magnitude onto the Y axis, ie. the standard amplitudes vs frequency plot is generated for each sample period but the origin for each successive plot is moved up the Y axis. In order to give the displayed EEG the appearance of being a three dimensional contoured surface, a hidden line algorithm is used to prevent new plots from crossing below previous plots.

Although the CSA is an aesthetically pleasing display it has a number of drawbacks which limit its effectiveness for intraoperative monitoring Firstly, the presence of a large component in a given spectral plot may, because of the hidden line algorithm, cover usable data in succeeding spectral plots. Secondly, since time and amplitude share an axis, it may be difficult to determine the time at which a given frequency characteristic was acquired.

A technique described by Fleming (61) creates a third dimension by using dot density to represent magnitude. This has been called the Dot density representation of the compressed Spectral Array (DSA). By using three orthogonal dimensions (X,Y, and intensity) the problems described for the CSA are avoided. A comparison of the CSA and the DSA can be found in figure 10. Both the CSA and the DSA were implemented in the EEGAL project (46). The opinion of researchers involved with EEGAL was that the DSA showed more potential as a display format. For these reasons, the DSA was used in this project. The same conclusion was reached in a recently published paper techniques for intraoperative monitoring (62), including the DSA, CSA, CFM, and ADQ. The CFM and ADQ techniques are discussed in section 4.4.1.

4.5.2 DSA Implementation

The terminal's character format flexibility makes it well suited for a DSA display. It is possible to define alternative character fonts, which are tables of 8 * 14 point dot matrix patterns. Each character is displayed when a given eight bit code is received from the computer. Any line in the workspace region of the terminal can display characters from either the standard alphabet or alternative character fonts or both. In order to produce a DSA display, even numbers of dots from 0 to 112 were scattered randomly within the 8 * 14 matrices to produce 57 gray levels. These gray levels are associated with the ASC11 characters with decimal codes of 33 to 89 inclusive. The magnitude of each element of the power spectral estimate is linearly encoded to one of these gray levels. Power spectral elements which exceed the upper to lower bounds of this relation are encoded to the maximum intensity or minimum intensity characters respectively. For the purpose of generating axes for the DSA, three additional characters were defined. The dot matrix patterns used to form this alternative character font can be found in Appendix 5 and the output of a character font testing routine (FNTTST) can be found in figure 11.

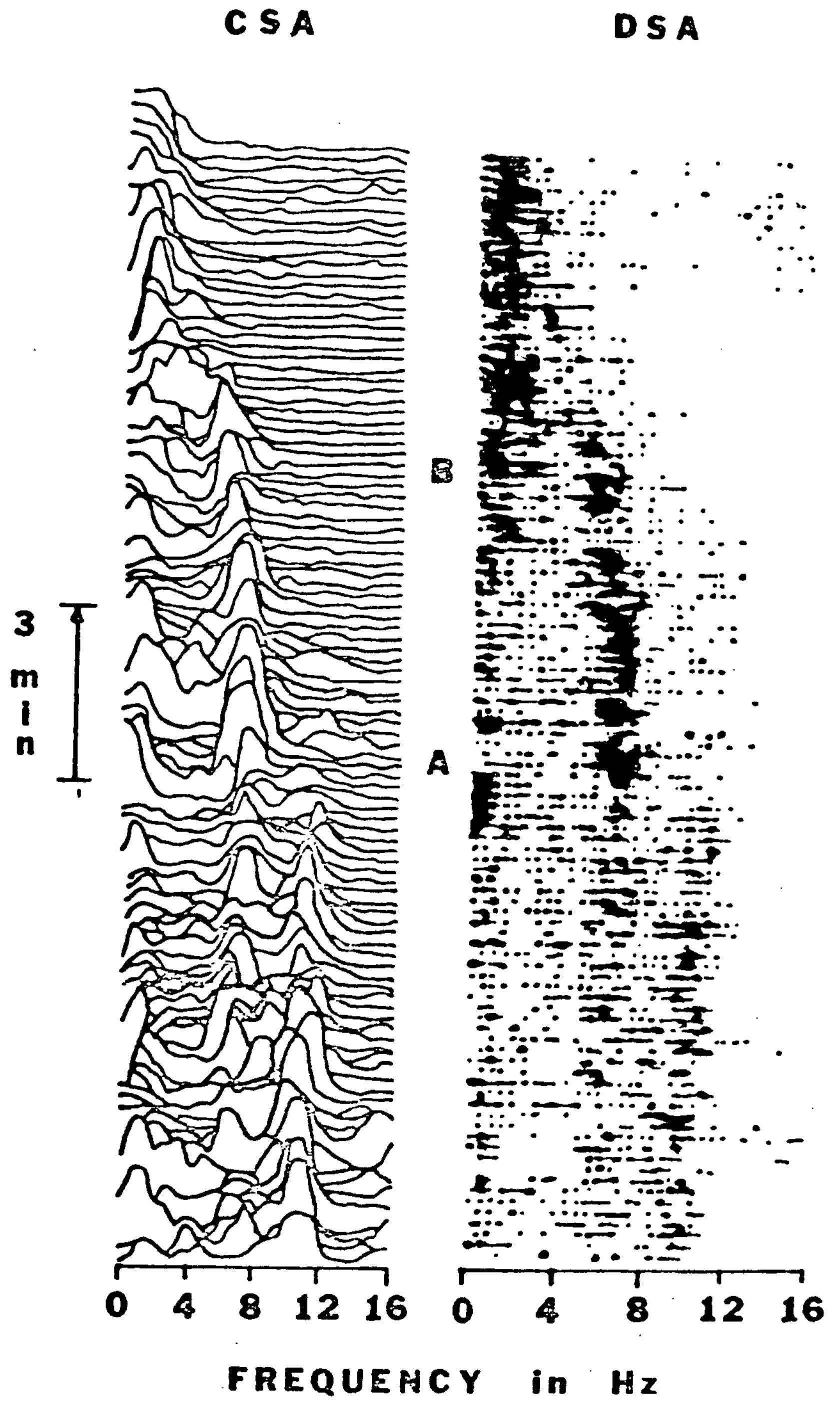


Figure 10: A Comparison of the CSA and DSA display formats (after Levi (62)). Recordings are from a patient during gradually deepening Halothane-N_2O anesthesia

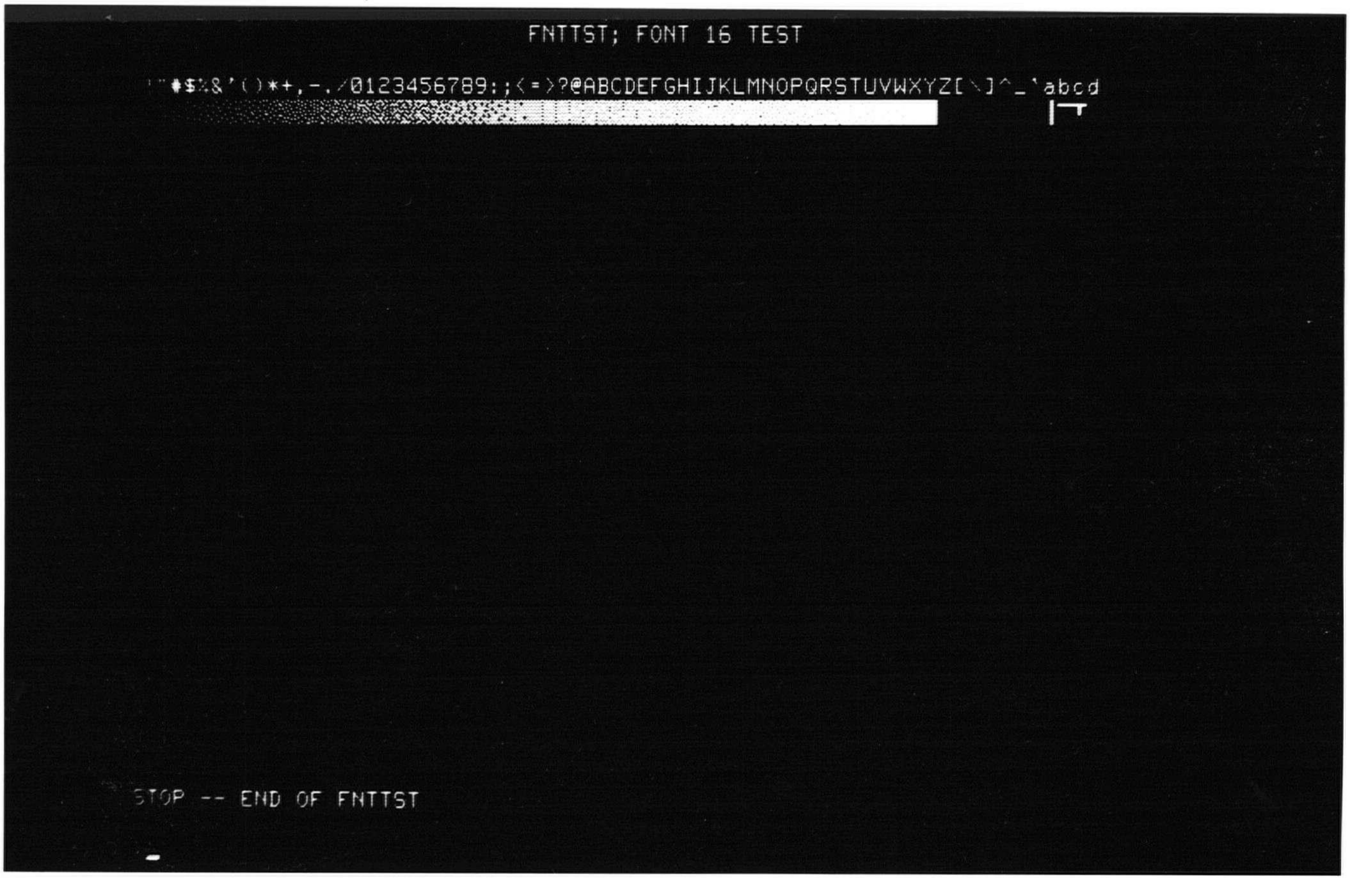


Figure 11: The output of the alternative character font testing routine FNTTST.

The fact that the graphics mode of the terminal need not be used in the DSA provides some additional bonuses. Firstly, the amount of the terminal's memory required to display a given number of spectra is substantially lower than would be required for a CSA display, thus more information can be displayed at one time. Secondly, since no fixed graphics region need be specified, the scrolling capabilities of the terminal may be more effectively used. Finally, the displays can be updated by simply overwriting data rather than redrawing a vector as would be necessary in the graphics mode. This results in more efficient and faster display processing.

4.5.2.1 The Effect of the Human Visual System

As described by Overington (63), the response of the human visual system's receptors is roughly proportional to the log of their illuminance. For this project, the intensity of the gray levels varies linearly and the power spectra are linearly encoded to these gray levels. Consequently, the human visual system effectively takes the log of the displayed power spectra. Although variables such as background luminance and structure, accommodation, border enhancement, etc. are not accounted for, the fact remains that the perceived intensity difference between two adjacent low intensity gray levels is greater than the perceived intensity difference between two similarly separated levels of a higher intensity.

4.5.3 Special Features

A summary of the EEG presentation format described below can be found in figure 12. The DSA displays implemented in MOMA are divided into two sections; a scrolled section and an unscrolled section. When presenting real-time data, the scrolled section is automatically updated when new data is recieved. While in a scrolled state (described below), however, the automatic update is inhibited. At the user's request, the scrolled section can be moved back and forth in time by a specified number of records. ie. if the scrolled section is presenting the L'th to the L-M'th EEG records, it can be moved forward in time so that it presents the L+N'th to the L-M+N'th EEG records or, it can be moved back in time so that it presents the L-N'th to the L-M-N'th EEG records. Any attempt to scroll to before the start of the operation or into the future is ignored. Even though display update is inhibited while in a scrolled state, data continues to be acquired and stored. Thus, when the display is returned to real-time, no data is lost. Using the scrolling feature, it is possible to access any EEG record stored earlier in the operation.

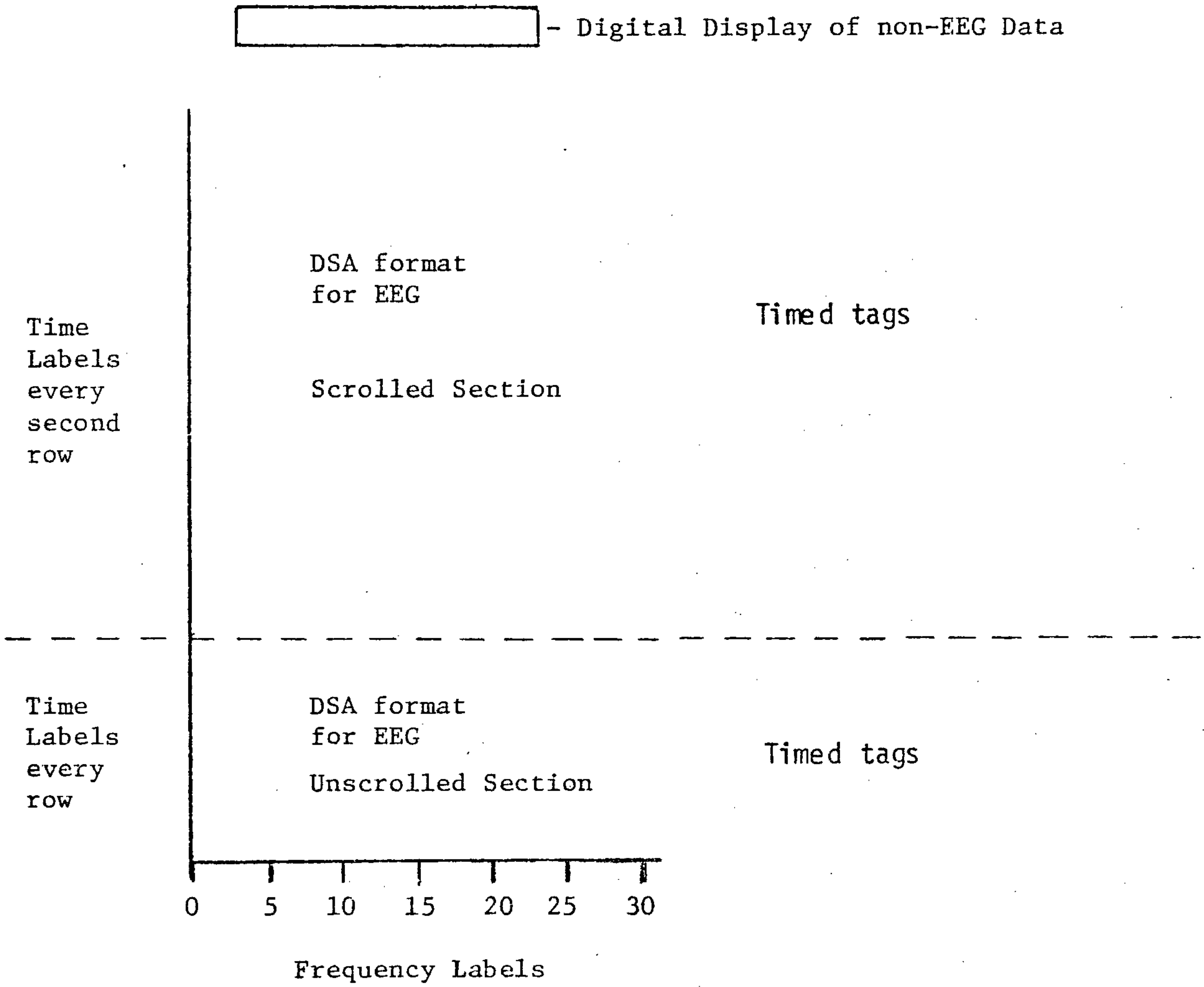


Figure 12: Format of the displays presenting the DSA.

The unscrolled section of the DSA displays is loaded by transferring an EEG record from the top of the scrolled section to the top of the unscrolled section. Records transferred, or saved, in this manner are not affected by future update, scrolling, or display change processes. This feature allows the anesthetist to perform comparisons between EEG records separated by long periods in time and should help in identifying long term trends in the EEG.

4.6 The Display of Non-EEG Parameters

To permit comparisons between non-EEG and EEG data, non-EEG data is displayed digitally in all the EEG displays. The digital display of non-EEG data always corresponds to the most recent EEG record on the screen.

During the observation of various surgical procedures, the author noted that the large digital displays of certain parameters, such as mean blood pressure and pulserate, are used by both the anesthetist and the surgeon, sometimes at a distance from the monitor. Since these displays are viewed at a distance, the small digital display of non-EEG data on MOMA will not take the place of the larger displays. However, since MOMA's digital displays are always associated with the most recent record on the screen, they are useful when the EEG is scrolled.

A display format was generated which presents the time trends in the non-EEG data. In this format, up to four channels of non-EEG data can be plotted vs time on the terminal's screen. This is equivalent to the paper record routinely maintained during all operations at VGH. Thus, in addition to making the time trends and interrelationships of non-EEG parameters more readily apparent, this facility may relieve the anesthetist of some of his record keeping duties. A summary of the non-EEG time-trending display format can be found in figure 13.

4.7 Timed Tags

Many events and conditions in the operating room must be considered when the data acquired and stored by MOMA is interpreted. In order to associate these events with the automatically acquired data, a timed tagging facility was implemented. Using this facility, it is possible to enter up to 255 typed comments to the computer. These comments, or timed tags, are

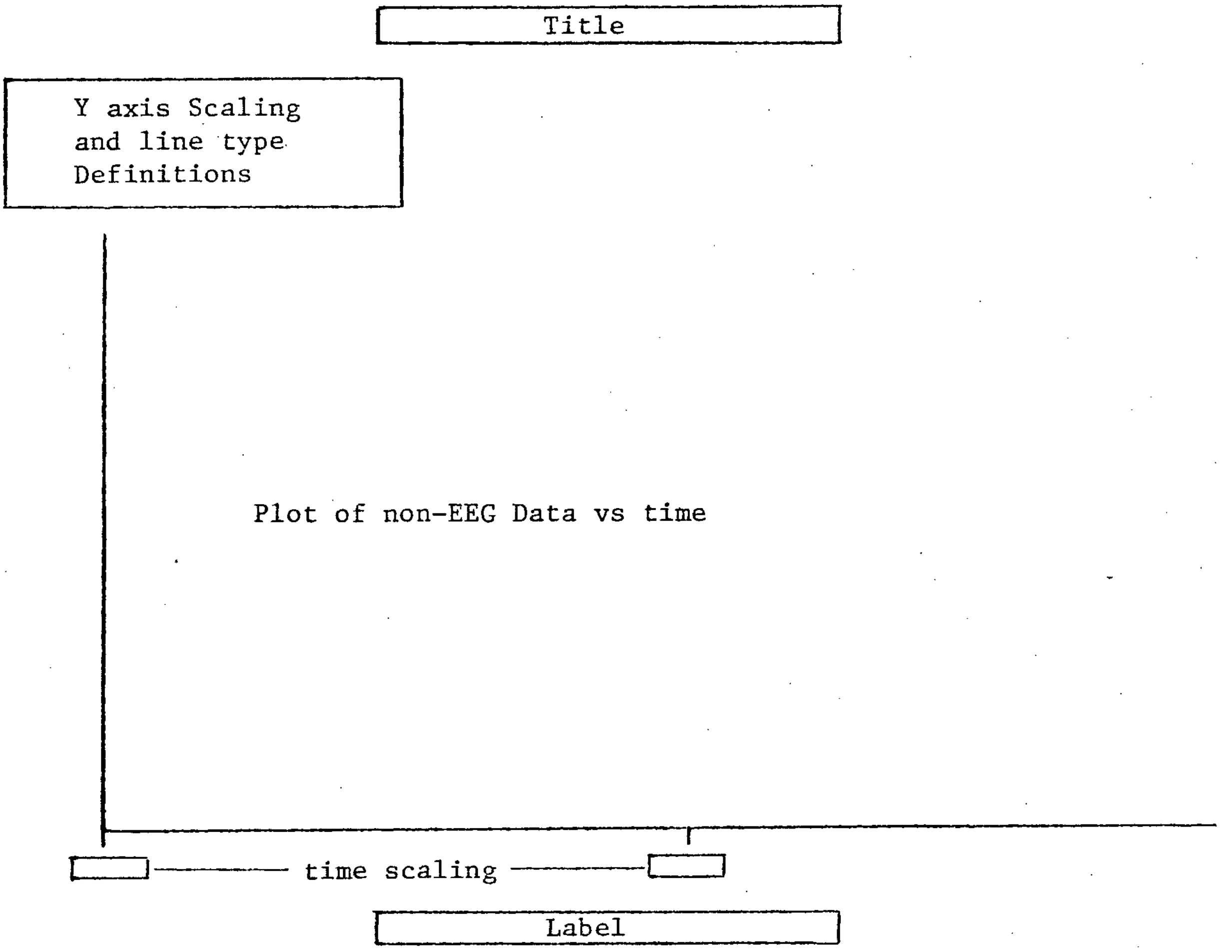


Figure 13: Format of the displays presenting time-trends of the non-EEG data.

associated with the next acquired record, stored on disk, and displayed beside the DSA in the EEG displays. The tags are truncated to 32 characters prior to storage on disk.

4.8 The Terminal Screen's Configuration

As shown in figure 14, the terminal's screen has been split into two regions, the workspace scroll and the monitor scroll. Characters typed into the workspace scroll from the keyboard are not sent to the computer but characters typed into the monitor are. The information contained in one scroll is independent of the information contained in the other.

The MOMA programs use the workspace scroll for the presentation of preformated displays. During the startup procedure, this scroll is loaded by the computer with checklists and startup instructions. During operation, the computer sends the displays to this scroll. For the displays presently implemented, the workspace scroll occupies the top thirty lines of the terminal's screen.

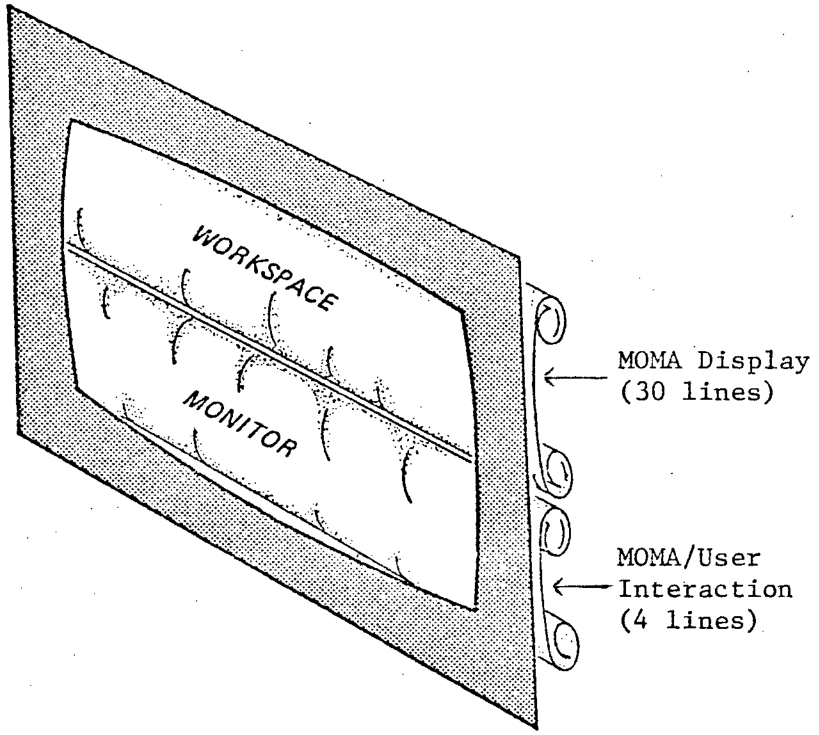


Figure 14: The structure of the Tektronix 4025 terminal's screen.

The MOMA programs use the monitor scroll for user interaction. The computer sends all messages and requests for servicing to this scroll. These requests are accompanied with a bell so as to attract the user's attention. In addition, all entries from the keyboard are echoed to this scroll. For the displays presently implemented, the monitor scroll occupies the bottom four lines of the terminal's screen.

4.9 Implemented Displays

As shown in figures 15 to 18 inclusive, four displays were implemented which combine the DSA with timed tags and non-EEG data. The non-EEG data is presented digitally at the top of the screen and is always associated with the most recent DSA record on display. The timed tags, if present, are typed on the right hand side of the DSA records. Two of these displays (D1 and D3) contain a seven line scrolled section and a three line unscrolled section for each EEG channel. The other two displays (D2 and D4) contain a 15 line scrolled section and a five line unscrolled section for each EEG channel. When these displays are updated, the new EEG records and timed tags are inserted into the top of the scrolled section and the digital non-EEG portion of the display is overwritten.

As shown in figures 19 and 20, two displays were implemented which plot the non-EEG data vs time by making use of the graphics routines. One of these displays (D5) plots the last 79 samples of each non-EEG signal and the other (D6) plots each non-EEG signal from the start of the operation to present. It is expected that the system will not be presenting these displays for long periods of time. For this reason, it was not deemed necessary to automatically update these displays. There is, however, no technical reason limiting the implementation of a display update capability.

4.10 User Interaction

An effort has been made to minimize the time required to use and to learn how to use the system. Towards this end, the amount of required interaction with MOMA was minimized. When interaction is necessary, a message is typed into the monitor scroll which describes the problem and the appropriate response(s). A bell is rung when the message is sent so as to

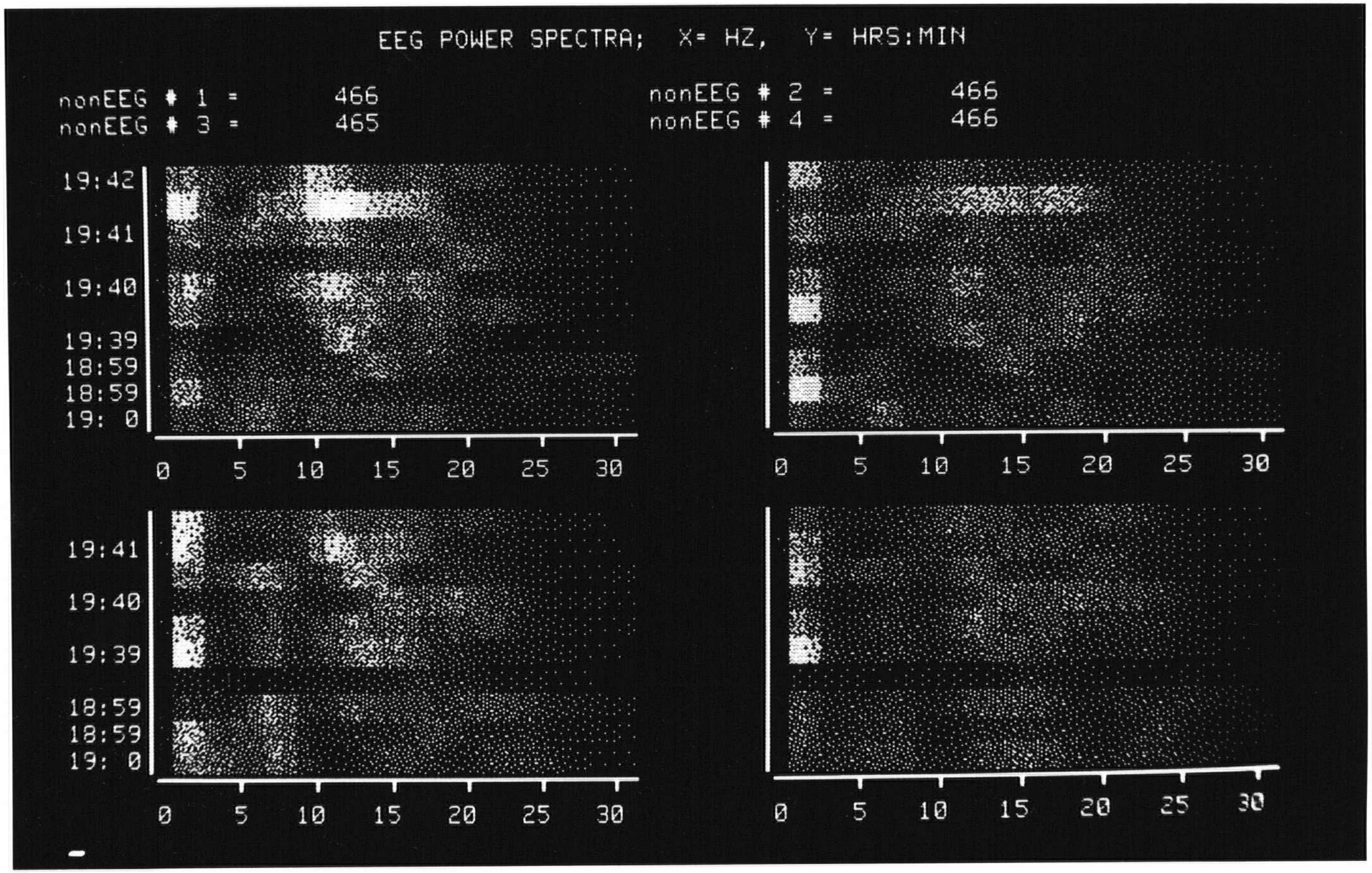


Figure 15: MOMA Display Number 1 (D1)
- 10 lines of DSA for four EEG channels
  - the scrolled section = the top seven lines
  - the unscrolled section = the bottom three lines
- Timed tags truncated to 8 characters
- Digital display of data from up to four non-EEG channels

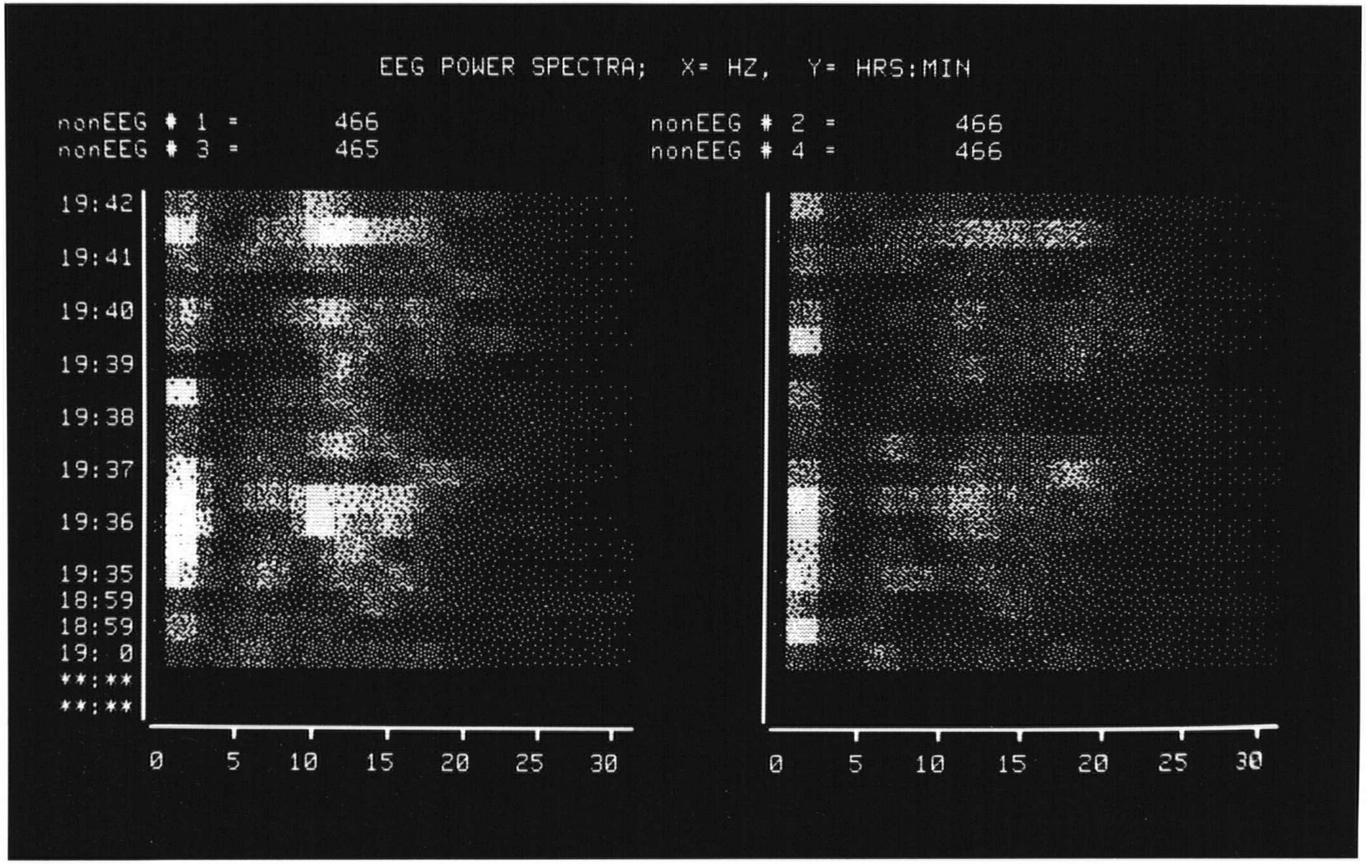


Figure 16: MOMA Display Number 2 (D2)
- 20 lines of DSA for two EEG channels
  - the scrolled section = the top 15 lines
  - the unscrolled section = the bottom 5 lines
- Timed tags truncated to 8 characters
- Digital display of data from up to four non-EEG channels

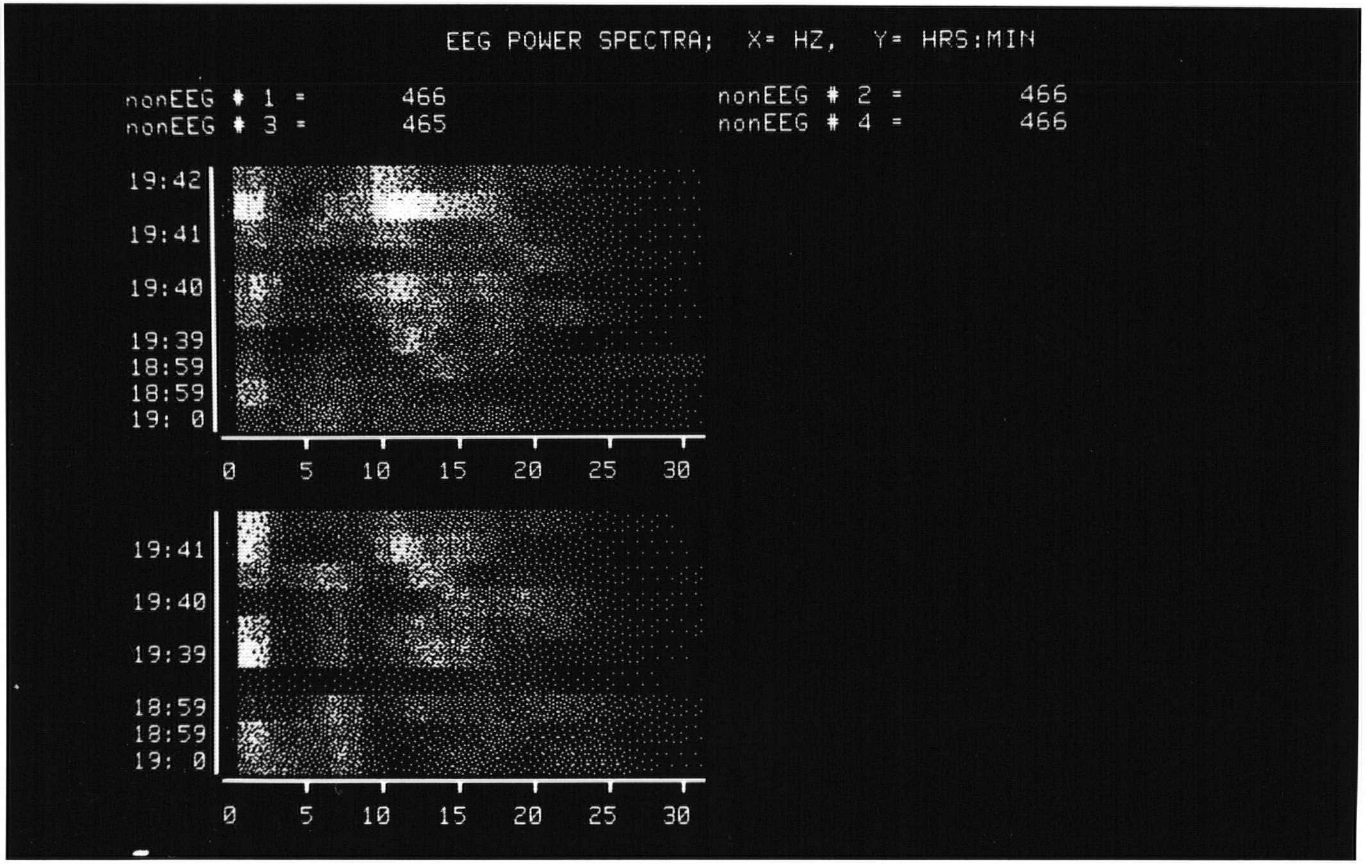


Figure 17: MOMA Display Number 3 (D3)
- 10 lines of DSA for two EEG channels
  - the scrolled section = the top 7 lines
  - the unscrolled section = the bottom 3 lines
- Timed tags truncated to 32 characters
- Digital display of data from up to four non-EEG channels

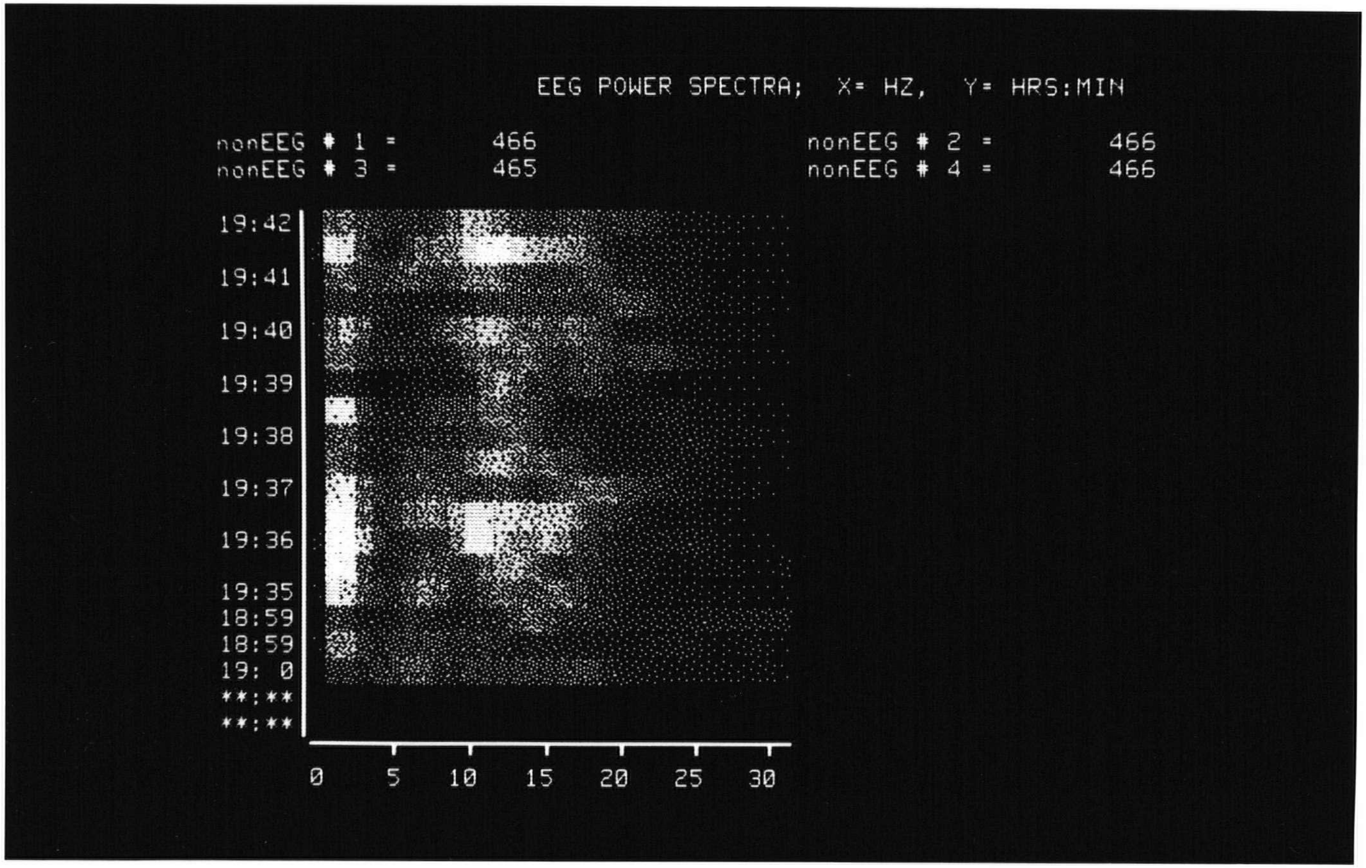


Figure 18: MOMA Display Number 4 (D4)
- 20 lines of DSA for one EEG channel
  - the scrolled section = the top 15 lines
  - the unscrolled section = the bottom 5 lines
- Timed tags truncated to 32 characters
- Digital display of data from up to four non-EEG channels

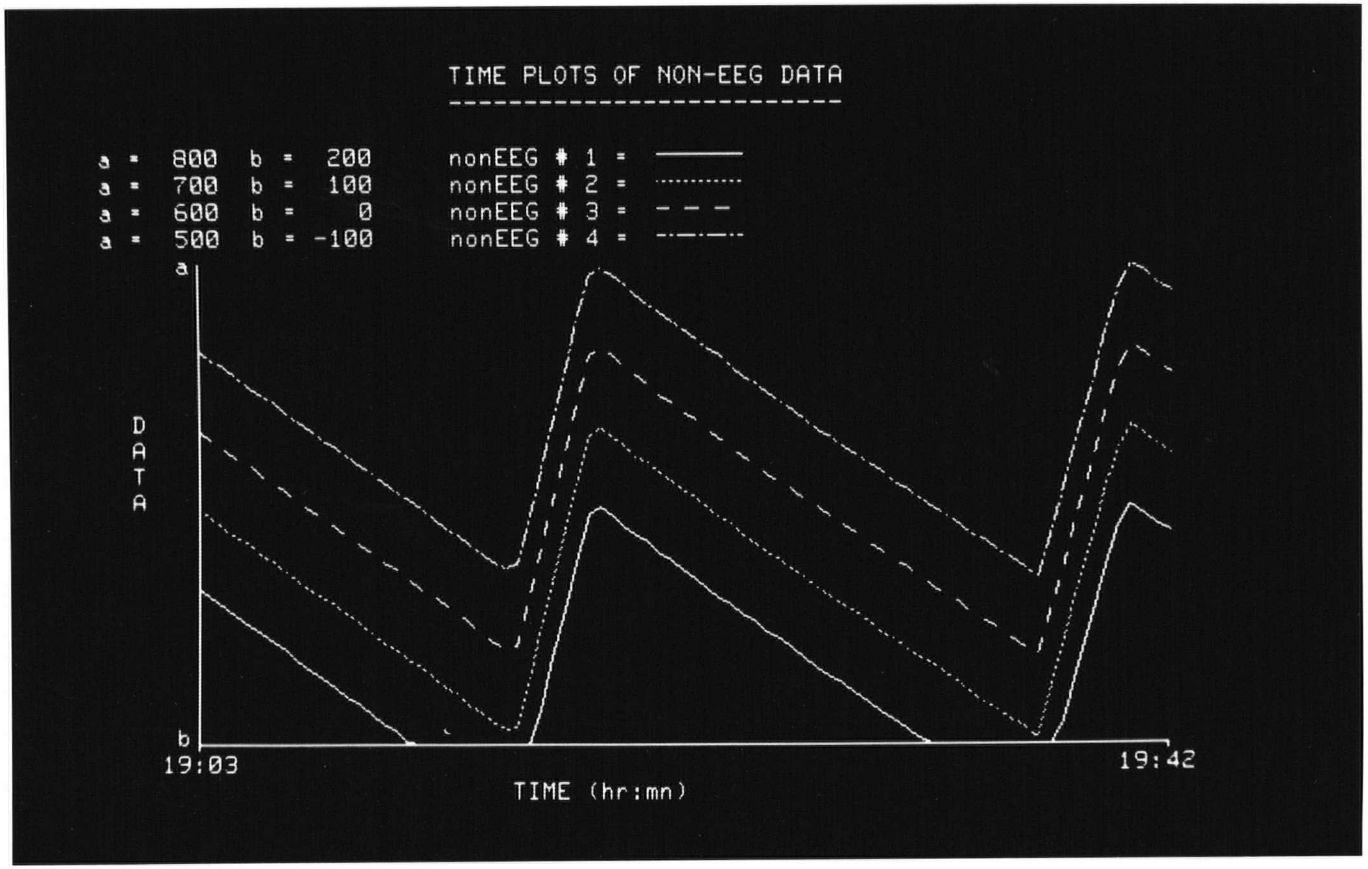


Figure 19: MOMA Display Number 5 (D5)
- Plot vs time of data from up to four non-EEG data channels
- Presentation of the last 79 samples

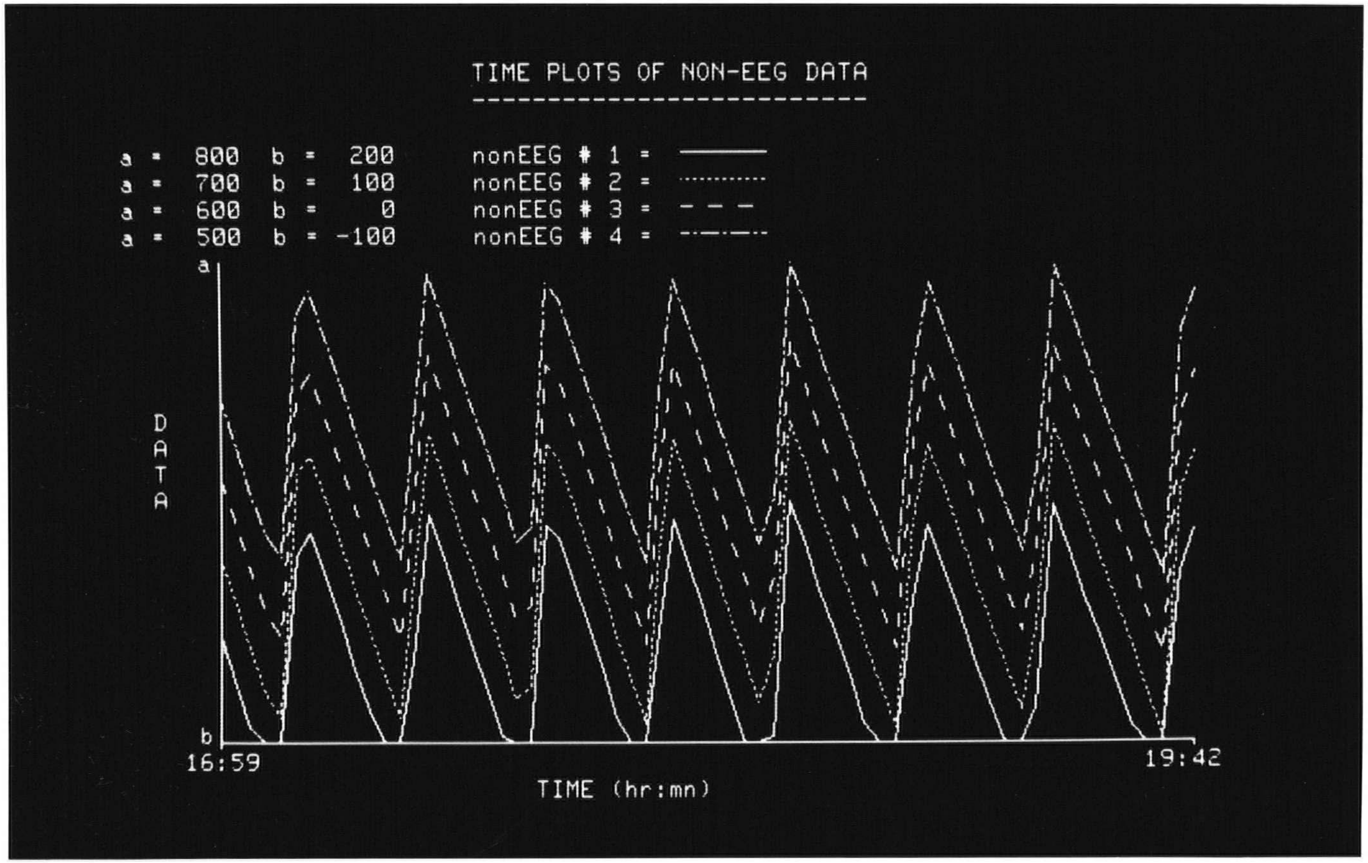


Figure 20: MOMA Display Number 6 (D6)
- Plot vs time of data from up to four non-EEG data channels
- Presentation of data from the start of the operation to present

attract the user's attention. The length of the required response is also minimized, often entailing the typing of a single key.

In addition to minimizing the amount of required interaction, importance has been placed on simplicity of use of MOMA's optional features. Firstly, commands are entered by typing a single labelled key. Secondly, the command keys are grouped according to their function so as to simplify the task of locating them. Thirdly, the user is informed if he attempts to use an illegal command or the unshifted version of a shifted key instruction. Finally, documentation about the features and operation of MOMA was written and will accompany the system within the operating room (see appendix 3).

In an attempt to increase the fault tolerance of MOMA, several features were implemented. Firstly, to generate some of the commands, two keys must be depressed simultaneously, thus reducing the likelihood of accidental entry of those instructions. Secondly, requests for certain irreversible instructions, such as termination or timed tags, are serviced only if the user responds to a query with a "Y". Thirdly, if for any reason the terminal fails to respond properly during operation, it is possible to reinitialize it without stopping the program. Finally, if the program stops during the operation, it is possible to restart it using the standard startup procedure. All data acquired prior to termination remains intact and is accessible after restart.

4.11 The Command Structure

MOMA commands can be entered using two methods, by typing a command string or by using the preprogrammed command keys. As shown in table 1, the command strings always start with the character "{" and are terminated with "RETURN". Any string starting with the character "{" is interpreted as a

Table 1: Moma Command Summary

KEY KEY	COMMAND STRING	FUNCTION
SCROLLING:		
"FOR"	{S.	Set scrolling direction to forward
"BACK"	{SO	Set scrolling direction to backward
KEYPAD 1-9	{S1 to {S9	Scroll display N records
"PRESENT"	{SP	Return display to present time
TIMED TAGGING:		
ALPHANUMERIC	---------	Timed tag entry
"RETURN"	--------	End of timed tag
"RUBOUT"	--------	Delete last character typed
CNTL/U	--------	Delete partially entered timed tag
"GET LINE"	--------	Reactivate last timed tag typed
"ON QUERY"	{Q	Reactivate query generation on timed tag entry
"OFF QUERY"	{N	Deactivate query generation on timed tag entry
DISPLAY CHANGE:		
"D1" to "D4"	{1 to {4	Change to EEG display
"D5" to "D6"	{5 to {6	Change to non-EEG display
SPECIAL FUNCTIONS:		
"INIT"	{I	Reinitialize terminal
"STOP"	{F	Stop MOMA programs
"SAVE"	{T	Transfer EEG record from scrolled to unscrolled section
------	{E	Generates the message "use SHIFTED VERSION of that key". Programmed into the unshifted version of the shifted special function keys (D1-D6,INIT,STOP,SAVE).

command. If an illegal command is entered, the user is informed and no action is taken.

4.12 The Keyboard Configuration

As shown on figure 21, all special function keys are labelled. In the keypad and timed tagging sections, these labels are written on top of the keys. In the display and special function section, keyboard overlays which are placed onto the keyboard are provided. If the overlay label is above the key, the shift key must be held down while that key is pressed. If the overlay label is below the key, the shift key is not held down.

The terminal is initialized when the system is started. During this procedure, the internal modes of the terminal are set, the special function keys are programmed, and the character font used for the DSA displays is initialized. If these settings are altered during the operation the terminal can be reinitialized using the "INIT" instruction. Displays are NOT erased during this procedure.

The scrolling feature uses the keypad portion of the keyboard. Through this pad the user can change the direction of scrolling by typing the "FOR" or the "BACK" key. The scrolled section of the DSA displays can be scrolled backward or forward one to nine records using the keypad numeric keys. The digital non-EEG display and the timed tags are also altered and automatic update is disabled during this process. When the "PRESENT" key is typed, the display is returned to real-time, automatic display update is re-enabled, and the scrolling direction is set to "BACK". Any attempt to scroll into the future or to before the start of the operation and the redundant typing or "PRESENT" is ignored.

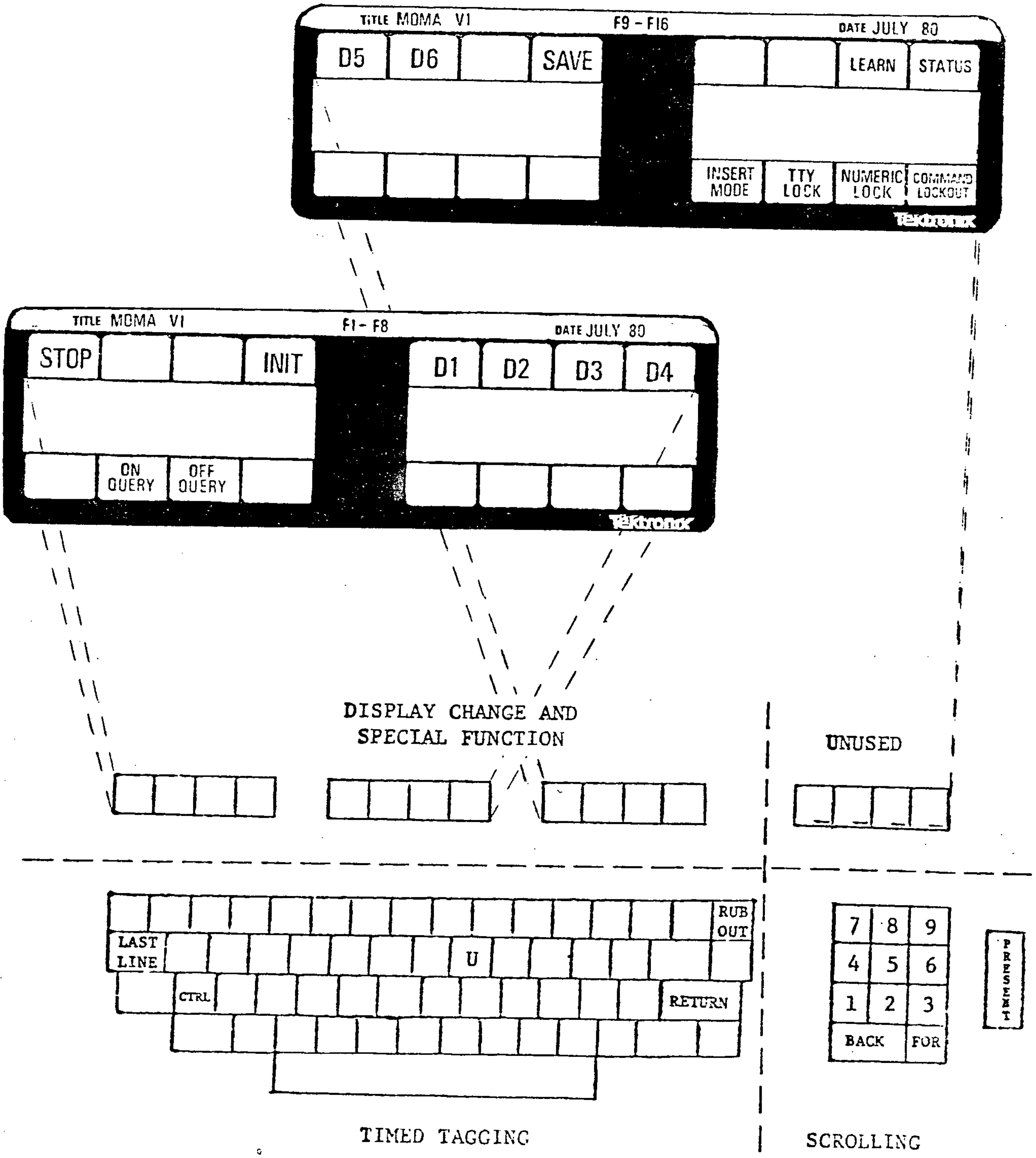


Figure 21: MOMA's keyboard configuration.

The keys which generate the display change instructions are labeled "D1" and "D6" on the keyboard overlays. In order to provide a quick reference for the displays associated with these keys, a brief display summary was attached to the front of the terminal (see figure 22). See figures 15 to 20 for photographs of the displays.

As described in section 4.5.3 the "SAVE" key is used to copy a DSA record and timed tag from the top of the scrolled section of the DSA display to the top of the unscrolled section of the DSA display. The saved record is not disturbed by subsequent display change or manipulation. Automatic loading and manipulation of the unscrolled section of the DSA display is possible. For this prototype, however, the only way to alter the contents of the unscrolled section is through the use of the "SAVE" key.

The MOMA programs are terminated using the "STOP" key. When this request is sent, the user is asked to confirm that the request is valid before termination is carried out. The writing of a period onto the terminal's screen indicates that the termination process is complete. At this point, RT-11's KMON program is running and standard RT-11 instructions can be entered.

Timed tags are entered through the alphanumeric portion of the keyboard by typing an appropriate comment followed by "RETURN". The only restrictions on the form of these comments is that they cannot start with the character "{" (MOMA command character) and cannot contain the character "'" (terminal's command character). Entered timed tags are truncated to 32 characters on entry. After a timed tag is entered, the program rewrites it onto the monitor scroll and asks the user to accept or reject it. If desired, the user can suppress this query by pressing the key labelled "OFF QUERY". Timed tags are then stored immediately on the typing of "RETURN". Query reactivation was performed by pressing the "ON QUERY" key. Three tag

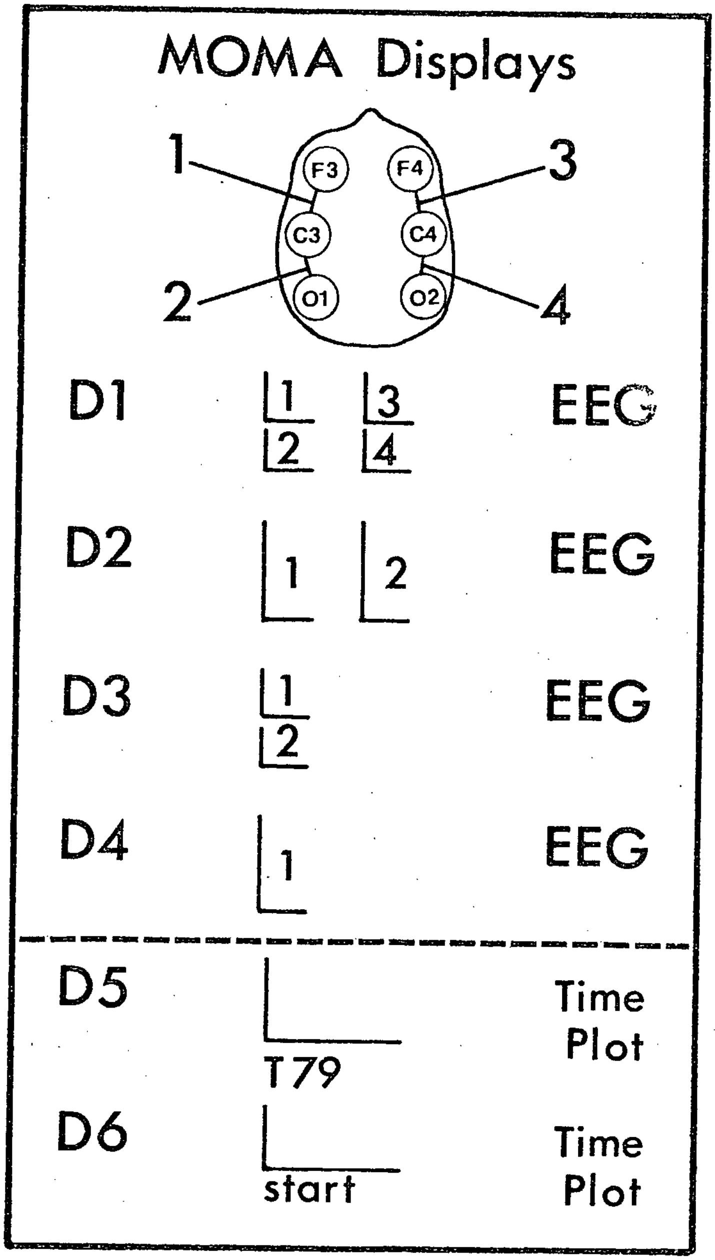


Figure 22: A guide for the MOMA displays. This guide was fastened to the front of the terminal beside the screen.

manipulation instructions were implemented. Firstly, the "RUBOUT" key deletes the last character typed, thus permitting the correction of typing errors. Secondly, the entry of CTRL/U deletes the currently active (partially completed) timed tag. Finally, the "LAST LINE" key reactivates the last timed tag entered.

The TAB, ERASE, BACKSPACE, and LINEFEED keys were not used in MOMA. In order to prevent visual confusion, these keys were fitted with blank key-caps. The accidental typing of these keys has no effect on MOMA.

5. SYSTEM PERFORMANCE

5.1 System Assessment

5.1.1 Rationale

For the purpose of assessing the potential usefulness of MOMA in the operating room, a demonstration of the system's features was arranged for anesthetists practicing at VGH. There were a number of reasons for choosing this form of assessment. Firstly, if MOMA is to be effectively used, the anesthetists should be familiar with its characteristics and the use of its features prior to entry to the operating room. It was felt that the most efficient means for the anesthetist to gain this familiarity was through a system demonstration. Secondly, using this format, the anesthetists will be able to concentrate solely on assessing the device. Thirdly, since the objective of the work done for this thesis was to produce a device from which an intraoperative monitor could be developed, it was thought that the potential usefulness of the features, rather than the clinical effectiveness of the prototype should be assessed.

One alternative to an assessment by demonstration would be to assess the system within the operating room. Much of the information required for an overall assessment of MOMA's present clinical usefulness can only be obtained through its use in the operating room. However, the time required to perform a proper assessment involving a number of anesthetists each using the device during a number of surgical procedures is beyond the scope of this work. If an abbreviated intraoperative assessment were used (involving one anesthetist and a few operations), the results might be biased by factors such as initial unfamiliarity with the system or inappropriate packaging features in the prototype. These limitations, combined with a desire to obtain the opinions of a number of anesthetists suggested that clinical assessment by demonstration rather than trials would be more appropriate for this work.

5.1.2 Procedure

Nine days prior to the demonstration, ten anesthetists at the Vancouver General Hospital were given a summary of the system's features, a user's guide describing those features in greater detail and providing instructions for operation of the system, and a copy of a structured questionnaire. (See appendix 3 for the summary and the users guide and appendix 4 for the questionnaire.) An introductory letter informed them that the purpose of this demonstration was threefold; to familiarize them with the features of the currently available system, to assess the potential usefulness of these features, and to gather ideas about the future direction of this project. Five of these anesthetists participated in the assessment.

During the demonstration, the author demonstrated and described the system's characteristics. Four channel EEG recordings from the previous EEGAL project were played back using a Hewlett Packard 3960 instrumentation recorder and were acquired, stored, and displayed by MOMA. An Anatek DC power supply was used to simulate non-EEG data. The format of the demonstration was intentionally informal, the anesthetists were urged to ask questions and were given the opportunity to use the system. Finally, each anesthetist was asked to complete the structured questionnaire and to return it to the Biomedical Engineering Department at the Vancouver General Hospital via the hospital mail system. A photograph of the hardware used for this demonstration can be found in figure 23.

The structured questionnaire used in the demonstration was divided into four sections. The first section was for biographical data, the second

Figure 23: The Hardware used for the system assessment.

section contained questions intended to elicit ideas for the future direction of the project and is discussed in chapter 6, the third section contained questions for the assessment of MOMA, and the fourth section provided space for general comments. In order to prevent existing technical difficulties from biasing the anesthetists' rating of the relative importance of various parameters, they were asked to disregard difficulties in obtaining signals appropriate for computer acquisition. In addition, they were asked to indicate when an answer applied only to a specific procedure or procedures.

5.3.3 Results

The results of the assessment of MOMA's features (question 3.1) are tabulated in table 2. Considering the dissimilarity of MOMA with currently used monitors the results of this assessment were favorable. The multiple display and the time plot features obtained the highest rating (4.6/5 and 4.4/5 respectively) and simplicity of operation received the lowest rating (3.0/5). The low rating of simplicity of operation was largely due to the response of two of the anesthetists (A and C). However, these same anesthetists suggested that after a period of familiarization, the information and trends obtained from MOMA should more than compensate for the time required to operate it. The overall assessment for MOMA was 4.2/5.

There was some hesitation on the part of the anesthetists in answering questions 3.3 to 3.5. They felt that more precise answers could only be given to these questions after a comprehensive device assessment. With this provision, all anesthetists felt that the combination of the scrolling feature and the two section display would be sufficient for performing the comparisons necessary for using the EEG as a monitoring parameter in the operating room. Also, with one exception, no ambiguities or unexplained characteristics were found in the documentation by the anesthetists. This ambiguity was clarified during the demonstration. The suggestions for possible

Table 2: System Assessment results

QUESTION: Give your initial impression of the potential usefulness of the following features of MOMA:

RESULT: 0 = do not know 1 = poor 5 = excellent

CATEGORY	ANESTHETIST A	B	C	D	E	AVERAGE
Multiple display	5	5	5	4	4	4.6
Scrolling of EEG displays	3	5	3	4	4	3.8
Two section EEG displays	3	5	3	4	3	3.8
Time plots	5	5	5	4	3	4.4
Timed tagging	3	5	3	4	3	3.8
Simplicity of operation	2	0	2	4	4	3.0
Keyboard configuration	4	0	4	4	4	4.0
Documentation	4	3	4	4	4	3.8
Overall assessment	4	5	4	4	4	4.2

modification (question 3.5) are listed later in this section.

Question 3.6 asked the anesthetists if they felt that MOMA would take a disproportionate amount of their time away from more essential tasks. One rater suggested that it would not, two raters suggested that after becoming familiar with the features and capabilities of the system, the extra information and trends obtained would more than justify the time required to operate it, one rater suggested that under normal conditions it would not but that in a crisis situation it might, and the last rater suggested that the system would be most appropriate for operations where two people are involved in patient maintenance. However, it was also the opinion of all the raters that MOMA showed the potential of being a useful addition to the operating room as a multiparameter monitor.

A general comment expressed by all the rating anesthetists was that assessment of the clinical effectiveness of MOMA could only be determined through its use in the operating room. Each anesthetist expressed an interest in participating in such clinical trials. In addition, the following general comments were given:

1) The computational capabilities of the computer could be used for the determination of various derived parameters.
2) The inclusion of non-EEG parameters and the multiple display feature should significantly enhance the usefulness of MOMA, both inside and outside the operating room.
3) One EEG channel is sufficient for monitoring during open heart surgery and two channels are sufficient during carotid endarterectomies.
4) Frequencies from 16 to 32 Hz are not needed.
5) The extraction and enhancement of clinically significant information from the EEG power spectra might make the EEG easier to use in the operating room.
6) A description of clinically significant changes in the EEG would be helpful for initial training purposes.
7) Timed tags might be put on the non-EEG displays.
8) The EEG might be deemphasized in displays incorporating both EEG and non-EEG parameters.

5.2 Software Assessment

5.2.1 Procedure

An engineer employed at V.G.H. and experienced in both FORTRAN and RT-11 was asked to assess MOMA's software. As an introduction to the features and the intent of MOMA he was given the MOMA user's guide and the MOMA summary (see appendix 3). He was then given a similar system demonstration to the one given to the anesthetists. Finally, he was asked to overview the programs and related documentation and to complete a structured questionnaire (see appendix 4). Due to the size of the programs and the limited amount of time that he could be expected to donate, only the general flexibility and quality of the programs and documentation were assessed.

5.2.2 Results

The assessor felt that his assessment could most appropriately be made in a written manner and that he was unprepared to answer the questions in section two of the MOMA software general assessment form (appendix 4). He did however, answer yes to all the questions in section one and expressed the opinion that the software would be useful for the development of an effective patient monitoring system.

The assessor's impression of useability of the overall system was favorable. He felt that it would be easy to use and that its features were to a large extent self explanatory. He found the users guide "readable, clear and concise" and felt that it was written in a form that should be understandable by a non-technical person. Four suggestions for possible expansions and modifications of the system were made.

1) The user should not need to press "SHIFT" to change a display.
2) The capability of removing a timed tag and/or saved record from the display should be implemented.
3) A record keeping system should be implemented which allows the user to enter long comments before, during, and after the operation.
4) A real-time display of raw EEG should be available since an EEG machine will likely not be present in future systems.

He indicated that only the fourth suggestion would require major software revisions.

The software and related documentation was assessed as being of high quality. The software's modularity and structure, the fact that it is written entirely in a high level language, and the large number of comments within the programs were all mentioned as being strong points. The existing documentation was also found to be useful, especially the cross reference tables and the routine interconnection diagrams. It was suggested, however, that higher level documentation also be provided. More specifically, the following suggestions were made.

1) A description and/or table indicating which routines affect given common block parameters should be provided.
2) A more complete discussion of the use of various program flags should be provided.
3) A more complete description of the linking between the foreground job, the background job, the user, the operating system, and the hardware should be provided.
4) A crossreference list of the SYSLIB procedures called by various routines should be provided.
5) A discussion of intertask communication and routine calling conventions should be provided.
6) Diagrams of the data structures should be provided.

His closing remarks were "My criticisms of the documentation are not major and should not be allowed to overshadow the overall quality of the system. In summary, MOMA appears to be a good basis for an elaborate patient monitoring and information retrieval system of real use to physicians in the operating theatre.".

5.3 Testing for Electrical Safety

Both the terminal and the computer have been certified for electrical safety by the Canadian Standards Association (CSA) as data processing equipment (CSA standard C22.2 No. 154-1975). In order to insure compliance with the CSA requirements for electromedical equipment (CSA standard C22.2 No. 125-1979), the grounding and risk current levels of MOMA were tested. The results of these tests can be found in table 3. Also, in addition to existing markings, MOMA must be plainly marked in a permanent manner with the following information.

1) Danger-Explosion Hazard. Do not use in the presence of flammable anesthetics.
2) Conforms to risk class 1 requirements
3) Caution: Total system chassis risk current should not exceed 500 uA.

Table 3: Moma Safety Test Results

CATEGORY	RISK CLASS 1 CSA REQUIREMENT	MEASURED VALUE
GROUNDING RESISTANCE (gnd pin to chassis)	.1 ohm	.07 computer pin to computer chassis .12 computer pin to terminal chassis .06 terminal pin to terminal chassis
RISK CURRENT		
(grounded and off)	10 microamp	0 microamp
(" " on)	" "	0 "
(ungrounded and off)	500 "	16 "
(" " on)	" "	400 "
(" rev pol and off)	" "	22 "
(" " " " on)	" "	390 "
TOTAL CURRENT CONSUMPTION	------------	6.6 amps

In its present form, MOMA has no patient applied parts, thus it is classified as a risk class 1 device. Since the grounding resistance from the computer power cord's ground pin to the terminal's chassis slightly exceeded the specified limitation, it was recommended that an electrical connection be made between the terminal's chassis to the computer's chassis when the system is repackaged. This discrepancy was not considered to be a serious one so no immediate action was required. In addition, it was recognized that the backpanel connections should be better organized and that protection from spillage should be included. Finally, when EEG amplifiers are included on the MOMA package, the entire package will be reclassified as a risk class 2 device (noninvasive patient applied part). Since MOMA does not meet the risk current requirements for this new class (100 microamperes) an isolation transformer will have to be included at that time.

The current required by the computer/terminal combination is specified at about 10 amperes. This current was measured and found to be 6.6 amperes. Since the circuit breakers in VGH operating rooms are rated at at least 15 amperes, the device by itself poses no electrical overload problems. However, care will have to be taken to insure that this device is not connected in parallel with other high current devices.

6. CONCLUSIONS AND FUTURE DIRECTIONS

6.1 Conclusions

The current state of patient monitoring in the operating room was discussed and it was suggested that an improvement in the quality of anesthetic practice would result from the simplification of the use of monitoring parameters and equipment. It was also suggested that these improvements will not necessarily be achieved by the simple addition of information to the operating room, rather, it is the overall presentation of information which must be improved.

In a discussion of the potential of computer control of the patient, it was concluded that present day computers lack appropriate data acquisition abilities and are unable to properly adapt to unexpected situations. Thus, computers can be best applied to aiding, rather than supplanting, the anesthetist.

This thesis investigates the application of the data acquisition, processing, storage, and display capabilities of existing computer hardware to surgical patient monitoring. As a result of this work, the following application areas of computer systems have been identified:

1) The data acquisition and display capabilities of an appropriately designed computer-based monitoring system may be used in perioperative patient monitoring to:
   a) Display a number of parameters on a single display medium.
   b) Emphasize parameters in a way that eases data interpretation.
   c) Provide a number of predefined displays accessible through a single display medium so that the display which best suits a given user and/or a given situation can be chosen.

2) The signal processing capabilities of an appropriately designed computer-based monitoring system may be used in perioperative patient monitoring to:

   a) Relieve the anesthetist of some of the comparison and correlation tasks required for data interpretation.

   b) Provide artefact detection and rejection schemes.

   c) Preprocess certain parameters so as to make them more practical for the real-time assessment of a surgical patient's state.

   d) Provide a warning system which would warn the anesthetist about possible dangerous situations.

3) The information storage capability of an appropriately designed computer-based monitoring system may be used in perioperative patient monitoring to:

   a) Acquire information without displaying it, thus allowing the anesthetist to change the subset of relevant information used for patient state assessment at any time during the operation by simply changing the display.

   b) Permit the design of displays in which trends in parameters are readily apparent.

   c) Relieve the anesthetist of some of his record keeping duties and, in the process, produce a detailed and objective record of the operation.

   d) Create hospital records of consistent legibility, completeness, and format.

   e) Permit the anesthetist rapid access to large volumes of relevant preoperative information without the addition of extra books and paper records.

The following design considerations should contribute to the acceptance of computer-based monitors in the operating room:

1) An effort must be made to identify and compensate for possible sources of artefacts.
2) Considering the possible danger resulting from the anesthetist being temporarily unable to assess the patient's state, the device must be extremely reliable.
3) For obvious reasons, the monitor should pose little threat to either the patient's or the user's safety in the event of a system breakdown or improper use.
4) Use of the system should not require computer expertise and/or typing skills.
5) System setup and operating procedures should require a minimum amount of time and technical expertise.
6) If possible, all components of the system likely to be used during the operation should reside in the operating room, ie., no external electrical connections.
7) The system should be reasonably compact, noise free, and moveable.
8) Compatibility with presently used devices and likely future developments in other devices must be considered.
9) Flexible prototype systems should be built and clinically assessed so as to incorporate the opinions of experts in many fields in the final design. To keep up with technological advances, this design and assessment cycle should continue indefinitely.

With the above potentials and design considerations in mind, a surgical patient monitor prototyping system was developed. General purpose hardware was purchased and programs were developed for visual editing, computer graphics, data acquisition, data storage, user interaction, and prototype testing and calibration. The overall design direction was towards maximized flexibility and programs were heavily documented.

An initial prototype monitor was developed for clinical assessment. Although this prototype concentrated on the introduction of the EEG into the operating room, the system has the capability of simultaneously acquiring, processing and displaying non-EEG parameters. All of the application areas mentioned above except for record keeping and access to preoperative information were included on this system and space for the exploitation of these remaining two applications was left.

The general flexibility and quality of the programs and related documentation was assessed by a software engineer employed at VGH and experienced in FORTRAN and RT-11. It was his impression that the software written for this thesis was of high quality and flexibility.

The initial prototype system was assessed by a panel of five anesthetists practicing at VGH. Each anesthetist was asked to complete a structured questionnaire after reading a user's guide and after having the features of the system demonstrated by the author. The results of this assessment were quite favorable and each anesthetist expressed an interest in being involved in future clinical application of the device.

In conclusion, no definitive statement can be made about the success or failure of computer systems as patient monitors thus far. However, the system developed will be useful for the exploration of this potential. Through a process of continued prototype development with interspersed clinical assessment, it is anticipated that an effective computer-based patient monitoring system will be developed.

6.2 Comparison with Previous Systems

In chapter 3, some problems of previously available systems were presented. By taking advantage of progress in computer technology, it was possible to implement features on this system which previously were not feasible.

In systems where the computer was located outside the operating room, (eg. 20) the direction for development has been to increase the number of patients simultaneously monitored. For reasons described in chapter 2, it may be difficult to achieve the level of reliability, fault tolerance, and usability required for an intraoperative patient monitor using this approach. In addition, the resulting system will not be mobile. The direction for development chosen for the MOMA project was to produce a single-patient mobile monitoring system which can be located entirely within the operating room.

The computer used for MOMA has considerably more computational power and data storage capability than previous systems which resided within the operating room. Consequently, the number and sophistication of the features implemented on MOMA is larger. For example, the La Jolla multiparameter patient monitor (45 pp. 34-36) monitored one EEG channel and outputted the data from that channel onto a thermal printer in a fixed format. MOMA, on the other hand, monitors 4 EEG channels and display the data from these channels in a variety of formats.

In previously available systems, the format for information presentation must be selected prior to entry to the operating room and cannot be changed while in the operating room. MOMA, on the other hand, allows the user to change between a number of predefined displays. It has been suggested earlier in this thesis that this feature will give the user the luxury of selecting a display which he feels is most appropriate for a given situation and of suppressing the display of information not important at a given time.

The use of the EEG as a monitoring parameter has, in the past, been limited by difficulties in performing the long term trending necessary for patient state assessment. Two features of MOMA should help alleviate this problem. Firstly, it is possible to recall any previously displayed EEG display. This feature, called scrolling, allows the user to scan through all EEG information acquired earlier in the operation. While the ability to access previously acquired EEG records has appeared in earlier systems, the user has generally been limited to fixed time jumps. Secondly, the display for each EEG channel is split into two sections, a scrolled section and an unscrolled section. The information in one section is independent of the information in the other section. By "saving" given EEG records in the unscrolled section and moving the scrolled section to a different point in time, it is possible to perform direct comparisons between EEG records separated by long time periods. No other system could be found which used a two section display for EEG presentation.

Timed tagging facilities similar to the one implemented on MOMA have been incorporated on previous patient monitors (eg. 46). However, the computers used for these other systems were located outside the operating room.

Due to hardware restrictions and/or by choice, an emphasis has been placed on the reduction of program size in previous systems. Consequently, programming has been done in assembler language and small application-specific program-control executives have been used. However, this approach complicates the task of programming and generally produces a less flexible, less readable, and less portable program. For MOMA, it was decided that program flexibility and readability should not be compromised for the sake of program size, so a relatively high level programming language (FORTRAN) and a frequently used relatively sophisticated operating system (RT-11) were used.

6.3 Direction for Future Work

6.3.1 Hardware Modifications

A number of useful hardware additions and alterations have been identified. The following paragraphs describe these potential improvements.

It was found that the amount of random access memory presently in the computer (64 kilobytes) placed unnecessary restrictions on the MOMA programs. Using an overlay scheme, it was possible to leave parts of the program on disk but this noticeably slowed down the system and resulted in the generation of noise from the disk. An LSI-11/23 system upgrade kit has been purchased to overcome this problem. This kit will double the amount of memory and will increase the processing speed of the system.

It is the opinion of hospital personnel and the author that some form of hard copy output is required if a monitor such as MOMA is to see routine use in the operating room. By periodically outputting various parameters during the operation the anesthetist would have a backup source of data in the event of a system failure. In order to minimize the amount of paper generated by the system within the operating room and to minimize the time required to interpret the hard copy record, this record should only produce a brief summary of certain parameters and should probably not serve as a record of the operation. Thermal, electrostatic, and ink jet printer are most appropriate for this purpose because they operate quietly. The task of generating a detailed summary of the operation is most appropriately handled outside of the operating room. Using this approach, the printer would not need to be housed on MOMA's mobile cart and could, in fact, be shared by a number of computers. Size and noise considerations are not as critical and, since the printer would serve a number of computers, a more expensive and powerful unit could be justified.

In the existing system, a separate EEG machine must be used to acquire, amplify, and filter the EEG signals. In order to maximize mobility, minimize size, simplify setup procedures, and minimize the likelihood of faulty connection and/or accidental disconnection, appropriate EEG acquisition hardware should be incorporated into the MOMA package. Towards this end, Grass model P511-J EEG amplifiers/filters have been purchased to replace the EEG machine. It has been the experience of hospital personnel that these units are safe and reliable.

A color display medium would be useful for the development of aesthetically pleasing displays with quickly and easily identified features. A Tektronix 4027 color graphics terminal would provide a color display capability and is compatible with existing programs.

Through the process of continued device assessment and redesign, various other special purpose devices may be added to MOMA. For example, devices such as plastic card readers, writing pads, or special function keyboards may be incorporated in the system to simplify the entry of data. Filters, preprocessing units, and data acquisition hardware may be added to allow the system to reject artefact, to relieve the central computer of some of its processing tasks, and to gather more data.

6.3.2 Recommendations for Immediate Action

After the processor upgrade kit and the EEG amplifiers described in section 6.3.1 are acquired, the entire system should be repackaged. When this is done, things like spillage protection, large wheels, workspace and drawers, holders for miscellaneous equipment (eg. floppy disks or documentation), standardized connection panels with noninterchangeable connectors, expandability for future modifications, improved aesthetic appeal, accessability and usability of control switches, reduced size, increased mobility, and greater durability should be considered. In addition, an Isolation Transformer should be included so as to comply with the risk current regulations for a class 2 device.

When the repackaging is completed, a series of clinical trials should be performed so as to determine overall clinical usefulness and to obtain more detailed ideas from the anesthetists about useful changes and additions. A technical observer familiar with the features, use, and setup of MOMA should be included in these trials. This observer will serve to record the features and displays most frequently used, to record the comments and difficulties which the anesthetists encounter, and will be able to aid in the solution of any problems which interfere with the anesthetist's ability to treat the patient. Also, with a minor software alteration, the computer can record the frequency and time of entry of any of MOMA's commands.

6.3.3 Directions for Future Development

The design of safer, more effective, and/or more easily applied signal transducers should ease the task of system setup and increase the effectiveness of computer-based monitors. These transducers may make it possible to acquire a number of signals from a single transducer (eg. 64) and quantize signals previously inaccessible to the computer (eg. 65). There has been interest expressed at Vancouver General Hospital in the development of an easily applied EEG electrode headset. Since the application of EEG electrodes is often in the critical path of presurgical events, a development such as this should reduce the delay time between operations and/or reduce the number of support personnel required in and around the operating rooms. In addition to becoming a more effective patient monitoring system with the addition of better transducers, MOMA may be found useful for assessing transducers under development.

As a result of meetings with hospital personnel, four application areas were identified. Two of these areas (open heart surgery and carotid endarterectomies) involve the use of the system by anesthetists within the operating room. The other two areas (monitoring during advanced brain resuscitation and monitoring of head injury patients) involve the use of the system by diagnostic neurophysiologists, neurosurgeons, and/or nurses outside of the operating room. Research projects using MOMA could be built around any of these application areas. There are, in fact, plans to start a two year research project to identify the clinically valuable variables that might lead to improvement of the quality of cardiac surgery.

The second part of the questionnaire used for the MOMA system assessment contained questions intended to elicit ideas for the future direction of the MOMA project. The purpose of question 2.1 was to determine the parameters considered to be important for patient monitoring. As indicated in table 4, parameters related to the cardiovascular system were considered to be most important, followed by the respiratory system and the renal system. Since heart rate and rhythm are monitored using the ECG, they were combined into one category. The EEG was considered to be more important during carotid endarterectomies than during open heart surgery. Since EEG is not routinely

Table 4: The Relative Importance of Monitoring Parameters (as determined in question 2.1 of the assessment questionnaire)

QUESTION: List, in order of importance, six parameters which you feel are essential for patient monitoring during open heart surgery and carotid endarterectomies (two lists).

RESULT: A

RATINGS FOR OPEN HEART SURGERY
(listed in order of rated importance)

PARAMETER	ANESTHETIST				
	A	B	C	D	E
Blood Pressures	2,4,6	2,4,5	2,4,6	1,4,5	1,4
[ECG	1	1	1	2	---
Heart rate and Rhythm]	---	---	---	---	2
Ventilation	3	---	3	---	---
Urine Output	5	---	5	3	5
Laboratory Analysis Data	7,8	3	7,8	---	3
EEG	8	6	8	8	---
Temperature	---	7	---	---	8

RESULT: B

RATINGS FOR CAROTID ENDARTERECTOMIES
(listed in order of rated importance)

PARAMETER	ANESTHETIST				
	A	B	C	D	E
Blood Pressures	2,6	3	2,6	1,5	1
[ECG	1	2	1	2	---
Heart rate and Rhythm]	---	---	---	---	2,5
Ventilation	3	---	3	---	---
Urine Output	4	---	4	4	---
Laboratory Analysis Data	---	---	---	---	4
EEG	5	1	5	3	3
Temperature	---	4	---	---	---

monitored during open heart surgery at VGH, the frequency with which it was listed in this question may be partially due to its presence on MOMA. The anesthetists were asked to list parameters for which they felt trend information was not adequately presented. They were also asked to make a list of parameters that would be more useful if recorded more frequently. A summary of these lists can be found in table 5. The replies to the above questions indicated that the unallocated A/D channels in MOMA should be allocated primarily to the acquisition of data related to the cardiovascular system. In addition to various blood pressures, it was suggested that the detection and presentation of ischemic trends in the ECG warrants investigation. Provided that an appropriate transducer could be designed, the acquisition of the urine output rate was also suggested as being potentially useful. Considerable interest was expressed in the use of MOMA's computational capabilities for the determination of various derived parameters, such as those used for haemodynamic tracking. The computer in this case would automatically acquire the parameters which can be sampled and prompt the user for others which are required. Although similar systems for the interactive computation of derived parameters have been developed (eg. 66,67), the superior computational and graphical capabilities of MOMA should permit the implementation of a more powerful facility. As the intended uses of MOMA become more precisely defined, appropriate sampling and processing schemes can be inserted (eg. 68).

Warning systems and artefact detection and rejection systems have not yet been implemented with MOMA. As described in chapter 2, the inclusion of these systems on a computer-based monitor may improve the quality of health care by speeding the detection of, and perhaps even predicting the occurrence of dangerous situations.

Table 5: Suggested Improvements in the Handling of Parameters

(A)

QUESTION: List any parameters with which you feel that trend information is not adequately presented using existing instrumentation.

RESULTS:

Cardiovascular parameters	RAP LAP PCWP ECG
Derived parameters	Vascular resistance Cardiac output Cardiac index Haemodynamic tracking
Laboratory derived parameters	PaO_2 $PaCO_2$ pH
	EEG

IMPROVEMENTS BY MORE REQUENT SAMPLING

(B)

QUESTION: List any parameters which you feel would be more useful either during or after an operation if they were recorded more frequently.

RESULTS:

Cardiovascular parameters	MAP PCWP
Derived parameters	Cardiac output Cardiac index Haemodynamic tracking Rate pressure product (HR*SAP) Triple index (HR*SAP*PCWP)
Renal parameters	Urine output rate
Laboratory derived parameters	SvO_2
	EEG

The timed tagging facility allows the user to enter comments during the operation via the keyboard. However, since typing these comments will unnecessarily take attention away from the patient, this feature will probably meet with limited acceptance in its present form. In addition, paper records must still be maintained in order to insure the existence of a record of the operation in the event of a system failure. Thus, timed tagging constitutes a duplication of record keeping duties. To use a computer as a viable intraoperative record keeping device, some quick means of manually entering data to the computer other than typing is required. Although this may be achieved through the use of special function keyboards,(eg. 20) such a scheme limits the user to a fixed number of predefined comments. A possible alternative is the entry of comments through a writing tablet which is interfaced with the computer. The record of the operation would then be a combination of the record on the tablet and the record on the computer. No duplication of effort by the anesthetist would be required.

As described in chapter 2, access to relevant preoperative information stored in the computer may serve as a useful supplement to the anesthetist's memory. The graphics terminal used for MOMA (Tektronix 4025) is particularly well suited to this application. Its split screen feature and form fillout mode could be used to produce a standardized and simple format for data entry. These same features along with the character enhancement feature allows for the development of highly readable display formats. Implementing a preoperative information capability on MOMA is largely a software task. A support program will have to be written to permit convenient data entry and record modification. Program modules will have to be added to the background task of MOMA to read and display various subsets of this information. The advantages of computer-based preoperative records have been discussed elsewhere(19).

The applications of MOMA in the hospital extend far beyond these areas of immediate interest. The patient monitoring capabilities of MOMA could be used in many areas, from diagnostic neurophysiology (eg. 69,70) to the delivery room (eg. 71). In addition, the ability to sample, process, store, and graphically display a variety of information can be used in areas such as teaching (eg. 72) and text processing. The number and diversity of these applications are limited only by the imagination of the user.

6.3.4 Specific Modifications

Listed below are a number of changes in MOMA's displays which might enhance their usefulness:

1) The scrolled section of the EEG displays might be separated from the unscrolled section by a dividing line to improve upon the clarity of presentation.
2) The unscrolling section of the EEG displays might be automatically loaded with select EEG records.
3) The ability to change the size of the scrolled and unscrolled sections may be desired.
4) The ability to scroll the unscrolled section and/or freeze the scrolled section may be desired.
5) Displays might be produced which plot the non-EEG data beside the EEG data in the space presently occupied by the timed tags.
6) Control over the time increment between the records in the scrolled section of EEG displays may be desired.
7) Control over which EEG channels are displayed may be desired by the system user.

8) A display may be produced which incorporates both type 1 (EEG) and type 2 (time plotting) formats.

9) Control over the number of channels plotted in the time plots may be desired.

10) Control over which channels are plotted in the time plots may be desired.

11) The ability to interactively alter the display labels for the non-EEG data might be desirable.

12) An indication of when the display is in a scrolled state might be desirable.

13) A logarithmic time scale on the non-EEG plots might be useful.

The desirability of these alterations will be determined through consultation with clinicians and device assessment. It is significant to note that suggestions 3, 4, 6, 7, 9, and 10 can be implemented by altering a single program parameter.

Given the present hardware configuration and data, a number of additional features may be beneficially added. A "pause" key, which temporarily suspends all samples and processing, may be desired. Keys to jump backward or forward a large number of records (greater than 9) or to jump to specific points in the operation may be desired. Keys which repetitively increment or decrement a display may be desired. As before, inclusion of these changes will depend on the results of clinical assessment. The above suggestions far from exhaust the changes that might be made.

6.4 Device Maintenance

The need for safety and reliability of equipment used in hospitals has been mentioned a number of times already. If these needs are to be satisfied, any complicated device should periodically undergo a routine inspection. The risk current and grounding current of electromedical equipment at VGH is routinely tested and device control records are maintained. In addition to these procedures, the following procedures should be periodically performed on MOMA.

1) The A/D converter should be calibrated and tested (see ADCAL).
2) All wires and connectors should be checked to ensure proper condition and connection.
3) System diagnostics should be run so as to check various components of the computer.
4) The diskettes should be checked for bad blocks using the RT-11 DIR command and visually inspected for signs of unusual wear.
5) The air filters of the disk drive should be cleaned.
6) Backup copies of all important information stored on disk should frequently be made. It is preferable that these copies be stored in a different location than the originals.

APPENDIX 1: THE DISK CONFIGURATION

The disk system used in MOMA (RX02) accepts two 512512-byte floppy disks. As the system is presently configured, it will bootstrap off of the left hand disk drive (drive 0) immediately upon powerup or upon restart. The program used within the operating room uses both disk drives. The disk inserted in drive 0 will be called the master disk and the disk drive 1 will be called the slave disk.

A1.1 Master Disk Formatting

The master disk is used to supply the programs to the computer. Since it is the disk which is used for bootstrapping, it must contain a copy of the operating system (RT-11V03-B Foreground-Background), the bootstrap procedure, and any system utility programs which may be used while the system is running. In addition to the above programs, it will also contain separately executable versions of the MOMA programs. A typical directory listing for a MOMA master disk can be found at the end of this appendix.

A1.2 Slave Disk Formatting

The slave disk is used for the storage of data acquired during the operation. The files on the disk must be initialized prior to entry by the program DY1INI. No other files but the ones initialized by DY1INI need to be, or should be, on this disk.

For the storage format used, data is separated into a number of predefined files. There is a separate file for each EEG channel's power spectra and a fifth file for the non-EEG data and the MOMA cycle time. The

The time required to store and/or read EEG data could be reduced by placing all the EEG data in a single file. However, this alternative format was rejected because it complicates the task of adding or subtracting EEG channels, thus limiting system flexibility.

A1.2.1 Data Storage Requirements

For the purpose of maintaining a record of the operation and permitting the recall of this record during the operation, data is stored on every MOMA cycle on a 512512-byte floppy disk. The computed power spectra of four channels of EEG are stored along with raw data from the four remaining channels of the A/D converter and the time of the cycle. Additionally, a status file is updated on each cycle so that critical program parameters are not lost in the event of a power down. Since the storage of 4 channels of raw EEG data sampled at 64 Hz would use up the disk's storage capacity in less than 17 minutes (512 bytes per second), such storage is clearly not feasible. Instead, each EEG channel is allocated 230 blocks (117,760 bytes) of disk storage. Since 128 bytes per channel are stored each cycle (32 four-byte real numbers), there is a maximum of 920 cycles. Using a cycle period of 30 seconds, each disk has enough storage capacity for 460 minutes, or 7 hours and 40 minutes. An allocation of 29 blocks for the storage of non-EEG data and cycle times is sufficient to match the above time restriction. To summarize, the EEG data consumes 920 blocks (4 * 230), the non-EEG data and cycle times consume 29 blocks, and the critical program parameters consume one block. Since each disk has a maximum capacity of 974 blocks, there remain 24 blocks available for other uses.

A1.2.2 Timed Tagging Storage Requirements

The timed tagging facility consists of two files. The first file contains a list of the cycle numbers associated with the stored tags and is one block long. The second file contains the timed tags truncated to 32 characters and is 16 blocks long. Thus there remain seven unallocated data disk blocks. This arrangement allows for the entry of up to 255 timed tags.

A1.3 Proposed Use of Free Blocks

The free blocks on the master disk may be used for more support and program development programs, an expanded help facility, or anything else that may be useful inside or outside the operating room. It is possible to use these blocks for data storage. If this were done, however, the record of the operation would no longer be on one disk.

The remaining unused seven blocks on the data disk may be used for the storage of preoperative information relevant to the operation, e.g. patient history, unusual pathologies, premedications, etc.. A display incorporating such information would serve to reduce the number of paper records in the operating room and enable the Anesthetist to access more information about the patient without increasing the number of paper records. This facility has not as yet been implemented.

Table A1.1: Moma Slave Disk Configuration

FILE	# BLOCKS	PURPOSE
STATUS.DAT	1	Saves critical program parameters every MOMA cycle in case of power failure.
SLOCHN.DAT	29	Saves 4 channels of slowly sampled data and the time of acquisition every MOMA cycle.
POWER1.DAT POWER2.DAT POWER3.DAT POWER4.DAT	230 230 230 230	Each power file saves the computed power spectral estimate (32 two-word real numbers corresponding to frequencies from 0 to 31 Hz) from one channel every MOMA cycle.
TAGSNC.DAT	1	Contains a list of cycle numbers which are associated with timed tags previously entered.
TAGDAT.DAT	18	Contains the 32 character timed tags.
RECORD.DAT	7	Proposed file for a preoperative patient summary.

Note 1: 256 16-bit words per block
Note 2: 974 blocks per disk using the double-density (RX02) format.
Note 3: Standard IBM compatible single-sided soft-sectoral diskettes.

APPENDIX 2: PROGRAM DOCUMENTATION

A2.1 General Text Editing (TECHLP)

This document gives a general overview of the tools used for editing in this system. The editor used is TECO, supplied in the system distribution software for RT-11V03-B. This editor can also be obtained from the DECUS program library. TECO documentation can be found in RT-11 SOFTWARE PRODUCTS MANUAL #2.

DISTRIBUTED FILES

Also distributed with TECO are a number of *.TEC files. These files are TECO macro routines again written by DECUS members. View README.TXT for general information about TECO related files and documentation for the following modules, README.TXT,TEADME.TXT, TECO.OBJ, TIO.MAC, LOCAL.TEC, SORT.TEC, INSERT.TEC, EDIT.TEC, VT52.TEC, VEG.TEC.
Note that for a PDP-11 - TEKTRONIX 4025 combination, INSERT.TEC and VT52.TEC will not function due to cursor control problems. For INSERT.TEC to work, the ^W command must be adapted to this terminal. For VT52.TEC to work, TECO's cursor control characters used in the -1^W command must be adapted to this terminal. It is suggested that you use the VIS*.TEC editing package for visual editing. This package takes advantage of both the built in visual editing capability of the 4025 and the general editing capability of TECO. See the next section for a more complete description.

LOCALLY WRITTEN MACROS

An attempt to write a full visual editor for the Tektronix 4025 using TECO instructions was made (TXVIS). The resulting routine was found to be too slow and prone to error. However, some of the routines, especially the subroutines, may be useful. See the file TXVIS.TXT for a listing.

A second attempt at a TECO-based pseudovisual editor was made. (TX4025.*). Instead of full visual editing, this macro performed specific single key instructions and typed out the result. Although TX4025.* works at a usable rate, it is prone to user error. This package is comprised of three programs:

1) TX4025.TEC Running version
2) TX4025.TXT Documented version
3) TX4025.KEY Key programming

Both TYVIS.* and TX4025.* are loosely related to INSERT.TEC.

As was mentioned earlier, the VIS*.TEC programs make up a visual editing package which interfaces the visual editing capabilities of the 4025 terminal with TECO. VISXXX.DOC gives a detailed description of these programs and their use. In the authors opinion, this package is the most useful of the packages mentioned in this section.

COMMAND CHARACTERS

The terminal's command character for the majority of the programs in this system is backquote (96). In many of the editing macros, the command character must be backquote for them to work. Consequently, unless there is a specific reason, the command character should be backquote while working with this system.

EDITING FILE SUMMARY

File	Function
TECHLP.DOC	-this file
README.TXT	-documentation of the following files
TECO.SAV	-runable copy of RT-11 TECO

File	Function
TECO.OBJ	-central module of TECO for version 28
TIO.MAC	-source of RT-11 I/0 interface for TECO
LOCAL.TEC	-macro to renumber local symbol blocks in MACRO programs
SORT.TEC	-macro for general purpose memory resident sorting
INSERT.TEC	-macro for visual editing with VT-11 terminals
EDIT.TEC	-VT-11 start up macro
VT52.TEC	-macro for visual editing with VT-52 terminals
VEG.TEC	-VT-52 start up macro
TX4025.DOC	-documentation for the following files
TX4025.TEC	-running version of a pseudovisual editor for the Tektronix 4025 graphics terminal
TX4025.KEY	-Key reprogramming instructions used by TX4025
TX4025.TXT	-commented version of TX4025
VISXX.DOC	-documentation for the following files
VISSTA.TEC	-package acquisition macro on TECO entry
VISFIN.TEC	-performs functions useful when exiting TECO
VISMAC.TEC	-running version of an interface between TECO and the Tektronix 4025 graphics terminal
VISCND.TEC	-communications command character changing macro
VISPRO.TEC	-prompt character changing macro
TECVIS.TXT	-code for an abandoned visual editor for the 4025
KEYPGM.TXT	-Key programming part of the system start up routine

Note: For visual editing it is suggested that the VIS.TEC package be used.

A2.2 Visual Editing

TEKTRONIX 4025-TECO EDITING INTERFACE DOCUMENTATION

DESCRIPTION

These TECO macros are designed to interface the editing capabilities of the 4025 graphics terminal to TECO. The necessary package acquisition and execution instructions are summarized later in this document.

VISMAC

VISMAC.TEC is the main editing interface macro. Its purpose is to read a specified portion of TECO's text buffer to the workspace region of the terminal and then to wait for edited text to be sent back. Editing is performed independent of the computer using the terminal's text editing capabilities.

The macro is entered using the instruction nMq$$ where n is some number and q is the name of the Q register which contains VISMAC (usually register M). When this instruction is entered, n lines each side of the current buffer pointer are typed into the workspace. These lines are also saved in Q register 1 and deleted from the TECO's text buffer. Additionally, the buffer is searched for the existance of the communications command and prompt characters and the macro will not be entered if they are found. The function of these characters is explained more fully later in this document. Finally, the "ERASE", communications command character, and prompt character keys are programmed to null and the "PT", "RUBOUT", "SEND", and "SHIFT/INSERT MODE" keys are programmed to perform their intended functions. Although the user can specify the argument n to be any integer number, VISMAC will bound it between 0 and 20.

The "SEND" key will cause the contents of the workspace to be sent to the computer. VISMAC will echo the characters as they are received and insert them in TECO's text buffer. The writing of "^E" into the workspace indicates that the insertion process has been completed.

The "PT" key will cause the old window to be read from Q register 1 back into the text buffer. Thus it is always possible to restore the buffer to its original status. This key actually sends the "CTRL/C" character.

In the process of terminating VISMAC with either of the above two instructions, all of VISMAC's special function keys except PT are set to their default values. The system will be returned to the standard TECO editing mode and the user will be prompted with a "*" in the terminal's monitor scroll when instructions can be entered. At this point, the text buffer pointer will be located at the end of the edited window.

The character string "10L15MM$$" is programmed into the "PT" key, on VISMAC exit. Consequently, the next time this key is typed, the text buffer pointer will be moved 10 lines down and VISMAC will be re-entered. Thus the "PT" key can be used to both enter and exit VISMAC and will be useful for "paging" through long text files.

The "SHIFT/INSERT MODE" key can be used while editing to replace the contents of the workspace with the contents of Q register 1. Thus it is possible to return the workspace to its status on VISMAC entry. This key actually sends the "CTRL/R" character.

A number of things must be avoided if the editor is to function properly. Firstly, during VISMAC entry and exit, characters must NOT be typed. Secondly, while in the visual editor, the user should only type into the terminal's workspace region, i.e., the only allowable interaction with the computer is the three special function keys described above. Thirdly, the communications command character and the prompt character must not be inserted into the workspace. Fourthly, due to the finite memory capacity of the terminal, no more than about 40 lines of text should be sent from (or read into) the workspace scroll at one time.

The following types of text will be automatically altered by using the "SEND" key to terminate VISMAC. Control characters will be converted to their caret-character equivalents. Tab characters will be converted to a series of spaces. Form feeds will be converted to their carriage-return/line-feed equivalents. Long lines will be shortened to 80 character segments. Finally, blank lines at the end of a window will not be sent back to computer.

VISCND AND COMMAND CHARACTERS

Since the computer automatically echoes characters as they are sent and the terminal must be receptive to commands at the same time, the presence of command characters in the text may produce undesireable results. As a solution to this potential problem, the normal, or interaction, command character is temporarily changed by VISMAC prior to any computer-terminal communication. The command character is changed back when the communication is complete. The new command character has been labeled the communication command character (C.C.C.).

It has been stated previously that VISMAC cannot be entered if the C.C.C. is present in the text buffer. However, through the use of the VISCND macro, it is possible to change the C.C.C. to any ASCII character. This macro is called using the instruction nMq where n is the ASCII decimal. Equivalent (ADE) of the new C.C.C. and q is the Q register containing VISCND (usually register C). If the new C.C.C. is; present in the text buffer, the same as the interaction command character (backquote), the same as the prompt character, or is a control character, a warning is sent to the user and no action is taken. If the above restrictions are satisfied, the C.C.C. is changed by directly altering the q register containing VISMAC.

The terminals command character will be backquote before, during, and after calling a window to the workspace. This character has been labeled the interaction command character in this text. The interaction command character must be backquote (shift/backslash) for VISMAC to function properly.

VISPRO AND BUFFERED MODE

In order to avoid TECO linebuffer overflow, VISMAC puts the terminal in the buffered mode prior to computer-terminal communication. Consequently, as with the C.C.C., the presence of the prompt character in a text window sent to VISMAC may cause problems. Using the VISPRO macro, it is possible to change the prompt character (P.C.) to any ASCII character. This macro is called using the instruction nMq where n is the ADE for the new PC and q is the Q register containing VISPRO (usually register P). The new PC cannot be; present in the test buffer, the same as the interaction command character (backquote), the same as the C.C.C., or a control character. The P.C. is changed by directly altering the q register containing VISMAC.

VISSTA

The VISSTA macro sets up TECO for the use of the VIS* editing package. The following character string will enter TECO from RT-11 KMON and execute VISSTA.

```
R TECO

@ER%VISSTA.TEC%YHXSHKMS$$
```

KEYPGM.TXT programs the F4 key with this string. When executed, VISSTA sets TECO for scope and lower case, loads QF with VISFIN, loads QK with KEYPGM.TXT, loads QM with VISMAC, loads QC with VISCND, loads QP with VISPRO, and programs the "PT" with "10L15MM$$".

VISFIN

The VISFIN macro performs functions that are useful when exiting from TECO. Assuming the VISFIN has been loaded into QF, the following character string will execute VISFIN and exit from TECO.

MF^C$$

KEYPGM.TXT programs the S4 key (SHIFT/F4) with this string. When executed, the pad terminator will be deprogrammed.

C.C.C. AND P.C. IDENTIFICATION

While in TECO's normal editing mode, the ADE of the current C.C.C. can be found in Q register C(num) using the instruction QC$$.

On entry to VISMAC, a message is typed into the terminal's monitor scroll which identifies both the C.C.C. and the P.C.. Note that a 30 line workspace scroll will leave the user with plenty of editing area and will also leave room for the display of this message.

COMMAND LOCKOUT

When the terminal is put into command lockout, a command character will be interpreted as a standard character. VISCND and VISPRO will function properly in this mode. However, it is not possible to enter or exit VISMAC while in this mode. Procedures for recovering from attempted entry or exit of VISMAC while in command lockout will be presented later in this text.

ERROR RECOVERY PROCEDURES

STANDARD VISMAC CRASH RECOVERY TOOLS

TECO

WHILE STILL IN VISMAC

^C	Attempts to simulate the PT termination of VISMAC.
^E<RETURN>	Terminates VISMAC character insertion (e.g. at end of send).
^C^C	Hard macro (or TECO) exit. (use with caution)

AFTER GETTING OUT OF VISMAC

^G^G$$	Deletes last TECO instruction character string.
G1$$	Inserts the contents of Q register 1 into TECO text buffer.
:GK$$	Types contents of Q register K to terminal. (Q register K can be loaded with system start up) (Key programming file KEYPGM.TXT on TECO entry)

TERMINAL (do not affect TECO)

SHIFT/STATUS-COMMAND LOCKOUT	Identifies command character.
'SYS<RETURN>	Types terminal status message.
'BUF N<RETURN>	Removes terminal from buffered mode.
'MON H K<RETURN>	Directs keyboard (K) and computer (H) to monitor.
!COM '<RETURN>	Changes command character from ! to '.
SHIFT/LEARN-NUMERIC LOCK	Locally reprograms keys.
MASTER RESET (on side of terminal)	Resets all terminal modes, Clears all terminal memory, Clears all key programmings.

SOLUTIONS TO SPECIFIC VISMAC PROBLEMS

PROBLEM	SYMPTOM	SOLUTION
ATTEMPTED VISMAC ENTRY WHILE IN COMMAND LOCKOUT OR WHILE COMMAND CHARACTER IS NOT BACKQUOTE	- Text window and terminal programming commands all typed into the monitor scroll	1) type "^E<RETURN>" 2) type "^G^G$$ 3) type "G1$$" 4) rectify problem 5) check text buffer for erroneous insertions
ATTEMPTED VISMAC SEND WHILE IN COMMAND LOCKOUT OR WHILE COMMAND CHARACTER IS NOT BACKQUOTE	- prompt characters and typed characters written below text window in workspace	1) check the command character 2) move cursor to monitor 3) type "^E<RETURN>" 4) type "^G^G$$" 5) type "G1$$" 6) check text buffer for erroneous insertions
TYPING ON VISMAC ENTRY	- prompt characters and typed characters written below text window in workspace	1) check the command character 2) move cursor to monitor 3) type "^E<RETURN>" 4) type "^G^G$$" 5) type "G1$$" 6) check text buffer for erroneous insertions
TYPING ON VISMAC SEND	- characters typed before TECO prompt character "*"	1) type "^G^G$$" 2) check text buffer for erroneous insertions
SENT TEXT WINDOW TOO LARGE	- characters are not echoes normally send key was typed	1) wait until a large number of garbage characters are typed into the workspace 2) press the MASTER RESET button 3) type "^E<RETURN>" 4) type "^G^G$$" 5) type ":GK$$" (assuming KEYPGM.TXT has been loaded into Q register K) 6) check text to see what was lost Note: the old window is still in Q register 1.

VISXXX EDITING PROGRAMS

VISSTA.TEC = package acquisition routine

found in Q register S

call with MS

VISFIN.TEC = TECO exit routine

found in register F after VISSTA execution

call with MF

KEYPGM.TXT = standard system start up key programming file

found in Q register K after VISSTA execution

type out with :GK

VISMAC.TEC = Main interface program

found in Q register M after VISSTA execution

call with nMM

n = number of lines each side of the buffer pointer

VISCND.TEC = Changes VISMAC's communication command character

found in Q register C after VISSTA execution

call with nMC

n = ASCII decimal equivalent of new C.C.C.

VISPRO.TEC = Changes VISMAC's prompt character

found in Q register P after VISSTA execution

call with nMP

n = ASCII decimal equivalent of new P.C.

VISMAC SPECIAL FUNCTION KEYS

SH/INSERT	(S13)	Replaces workspace's contents with the old window.
SEND	(F8)	Replaces the old window with the workspace's contents and exits VISMAC.
PT		Puts the old window back in the text buffer and exits VISMAC.
ERASE		Disabled on VISMAC entry (re-enabled exit).
RUBOUT		Programmed for intended function.
C.C.C. Key		Disabled on VISMAC entry (re-enabled on exit).

P.C. Key Disabled on VISMAC entry (re-enabled on exit)

*Note: all other user interaction should be independent of the computer

PT On macro exit the PT key is programmed to generate 10L15MM$$. Consequently, this key can be used to both enter the macro and exit the macro. This is useful for scanning through a long text buffer.

TECO ENTRY AND EXIT

ENTRY (programmed into F4 by KEYPGM.TXT)

```
R TECO
@ER%VISSTA.TEC%YHXSHKMS$$
```

EXIT (programmed into S4 by KEYPGM.TXT)

```
MF^C$$
```

INITIAL PARAMETER SETTINGS

Interaction command character = backquote (96)

cannot be changed

Communication command character = vertical bar (124)

Alterable with VISCND to any noncontrol character

Prompt = backslash (92)

Alterable with VISPRO to any noncontrol character

*Note 1): the above three must always be different

*Note 2): this string is usually stored in a Q register for easy package acquisition, i.e.; Q register S

*Note 3): ASCII equiv. of C.C.C. found in QC (num)

*Note 4): ASCII equiv. of prompt found in QP (num)

Q REGISTER USAGE

VISSTA stored in QS(text) by the F4 key

QF(text) = storage for VISFIN

QK(text) = storage for KEYPGM,TXT

QM(text) = storage for VISMAC

QC(text) = storage for VISCND

QP(text) = storage for VISPRO

QC(num) = storage for the C.C.C.

QP(num) = storage for the P.C.

VISFIN stored in QF(text) by VISSTA

VISMAC stored in QM(text) by VISSTA

Q1(text) = old window storage

QL(text) = temporary buffer save

QL(num) = temporary buffer pointer save

QV(num) = number of lines each side of pointer in window

QW(num) = top of window location

QX(num) = bottom of window location

VISCND stored in QC(text) by VISSTA

QC(num) = ADE of current communication command character

QD(num) = input buffer for new C.C.C.

QL(num) = text buffer pointer storage

QL(text) = text buffer storage

VISPRO stored in QP(text) by VISSTA

QL(text) = text buffer storage

QL(num) = text buffer pointer storage

QP(num) = ADE of current prompt

QQ(num) = input buffer for new prompt

*Note 1): text registers C. M, and P must remain intact while in TECO

*Note 2): Q1(num) is useful in the event of VISMAC crash

*Note 3): previous contents are destroyed on Q register use

A2.3 Documentation Directory

```
 @DOCTYP
 Files copied:
DK:DOCHLP.TXT  to TT:
                       MOMA PROGRAM DOCUMENTATION DESCRIPTION
                       --------------------------------------

        The following is a description of the files written for
documenting the MOMA programs.

RT-11 FILE             PURPOSE
----------             -------

COM*.TXT        describes MOMA command files
SRT*.TXT        describes MOMA startup files
FOR*.TXT        describes MOMA foreground (data acquisition) programs
BAK*.TXT        describes MOMA background (display) programs
GPH*.TXT        describes GPHLIB graphics package

*EXP.TXT        describes the general approach for the given package
*RTN.TXT        describes the routines of a given package
*CMN.TXT        describes the common blocks of a given package
*CAL.TXT        describes the calling and called routines within a
                given package (not including SYSLIB calls).
*TST.TXT        describes the testing routines for a given package

eg; FORCMN describes the common blocks of the MOMA foreground routines

TECHLP.DOC      describes *.TEC programs and documentation
VISXXX.DOC      describes VIS*.TEC visual editing interface

        Routine connection diagrams were drawn from the information
in *CAL.TXT (FORCAL, BAKCAL, GPHCAL).  These diagrams were then used to
portray the flow of program control in the MOMA foreground and background
programs.  See FLOW.TXT for a description of these diagrams.
```

A2.4 Command Files Documentation

Files copied:
DK:COMRTN.TXT to TT:

COMMAND FILE DESCRIPTIONS

COMPILING COMMAND FILES

- all files are executed using the instruction "@filename<RETURN>".
- all files are named *.COM unless otherwize specified.
- all files produce object modules on DY1: from source modules on the default disk.

CPLGPH = Compiles all the graphics routines in the GPHLIB.OBJ library.

CPLMMB = Compiles all the routines used in DISPLA (the MOMA display and user interaction program).

CPLMMD = Compiles all routines associated with the highly interactive background version of AQUIRE.

CPLMMF = Compiles all routines associated with the foreground version of AQUIRE (the MOMA acquisition, processing, and storage program).

CPLMMG = Compiles all graphics routines used in the MOMA display and user interaction program DISPLA. The GPHLIB library could not be used in the linking process because the graphics routines appear in overlay regions.

LINKING COMMAND FILES

- All files named LNKMP* are executed using "@filename<RETURN>".
- All other files are executed using "LINK@filename<RETURN>".
  - for a link map typeout, use "LINK/MAP:TT:@filemane<RETURN>".
  - for a link map file, replace TT: with the file specification.
- All files are named *.COM unless otherwize specified.
- All files produce executable module(s) on DY1: from object module(s) on the default disk.

ADCAL = Links the A/D callibration routine. Produces DY1:ADCAL.SAV.

AQUIRE = Links the foreground version of AQUIRE (the MOMA acquisition, processing and storage program). Produces DY1:AQUIRE.REL.

AQUSAV = Links the highly interactive background version of AQUIRE. Produces DY1:AQUIRE.SAV.

BAKSIM = Links the background simulation routine. Produces DY1:BAKSIM.SAV.

DISPLA = Links DISPLA (the MOMA display and user interaction program). Produces DY1:DISPLA.SAV.

DSKCHK = Links the slave disk checking routine. Produces DY1:DSKCHK.SAV.

DY1INI = Links the slave disk initialization routine.
Produces DY1:DY1INI.SAV.

FORSIM = Links the foreground simulation routine.
Produces DY1:FORSIM.REL.

LINKBG = Alternate overlay scheme for DISPLA.
Produces DISPLA.SAV.

LNKBG2 = Alternate overlay scheme for DISPLA.
Produces DISPLA.SAV.

LNKMPB = Separately links each routine related to DISPLA and types out the link maps on the terminal. A number of nonexecutable .SAV modules are produced on DY1:.

LNKMPD = Separately links each routine related to the highly interactive background version of AQUIRE and types out the link maps on the terminal. A number of nonexecutable .SAV modules are produced on DY1:.

LNKMPF = Separately links each routine relate to the foreground version of AQUIRE and types out the link maps on the terminal. A number of nonexecutable .SAV modules are produced on DY1:.

LNKMPG = Separately links each routine in GPHLIB.OBJ. A number of nonexecutable .SAV modules are produced on DY1:.

OPTION = Links the MOMA startup routine OPTION.
Produces OPTION.SAV.

PAUSE = Links the MOMA startup routine PAUSE.
Produces PAUSE.SAV.

STARTUP COMMAND FILES

- All files are executed using "@filename<RETURN>".
- All files are named *.COM unless otherwize specified.
- See "MOMA STARTUP ROUTINES DESCRIPTIONS" in SRTRTN.TXT for a description of files related to STRTRT and MOMSET.
- Depending on the intended use of a given disk, STRTRT or MOMSET may be renamed to STARTF. If not, STARTF will call either STRTRT or MOMSET.

STARTF = Standard system startup file.

STRTRT = Startup routine with no queries. Puts the system in a configuration which has been found useful for general RT-11 use and then goes directly to KMON.

MOMSET = Startup routine with queries. Asks the user to check certain things in order to insure proper startup and allows the user to choose between 6 startup routes. Five of these routes lead to the execution of MOMA related routines and the sixth goes directly to KMON.

MOMSHT = Startup routine with no queries. Runs AQUIRE/DISPLA.

See "MOMA STARTUP ROUTINES DESCRIPTIONS" in SRTRTN.TXT for a description of the following command files;
CHOOSE.COM, MOMABB.COM, MOMABG.COM, MOMABT.COM
MOMAFB.COM, MOMAFT.COM, MOMSET.COM, MOMSHT.COM
NOMOMA.COM, STARTF.COM, STRTRT.COM.

A2.5 Startup Routines Documentation

Files copied:
DK:SRTEXP.TXT to TT:

MOMA STARTUP GENERAL DESCRIPTION

The command file which is automatically run whenever the system is booted is STARTF.COM. In its present form, this routine simply calls another command file to perform the system initialization procedure. Using a two level approach such as this, it is possible to choose between a number of system startup procedures with minor alterations to STARTF.

STRTRT.COM is the standard RT-11 startup file. It sets up the system in a configuration which has been found to be useful for general RT-11 usease and goes directly to KMON. More specifically, a number of RT-11 mode flags are set, the terminal's modes are initialized (TRMSET), the terminal's keys are programmed (KEYPGM) and the date and time are read from the TCU-50 battery operated clock. no queries or messages are sent to the terminal during this process.

The general MOMA startup routine is MOMSET.COM. This routine sets a number of RT-11 mode flags, types a message to the terminal (CMNDCK), and waits for the user to respond. The purpose of this message is to ask the user to insure that the command-lockout and the numeric-lock keys are off and the required responce from the user is "RETURN". After the user has responded, modes of the terminal are set (TRMSET) and the time and date are read from the battery operated clock (SETDAT). Finally, a program which allows the user to choose between a number of MOMA startup modes is entered (OPTION).

The OPTION program asks the user if he(she) wants to view the available startup options. If he(she) does, the CHOOSE.COM command file is entered. If not, it is assumed that MOMA is to be run in the normal operating room mode (AQUIRE/DISPLA) and MOMAFB.COM is entered.

The CHOOSE package is composed of a text file and a program. The text file (CHOOSE.TXT) describes a number of startup options available and the program (CHOOSE.MOM) accepts the users choice. There are presently six startup options available through CHOOSE.

1) AQUIRE alone in the background (MOMABB.COM).
2) FORSIM in foreground, DISPLA in background (MOMABT.COM).
3) DISPLA alone in the background (MOMABG.COM).
4) AQUIRE in foreground, BAKSIM in background (MOMAFT.COM).
5) go directly to RT-11's KMON.
6) AQUIRE in foreground, DISPLA in background (MOMAFB.COM).

All the above startup options follow a standard format. A message (GREET) asking the user to insure that appropriate connections have been made and that the data (slave) disk has been inserted is typed onto the terminal. The procedure waits at this point until the user types "RETURN". Once the user has responded, the slave disk is checked (DSKCHK). If it is empty, the initialization program (DY1INI) is entered. If not, no action will be taken. At this point, the requested MOMA programs are run. After the MOMA programs are terminated, the terminal's modes are reset (TRMSET), the keyboard is programmed to a configuration which has been found useful for general RT-11 use, and the command file is exited.

There are two deviations from this format. Firstly, for obvious reasons, GREET is not typed out when NOMOMA.COM is run. Secondly, in routines involving the use of the foreground (MOMAFT, MOMAFB), the user is asked to type a key to stop and unload the foreground job (FGOUT). This was necessary because no way could be found to terminate and unload the foreground job from a background command file.

DK:SRTRTN.TXT to TT:

MOMA STARTUP ROUTINES DESCRIPTIONS

COMMAND FILES (file names = *.COM unless specified otherwise)

```
CHOOSE = Types out CHOOSE.TXT and then runs CHOOSE.MOM.

MOMABB = Runs AQUIRE.SAV.  Note that this version of AQUIRE should have been
         compiled with the /DEBUG option and requires a considerable amount
         of user interaction.  Called from MOMSET.

MOMABG = Runs DISPLA without a foreground job.  Useful for reviewing old data.
         Called from MOMSET.

MOMABT = Runs DISPLA in the background and FORSIM in the foreground.  Useful
         for testing DISPLA.  Called from MOMSET.

MOMAFB = Runs DISPLA in the background and AQUIRE in the foreground.  This
         is the configuration used in the operating room.  Called from MOMSET.

MOMAFT = Runs BAKSIM in the background and AQUIRE in the foreground.  Useful
         for testing AQUIRE in the foreground.  Called from MOMSET.

MOMSET = General MOMA startup file.  Allows the user to choose between
         the MOMA** or the NOMOMA startup configurations.

MOMSHT = Short MOMA startup file.  Runs AQUIRE/DISPLA with no queries.

NOMOMA = Programs the terminal and goes directly to RT-11 KMON.
         Called from MOMSET.

STARTF = System startup file.  Automatically called when the system is booted.
         Depending on the intended use of the disk in question, MOMSET or
         STRTRT may be renamed to STARTF.  Otherwize, all three files will
         be on the disk and STARTF will call one of the others.

STRTRT = General startup file.  Sends no queries and goes to RT-11 KMON.
```

TEXT FILES (file names = *.TXT unless specified otherwize)

```
CHOOSE = Creates its own 30 line workspace and fills it with information
         about the startup choices available when CHOOSE.MOM is run.

CMNDCK = Creates its own 30 line workspace and reminds the user to check
         that the command lockout and numeric lock keys are are off.
         A procedure for reseting the command character is also typed out.

FGOUT  = Asks the user to type the pad terminator and programs it to
         unload the foreground job.  When typed, PT will deprogram itself.

GREET  = Creates its own 30 line workspace, introduces the user to MOMA,
         and asks if the backpanel connections have been made and the
         slave disk inserted.

KEYPGM = General purpose terminal key programming file.  Programs the keys
         in a configuration which has been found useful for general use of
         RT-11.

TRMSET = Initializes the terminal's modes in a format which has been found
         useful for general use of RT-11.
```

```
PROGRAMS  (file names = *.MOM unless otherwize specified)
--------  (sourse language = fortran unless otherwize specified)

CHOOSE = Asks the user to choose between the startup configurations described
         in CHOOSE.TXT and then calls the appropriate command file.

DSKCHK = Checks if the disk in DY1: is blank.  If it is blank, it chains to
         DY1INI.MOM.  If not, it simply returns.

DY1INI = MOMA slave disk initialization.  Initializes the disk in DY1: with
         all the data files used in MOMA.

OPTION = Asks the user if he(she) wants to view the CHOOSE.* options.  If
         he(she) does, the CHOOSE.COM command file is run.  If not,
         the MOMAFB.COM command file is run, thus going directly to
         the configuration used in the operating room.

PAUSE  = Simply asks the user to type "RETURN" and waits intil he(she) does.

SETDAT.MAC = A MACRO-11 routine supplied with the Digital Pathways TCU-50
             computer clock to read the time and date from the TCU-50 to
             the RT-11 time and date words.
```

A2.6 Foreground Routines Documentation

Files copied:
DK:FOREXP.TXT to TT:

MOMA FOREGROUND GENERAL DESCRIPTION

note; refer to the "MOMA FOREGROUND ROUTINE INTERCONNECTIONS" diagrams while reading the following descriptions.

COMPILED WITHOUT THE /DEBUG OPTION

All foreground activity is initiated from AQUIRE. On startup, AQUIRE calls FILACT to initialize the disk I/O channels and then waits until the background tells it to proceed. When this start message is received, AQUIRE enters the sampling, processing, and storage loop, called a "MOMA CYCLE". Note that the start message sets the value of LOPCNT.

At the start of a MOMA cycle, the time is saved in TIME and various pointers and counters are initialized. The SLOSMP routine is then called to sample NUMSLO non-EEG channels to the array ISLDAT. This array will not be altered until after the data is stored on disk and sent to the background job.

When AQUIRE has completed sampling the non-EEG data, it proceeds to set up the programmable real-time clock (KWV11-A) and the A/D converter (ADV11-A) with the SETR and the RTS routines. The Sampling frequency is set to 64 Hz and RTS is set to split the sampling buffer (INPUT) into 2 subbuffers for each of 4 EEG data channels. Each subbuffer contains 64 elements. Once these settings have been made, AQUIRE relinquishes control to the background job until RTS indicates that a subbuffer has been filled.

When a subbuffer is filled, RTS causes the routine RTSCOM to be entered. This routine takes the data in INPUT and places it in the data buffer ICHBUF. The 64 data samples per channel which are copied from INBUF to ICHBUF will from now on be called data segments. RTSCOM calculates the averages of the data segments and saves them in CHNAV. It then calls PREPAR to remove these averages from the segments and to window the segments (PREPAR calls WINDOW to perform the windowing). The above process is repeated until 8 segments have been acquired. At that point the clock and the A/D converter are turned off and the processing phase of the MOMA cycle is entered. Note that since RTSCOM's preprocessing is done in parallel with the sampling, it must be completed before the next subbuffer is filled.

Once the 8 segments of EEG data have been acquired and preprocessed, PWRSPC is called by AQUIRE to compute a power spectral estimate of the data. The technique used is a method of averaging the periodograms from a number of short data segments and was described by P. D. WELCH in 1967 (IEEE Trans. Audio and Electroacoust., AU-15, pp. 70-73, June 1967). As implemented here, the periodograms of 8 nonoverlapping preprocessed (in PREPAR) segments of 64 samples are computed for each data channel using FFT and POWRSP. The average of these periodograms is then computed and stored in the array POWER. Finally, in order to improve the statistical properties of the power spectral estimates, a three point moving average smoothing window is used (SMOOTH).

The data acquisition and processing is now complete. POWER contains the power spectral estimate of the 4 EEG channels, ISLDAT contains the data from the non-EEG channels, and TIME contains the time at which this data is sampled. This data is now stored on disk along with the cycle counter (LOPCNT) by FSTORE and sent to the background job from AQUIRE. Finally, the AQUIRE relinquishes control to the background job until it is time to run through another MOMA cycle. The AQUIRE parameter IDELAY sets the cycle repetition rate. (in clock ticks per cycle == 60 * seconds per cycle).

COMPILED WITH THE /DEBUG OPTION

All the acquisition, processing, and storage described above is the same when the programs are compiled with the /DEBUG option in effect. However, this version is designed to run in the background and is highly interactive. During each MOMA cycle, the user will be asked if he(she) wants various things typed out on the terminal. These things include a plot of the sampled EEG data (DATPLT) and a plot of the power spectral estimate (AQUTST). Finally, at the end of each cycle, the user is given the opertunity to terminate the program.

A character will be typed to the terminal when the program passes certain points in the acquisition and processing parts of the routine. The time at which these points are passed is also saved within the program. Using this feature, the user can trace the flow of program control and get an idea about the amount time being spent in the various modules of the system.

DK:FORRTN.TXT to TT:

MOMA FOREGROUND ROUTINES DISCRIPTIONS

NOTE; FFT, IADC, POWRSP, RTS, and SETR are found in the object library LABLIB.OBJ.

NOTE; When the routines AQUIRE, PREPAR, PWRSPC, and RTSCOM are compiled using the /DEBUG option. the resulting program will be in a highly interactive format suitable only for running as a stand-alone background job. When this compiling option is not included, the resulting program is suitable only as a foreground job and requires that a display and user interaction job be run in the background. (such as DISPLA or BAKSIM).

AQUIRE Main foreground routine. Coordinates the timing and calls appropriate routines for data acquisition, processing, and storage.

AQUTST Processed data plotter. Plots the power spectral estimates and outputs various other program results under user control. This routine is included only when AQUIRE is compiled using the /DEBUG option.

DATPLT Raw data plotter. Plots the raw data obtained from the EEG channels under user control. This routine is included only when AQUIRE is compiled using the /DEBUG option.

FFT Fast Fourier Transform. Routine supplied by DEC in the fortran extensions library to perform fast fourier transforms.

FILACT Data file activation. LOOKUP and ISAVES all data files used in MOMA.

FSTORE Data storage. Stores data on MOMA's data disk every MOMA cycle.

IADC A/D sampling. Routine supplied by DEC in the fortran extensions library to take a single sample from a specified number of A/D channels.

POWRSP Periodogram computation. Routine supplied by DEC in the fortran extensions library to compute a periodogram from the output of FFT.

PREPAR Preprocessing for power spectral estimation. Subtracts a given number (usually the mean of the array) from a given array and then uses WINDOW to taper the ends of the array.

PWRSPC Power spectral estimator. Uses a method of time averaging over short modified periodograms to estimate the power spectrum of a given sampled data signal.

RTS Real-time A/D sampling. Routine supplied by DEC in the fortran extensions library to repetitively sample a given number of A/D channels.

RTSCOM EEG preprocessing. Entered every time a subbuffer in RTS is filled to call routines which will preprocess the incoming EEG data. It reads the data in the sampling buffer into the data buffer, computes and saves the subbuffer average for each channel, and then uses PREPAR to preprocess the data for power spectral estimation.

SETR Programmable clock setter. Routine supplied by DEC in the fortran extensions library to set the KWV11-A programmable real-time clock.

SLOSMP Non-EEG sampling. Samples a given set of channels once every MOMA cycle.

SMOOTH Three point averager. Routine to smooth the power spectral estimate from PWRSPC with a three point average moving window.

WINDOW Time domain windowing. Routine to taper a given percent of the start and end of an array using a cosine taper.

DK:FORCMN.TXT to TT:

MOMA FOREGROUND COMMON BLOCK TABLE

COMPILED USING THE /DEBUG OPTION

ROUTINE \ COMMON BLOCK	COMAQU	GARBAG	IOCHAN	STORE	TIMES	ONEDIM
AQUIRE	X			X	X	
AQUTST	X				X	
DATPLT	X					X
FILACT			X			
FSTORE		X	X	X		
PREPAR					X	
PWRSPC		X			X	
RTSCOM	X				X	
SLOSMP						
SMOOTH						
WINDOW						

COMPILED WITHOUT THE /DEBUG OPTION

ROUTINE \ COMMON BLOCK	COMAQU	GARBAG	IOCHAN	STORE	TIMES	ONEDIM
AQUIRE	X			X		
FILACT			X			
FSTORE		X	X	X		
PREPAR						
PWRSPC		X				
RTSCOM	X					
SLOSMP						
SMOOTH						
WINDOW						

NOTE: ONEDIM is a graphics common block

COMAQU raw data buffers and pointers

```
INTEGER ICHBUF(512,4),INBUF(4,128)
REAL CHNAV(9,4)
COMMON/COMAQU/ CHNAV,ICHBUF,INBUF,IBEF,IBFOFS,IBLKCT
```

CHNAV = This array is loaded with the averages of 64 sample segments of EEG data from each of four channels by RTSCOM. These values are then used by PREPAR to preprocess ICHBUF for power spectral estimation. The nineth element of CHNAV will accept the average of the averages, ie; the average of 512 samples.

ICHBUF = Data buffer. Accepts 512 samples from each of four EEG channels. This buffer is split into 8 64-sample segments. RTSCOM copies the contents of a given segment of INBUF to ICHBUF and at the same time loads CHNAV with the averages of the segments. PREPAR then removes these averages from each segment and windows the segments with WINDOW. Finally, PWRSPC uses a data segmentation processing technique to compute a power spectral estimate of the data in this array.

INBUF = Cyclic sampling buffer. Used by the routine RTS to buffer the data as it is acquired. After 1 segment (64 samples) has been acquired, RTS signals that RTSCOM should be entered so as to preprocess that segment. RTS then proceeds to fill up the second segment of INPUT. Since RTS loops back to the first segment when the second segment is filled, RTSCOM must have completed its processing before that time.

IBEF = The free subbuffer indicator of the RTS routine. Incremented in RTSCOM To signal to RTS that its processing is complete.

IBFOFS = INBUF subbuffer offset pointer. Used by RTSCOM to switch between the two subbuffers in INPUT each time it is entered. Reset in AQUIRE at the start of a new set of 8 segments.

IBLKCT = Segment counter. Used in RTSCOM to keep track of the number of segments which have been acquired and distributed to ICHBUF. Reset in AQUIRE at the start of a new set of 8 segments.

GARBAG 256 word area used as a disk I/O buffer. Passed to other routines to be used for temporary data storage so as to conserve memory.

```
INTEGER IWRTBF(256)
COMMON/GARBAG/ IWRTBF
```

IWRTBF = 256 word area used as a disk I/O buffer and for temporary data storage in routines which do not interact with the disk.

IOCHAN contains channel information about the disk data files.
ie; permits the use IREOPN for all disk I/O.

```
   INTEGER ICHSAV(5,9)
   COMMON/IOCHAN/ ICHSAV
```

ICHSAV = Used by FILACT to ISAVES the status of all the files on the "MOMA slave disk". This will then be used in all the disk I/O routines (FSTORE) to IREOPN all the files, thus avoiding the use of the USR routine.

STORE processed data buffer.

```
   REAL POWER(32,4)
   INTEGER ISLDAT(4),MSGBUF(263)
   EQUIVALENCE (MSGBUF(1),POWER(1,1)),(MSGBUF(257),ISLDAT(1)),
 *  (MSGBUF(261),TIME),(MSGBUF(263),LOPCNT)
   COMMON/STORE/ MSGBUF
```

MSGBUF = buffer containing the data to be both sent to the background job and stored on disk. The organization is as follows;

```
MSGBUF(1-256)     == POWER(1-32,1-4) --> POWER1-4.DAT
MSGBUF(257-260)   == ISLDAT(1-4) ------> SLOCHN.DAT
MSGBUF(261-262)   == TIME -------------> SLOCHN.DAT
MSGBUF(263)       == LOPCNT -----------> STATUS.DAT
```

TIMES Routine timing flags (included when compiled with debug option)

```
   REAL TIMTAG(14)
   COMMON/TIMES/ TIMTAG
```

TIMTAG = Used to record the time of arrival at 14 predefined points in the data sampling and processing routines. The time is stored in RT-11's internal time format.

```
DK:FORCAL.TXT  to TT:
                  MOMA FOREGROUND CALLED ROUTINES
                  -------------------------------
                  (not including SYSLIB routines)


ROUTINE |         CALLED ROUTINE(S)
--------+-------------------------------------------------------
        |
        compiled without /DEBUG option
        |
AQUIRE  | FILACT,FSTORE,PWRSPC,RTS,RTSCOM,SETR,SLOSMP
FFT    *| ------
FILACT  | ------
FSTORE  | ------
IADC   *| ------
POWRSP *| ------
PREPAR  | WINDOW
PWRSPC  | FFT,POWRSP,SMOOTH
RTS    *| ------
RTSCOM  | PREPAR
SETR   *| ------
SLOSMP  | IADC
SMOOTH  | ------
WINDOW  | ------
        |
        compiled with /DEBUG option
        |
AQUIRE  | AQUTST,DATPLT,FILACT,FSTORE,PWRSPC,RTS,RTSCOM,SETR,SLOSMP
AQUTST  | AREA,AUTOSC,PLOT1D,RLINE,STRING,TITLE,XAXIS,YAXIS
DATPLT  | AREA,RLINE,STRING,TITLE,VECGEN,XAXIS,YAXIS
FFT    *| ------
FILACT  | ------
FSTORE  | ------
IADC   *| ------
POWRSP *| ------
PREPAR  | WINDOW
PWRSPC  | FFT,POWRSP,SMOOTH
RTS    *| ------
RTSCOM  | PREPAR
SETR   *| ------
SLOSMP  | IADC
SMOOTH  | ------
WINDOW  | ------
        |
        graphics routines from GPHLIB (called from debug version only)
        |
AREA    | ------
AUTOSC  | PARSET
LINTYP  | ------
LINS1D  | LINTYP,VECGEN
PARSET  | ------
PLINE   | RLINE
PLOT1D  | LINS1D
PNTOUT  | ------
RLINE   | LINTYP,VECGEN
STRING  | ------
TITLE   | PLINE,STRING
VECGEN  | PNTOUT
XAXIS   | PLINE,RLINE,STRING
YAXIS   | PLINE,RLINE,STRING

note;   * = LABLIB routine supplied in DEC fortran extensions package
```

MOMA FOREGROUND CALLING ROUTINES

ROUTINE		CALLING ROUTINE(S)
		compiled without /DEBUG option
AQUIRE		------
FFT	*	PWRSPC
FILACT		AQUIRE
FSTORE		AQUIRE
IADC	*	SLOSMP
POWRSP	*	PWRSPC
PREPAR		RTSCOM
PWRSPC		AQUIRE
RTS	*	AQUIRE
RTSCOM		AQUIRE
SETR	*	AQUIRE
SLOSMP		AQUIRE
SMOOTH		PWRSPC
WINDOW		PREPAR
		compiled with /DEBUG option
AQUIRE		------
AQUTST		AQUIRE
DATPLT		AQUIRE
FFT	*	PWRSPC
FILACT		AQUIRE
FSTORE		AQUIRE
IADC	*	SLOSMP
POWRSP	*	PWRSPC
PREPAR		RTSCOM
PWRSPC		AQUIRE
RTS	*	AQUIRE
RTSCOM		AQUIRE
SETR	*	AQUIRE
SLOSMP		AQUIRE
SMOOTH		PWRSPC
WINDOW		PREPAR
		graphics routines from GPHLIB (called from debug version only)
AREA		AQUTST,DATPLT
AUTOSC		AQUTST
LINTYP		LINS1D,RLINE
LINS1D		PLOT1D
PARSET		AUTOSC
PLINE		TITLE,XAXIS,YAXIS
PLOT1D		AQUTST
PNTOUT		VECGEN
RLINE		AQUTST,DATPLT,PLINE,XAXIS,YAXIS
STRING		AQUTST,DATPLT,TITLE,XAXIS,YAXIS
TITLE		AQUTST,DATPLT
VECGEN		DATPLT,LINS1D,RLINE
XAXIS		AQUTST,DATPLT
YAXIS		AQUTST,DATPLT

note; * = LABLIB routine supplied in DEC fortran extensions package

DK:FORTST.TXT to TT:

MOMA FOREGROUND TESTING ROUTINES

note: all these routines should be run in the background

ADCAL = Samples A/D channels 0 to 7 every second and types out the results. (quasidifferential mode). Terminated by typing "RETURN".

ADTST = Samples A/D channels 0 to 7 every clock tick (approx) for 32 ticks and types out the result. (quasidifferential mode).

AQUTMP = Uses the DATPLT and AQUTST routines. DATPLT is given modulated cosine patterns and AQUTST is given cosine patterns.

BAKSIM = Routine to run in the background while AQUIRE is run in the foreground. Simply takes AQUIRE's data update message and types it onto the terminal.

CLKT1 = Uses SETR for a high frequency KWV11-A clock test. Sets the clock to repeated interval noninterupt mode and pulses the clock overflow line low at 500 KHz (500 ns pulses). Pulses continue until "RETURN" is typed. Useful for looking at the clock overflow line with an oscilloscope. If ICMF is altered, an error message is generated and ICMF is typed out.

CLKT2 = Uses SETR for a low frequency KWV11-A clock test. Sets the clock to repeated interval interupt mode and pulses the clock overflow line low at 100 Hz (500 ns pulses). Pulses continue until "RETURN" is typed. If ICMF is altered, an error message is generated and ICMF is typed out.

CLKTST = Uses SETR with a completion routine for a low frequency KWV11-A clock test. Sets the clock to repeated interval interupt mode and pulses the clock overflow low at 64 Hz. A completion routine is entered every 64 samples. Within this routine, a counter is incremented. This counter is then typed out along with ICMF. If ICMF is altered, an error message is generated and ICMF is typed out.

FFTCHK = Runs a cosine through FFT and POWRSP. Types out data before entry to FFT, after FFT, and after POWRSP.

PRETST = Tests the RTSCOM routine with 4 test patterns. Note that PREPAR and WINDOW will indirectly entered.

PWRTST = Tests the PWRSPC routine with a double cosine test pattern. Note that FFT, POWRSP, and SMOOTH will be indirectly called.

RTCT1 = Uses the RTS-SETR combination to take 200 samples from A/D channels 0 and 1 at 100 Hz. RTS has a completion routine but SETR doesnt.

RTCTST = Uses the RTS-SETR combination to take 200 samples from A/D channels 0 and 1 at 100 Hz. Both RTS and SETR have completion routines.

RTSTST = Uses the RTS-SETR combination to take 100 samples from A/D channels 0 and 1 at 100 Hz. Neither RTS or SETR have completion routines.

SLOTST = Tests the SLOSMP routine. Uses SLOSMP to sample A/D channels 4 to 7 20 times at approximately 2 Hz and types out the results. Note that IADC is indirectly called.

A2.7 Background Routines Documentation

```
 Files copied:
DK:BAKEXP.TXT   to TT:
                   MOMA BAKCGROUND GENERAL DESCRIPTION
                   -----------------------------------

note: refer to the "MOMA BACKGROUND ROUTINE INTERCONNECTIONS" diagrams
      while reading this section.

        All background activity is initiated from DISPLA.  On startup,
DISPLA calls a number of initialization routines
(on start: LBLINI, FILACT, TVSET, and TAGSTO)
(on restart: LBLINI, FILACT, TVSET, and RESTRT)
and then sets up the terminal's screen for a type 1 display (EEG4DD).
Restart goes one step further and reads the records stored on disk and
updates the display (SCROLL, EEG4UP).
        After signalling the foreground routine to start sampling, DISPLA
simply waits for either an update message from the foreground or a typed
character.  When an update message is received, RCVCOM flags it and
DISPLA calls an appropriate display updating routine (EEG4UP, PLOTUP).
When a character is typed, DISPLA calls INPUT to service that character.
Note that TTY is in a no wait, special input mode.
        When INPUT determines that a complete line has been entered, the
line interpreting routine INTERP is called.  This routine will analyze
the line entered and dispatch to the appropriate service routine(s).
All timed tags and all instructions except RUBOUT, DELETE LINE and LAST LINE
will be serviced from INTERP.  The three exceptions mentioned above are
single character immediate action commands and are serviced by INPUT when
they are recieved.
        When INTERP returns to DISPLA (via INPUT),
it sets the argument ICHAR to produce three actions on return. 1) it can
return directly to the main idle loop, 2) it can run through the display
updating routine prior to going to the main idle loop, or 3) it can run
through both the display initialization routines and the display updating
routines prior to going to the main idle loop.  Thus, the function of INTERP
is to set up the display buffer and the mode flags and then to request
that DISPLA call appropriate display manipulation routines.
        The LUPDIS flags in the MODES common block are treated specially
by INTERP and DISPLA.  These flags control the update of the buffer and/or
the display.  When INTERP returns to DISPLA (via INPUT), it sets the argument
RESET to the value that the LUPDIS flags should be at complition of the
requested function.  LUPDIS is set equal to RESET within the
main idle loop of DISPLA.  Since LUPDIS is not set to RESET until the
idle loop of DISPLA is entered, one can set LUPDIS within INTERP to a mode
appropriate for use in routines entered before the idle cycle.  eg: when
a record is transfered from the scrolled section to the unscrolled section of
a type 1 display, RESET is set for active section update, LUPDIS is set for
nonactive section update, and ICHAR is set so that EEG4UP is entered before
returning to the main idle loop.  With this configuration, the unscrolled
section will be updated when INTERP returns to DISPLA but will not be touched
by subsequent DISPLA activity.
        Two display types are presently implemented on this system.
Display type 1 presents the EEG in a DSA format along with a digital display
of the non-EEG data and uses buffer modes 1 and 2.  Display type 2 plots
the non-EEG data vs time and uses buffer mode 3.  The routines for each
display type have been split into two catagories, initialization routines
and update routines.  The initialization routines (EEG4DD for type 1 and
PLOTDD for type 2) initialize the terminal's screen, plot axes, output
titles, etc.  The Update routines (EEG4UP for type 1 and PLOTUP for type 2)
handle the display buffer and output the data to the terminal.  Note that
since display type 2 is not updated on foreground message receipt, the
PLOTUP routine is set so that it changes the buffer and updates the
display only on display change.
        The display buffer can be loaded by three routines, BREAD, SLORED,
and DISBUP.  BREAD loads it in mode 1 or 2 with a given number of records
from the data disk.  SLORED performs the same function for mode 3 as BREAD
does for modes 1 and 2.  DISBUP loads it with data from the forground message
while in modes 1 and 2.  For modes 1 and 2, the EEG data is encoded to
ADE's of the gray levels for the DSA (ENCO16) and the non-EEG data is scaled
(ISLSCA) prior to buffering.  For mode 3, the non-EEG data is scaled (ISLSCA)
and bounded (BOUND) prior to buffering.  Note that since the type 2 displays
(mode 3) are not updated on foreground message receipt, it was not necessary
to include a mode 3 equivalent to DISBUP.
        If a type 1 display is currently active and a display change is
requested, it is necessary to remember what has been saved in the unscrolled
section.  This is performed by the routines NACTSV and NACTIN.  Just before
leaving a type 1 display, NACTSV is called to save the record numbers of the
records contained in the nonactive scroll.  When a new type 1 display is
subsequently requested, NACTIN is called to read these records into the
unscrolled section of the new display.
```

DK:BAKRTN.TXT to TT:

MOMA BACKGROUND ROUTINES DESCRIPTIONS

NOTE: DISPLA, the main background routine, can be run with or without a foreground (data acquisition and storage) job present. However, if a foreground job is NOT present, the only meaningful startup mode is restart.

BFBLNK Display buffer line blanking. Blanks one line of the display buffer.

BOUND Non-EEG data bounder. Sets upper and lower limits on the non-EEG data for type 2 displays. If the data is outside a limit it is set equal to that limit before being inserted into the display buffer.

BREAD Disk data read. Reads a specified number of data records from a specified point in time to the display buffer.

COPY Display buffer line duplication. Copies the contents of one line of the display buffer onto another line of the display buffer.

D*SET Display change parameter setting. Sets global parameters for a specified display.

DD2SET Terminal screen attribute setter. Sets the attributes of a given number of lines on the terminal's screen for a DSA display of a given format.

DISBUP Display buffer update. Takes the content of the buffer which receives the foreground messages and, after formating it, inserts it into the display buffer.

DISPLA Main background routine. When the background is inactive, the system waits in this routine. When a request for background activity is received from either the foreground job or the user, this routine dispatches to the appropriate service routine(s).

EEG4DD Type 1 display setup. Calls appropriate routines to set up a given type 1 display.

EEG4UP Type 1 display update. Calls appropriate routines to update the display buffer and/or the display for type 1 displays. Note that this routine is entered on the receipt of new data from the foreground.

EEGDLN DSA line sender. Sends one line of DSA to the terminal. A number of output formats are available.

ENC016 Density encoder. Encodes a power spectral estimate to a series of ADEs for gray level characters fo[illegible] character font 16. Character intensity is directly proportional to the magnatude of the power spectral estimate at a given frequency and time.

FASTDD Type 1 DSA display setup. Sets up the terminal's screen for a given DSA display and outputs the labels, axes, etc..

FASTUP Type 1 DSA update. Updates the DSA part of type 1 displays.

FILACT Data file activation. LOOKUP and ISAVES all data files used in MOMA.

FNTS* Font 16 initialization. Initializes the character font (font 16) used to generate the density modulation in DSA displays.

INCBUF Display buffer pointer mover. Moves the display buffer's pointer forward or backward a given number of lines.

INPUT Character input handler. If the received character is a single character immediate action command (RUBOUT, CTRL/U, or LAST LINE), it is serviced. If the received character is a carriage return that terminates a timed tag or a command, INTERP is called. If the received character is a trivial carriage return, it is ignored. If none of the above and no more than MAXCHR characters have been entered for the current line, the character is inserted into MOMA's line input buffer.

INTERP Line interpreter. After being called from INPUT, this routine interprets the line in MOMA's line input buffer and chooses an appropriate course of action.

INQUERY Question asker. Routine to send a question to the user and to wait for the responce. The first character of this responce is sent back to the calling program.

ISLSCA Non-EEG data scaler. Scales the non-EEG data according to a given multiplicative and additive constants.

ISWTCH Display buffer context switch. Changes the destination of all display buffer operations to the other section ie; switches between the scrolled section and the unscrolled section of the display buffer.

LBLINI Label initialization. Initializes the common block containing the labels to be used for the non-EEG data.

MEXIT Termination routine. Routine entered to stop the MOMA programs.

MOMKEY Terminal key programmer. Programs the terminal's keys for MOMA.

NACTIN Unscrolled section record number recovery. Takes the record numbers saved by NACTSV and reads the appropriate records into the unscrolled section of the display buffer.

NACTSV Unscrolled section record number save. Saves the numbers of the records stored in the Unscrolled section of the display so that they will be preserved during a display change.

PLOTDD Type 2 display initialization. Initializes the terminal screen for time plots and types out the axes, labels, titles, etc.

PLOTUP Type 2 display update. Calls appropriate routines to read data from the disk and update the display. Note that this routine is not entered on the receipt of new data from the foreground.

RCVCOM Foreground message flagger. Entered as a completion routine and raises a flag to tell DISPLA that the foreground has sent a message. This flag is dropped in DISPLA when the message is serviced

RESTRT LOPCNT initialization for restart. Reads LOPCNT from the STATUS file and sets it in the background job. This setting is sent to the foreground through the start message.

SCROLL Sets the calling arguments for BREAD so that the display buffer will be moved forward or back by a specified number of records.

SLOOUT Type 2 display data output. Outputs the data stored in the display buffer to the terminal.

SLORED Type 2 data reader. Reads data from disk to the display buffer for type 2 displays.

SLOWDD Type 1 non-EEG display setup. Outputs the labels for the digital display of non-EEG data in type 1 displays.

SLOWUP Type 1 non-EEG update. Updates the digital display of non-EEG data in type 1 displays.

TAGSTO Timed tag storage on disk. Stores a given timed tag on disk.

TBLANK Tag blanking. Blanks the timed tag portion of a given line of the display buffer.

TINSRT Tag insertion. Inserts a timed tag into a specified line of the display buffer.

TGREAD Timed tag reading. Checks if a timed tag has been attached to the records contained in a specifed number of lines of the display buffer. It then blanks the timed tag portion of those lines which do not have attached timed tags and reads tags into those that do.

TRNSFR Takes a line from the top of one section of the display buffer and inserts it to the other section. Control is returned to is returned to the first section when this routine is terminated.

TVSET Terminal screen initialization. Calls routines which will set the terminal's modes, program the keys, and initialize a character font for the DSA (Font 16).

DK:BAKCMN.TXT to TT:

MOMA BACKGROUND COMMON BLOCK TABLE

ROUTINE \ COMMON BLOCK	BLANK	DISBUF	DSPLAY	INPTRT	INPUTB	IOBUF	IOCHAN	MODES	NACT	NONEEG	PNTSAV	STATE	USRTAG	PLOT
BFBLNK	X	X												
BOUND														
BREAD		X	X			X	X					X		
COPY		X												
D*SET		X	X					X			X	X	X	
DD2SET														
DISBUP		X	X									X	X	
DISPLA	X	X	X				X	X	X	X		X		
EEG4DD			X					X				X		
EEG4UP		X	X					X						
EEGDLN														
ENC016														
FASTDD														
FASTUP		X												
FILACT							X							
FNTS*														
INCBUF														
INPUT				X	X								X	
INTERP		X	X	X	X			X				X	X	
IQUERY														
ISLSCA														
ISWTCH		X									X			
LBLINI										X				
MEXIT														
MOMKEY														
NACTIN		X							X					
NACTSV		X							X					
PLOTDD			X							X		X		
PLOTUP								X						
RCVCOM												X		
RESTRT						X	X							
SCROLL		X										X		
SLOOUT		X	X									X		X
SLORED		X	X			X	X					X		
SLOWDD										X				
SLOWUP		X								X				
TAGSTO						X	X							
TBLANK	X	X	X											
TINSRT		X	X											
TGREAD		X	X			X	X							
TRNSFR		X												
TVSET														

NOTE: PLOT is a common block from the graphics package

MOMA BACKGROUND COMMON BLOCK DESCRIPTION

BLANK defines a 32 character resion containins spaces

```
LOGICAL*1 LSPACE(32)
COMMON/BLANK/ LSPACE
```

LSPACE = losical*1 array initialized to 32 space characters in DISPLA on system startup.

DISBUF display buffer and pointers

for display type 1 (D1 to D4 or buffer modes 1 and 2)

```
LOGICAL*1 LPW4BF(32,10,4),LPW2BF(32,20,2),LTGBUF(9,20)
EQUIVALENCE (LPW4BF(1,1,1),LPB2BF(1,1,1))
INTEGER LOPBUF(20),ITMBUF(2,20),ISLBUF(4,20)
COMMON/DISBUF/ IDSBFP,IBFTOP,IBFBOT,IBFMOD,
     if IBFMOD = 1; the rest of DISBUF = LOPBUF,ITMBUF,ISLBUF,LPW4BF,LTGBUF
     if IBFMOD = 2; the rest of DISBUF = LOPBUF,ITMBUF,ISLBUF,LPW2BF,LTGBUF
                                         ----------- 870 words -----------
```

IDSBFP = Display buffer pointer. Gives the buffer offset of the record which resides at the top of the currently active section of the display buffer.

IBFTOP = Top of currently active section pointer. Gives the hishest buffer offset allocated to the currently active section of the display buffer.

IBFBOT = Bottom of currently active section pointer. Gives the lowest buffer offset allocated to the currently active section of the display buffer.

note; On startup or display chanse, the initial settins of IBFTOP and IBFBOT define the "scrolled section" of the display buffer. ie; the scrolled section incorporates all buffer locations between IBFBOT and IBFTOP inclusive and the unscrolled section will occupy the remainins buffer locations. Note that since the rows allocated to these sections must be contisuous, either IBFBOT must be set to 1 and/or IBFTOP must be set to the larsest offset possible (10 for mode 1 and 20 for mode 2). The ISWTCH routine may produce undesireable results if this condition is not satisfied.

IBFMOD = Buffer mode indicator. Indicates the way in which the 870 words of buffer are used.
= 1; 4 sections of 10 DSA records, each with 32 elements.
= 2; 2 sections of 20 DSA records, each with 32 elements.
= 3; see below

LOPBUF = MOMA cycle numbers of the buffered records.

ITMBUF = Time labels for the buffered records.
ITMBUF(1,*) = the hour of the sample
ITMBUF(2,*) = the minute of the sample

```
ISLBUF = Buffer for the non-EEG data.
                ISLBUF(N,*) = data sample from the Nth non-EEG channel
                              1<=N<=4

LPW*BF = Buffer for the DSA.  Each element contains an ASCII character
         which will be represented on the terminal's screen as a
         gray level according to the definitions in character font 16.
           LPW4BF(32,*,4) = 32 elements of DSA from 4 channels.
                          - maximum of 10 records
           LPW2BF(32,*,2) = 32 elements of DSA from 2 channels.
                          - maximum of 20 records
           - note that instead of DSA data, it is possible to
             buffer other things, such as timed tags.
             (see displays D3 and D4)

LTGBUF = An 8 character buffer for the storage of timed tags.
         A nineth element was included becaure a null character
         is inserted as a character string terminator by the
         system's character string handling routines.

      for display type 2 (D5 to D6 or buffer mode 3)
REAL SLOBUF(4,79),TMBUF(79)
INTEGER LOPBUF(79)
COMMON/DISBUF/ IDSBFP,IBFTOP,IBFBOT,IBFMOD,
      if IBFMOD = 3; the rest of DISBUF = LOPBUF,TMBUF,SLOBUF
                                          ---- 869 words ----

IDSBFP = Not used

IBFTOP = Not used

IBFBOT = Not used

IBFMOD = Buffer mode indicator.  Indicates the way in which the
         870 words of buffer are used.
        = 1; See above
        = 2; See above
        = 3; Buffering for up to 79 samples from 4 non-EEG channels.
             Note that the graphics routines use real numbers so
             real numbers are buffered.

LOPBUF = MOMA cycle numbers which correspond to the buffered samples.

TMBUF  = Buffer for the storage of times at which the buffered samples
         were acquired.  Time is stored in RT-11's internal time format.

SLOBUF = Storage for up to 79 samples from 4 non-EEG data channels.
         This data is bounded by BOUND and scaled by ISLSCA prior
         to storage in this array.  Real numbers are stored in order
         to obtain compatability with the graphics package.
```

DSPLAY parameters related to the display configuration

for display type 1 (D1 to D4 or buffer mode 1 and 2);

```
INTEGER ITMFRQ(2),IROW(2),LENTHY(2),ICH(4)
COMMON/DSPLAY/ IDISPL,NUMSLO,LINE,NUMEEG,NUMSEC,INC,ITMFRQ,IROW,LENTHY,ICH
```

IDISPL = The number of the currently active display.

NUMSLO = The number of channels of non-EEG data buffered and displayed.
ie; ISLBUF(1,*) to ISLBUF(NUMSLO,*) will be used to buffer data from the first NUMSLO non-EEG channels. The non-EEG data associated with the EEG record at the top of the scrolled section will be displayed in a digital format.

LINE = The top row in the workspace portion of the terminal's screen which will be used for the digital display of non-EEG data. The following NUMSLO/2-1 rows will be used for this purpose.

NUMEEG = The number of EEG channels buffered for display.
ie; LPW#BF(32,*,1) to LPW#BF(32,*,NUMEEG) will be used to buffer EEG data. The EEG channels which are buffered in these locations are determined by ICH.

NUMSEC = Using the routine FASTUP, ASCII data from two of the EEG channel's buffers is typed out side by side. When using the 4 channel buffering mode (mode 1), there are two regions of the screen into which output in this format will be typed. These regions have been called "display regions". It is possible to output to only one display region while in mode 1, thus using only half of the display buffer. This is done by setting NUMSEC.

Buffer mode 1,	NUMSEC=1;	one display region, only half of the buffer used
	=2;	two display regions, complete buffer is used.
Buffer mode 2,	NUMSEC=1;	one display region, complete buffer is used.

Note that using FASTUP in its present form and while in buffer mode 1, buffer channels 1 and 3 are typed into the first display region and buffer channels 2 and 4 are typed into the second display region. This allocation of buffer channels can easily be changed by altering the EEGDLN calls in FASTUP and/or changing the setting of the parameter IOFST within FASTUP.

INC = The record increment between consecutive buffered records. within the scrolled section.
ie; if INC = 2, every second record acquired by MOMA will be buffered and displayed, even though every record is stored on disk.

ITMFRQ = The frequency of time labelling in DSA displays.
ITMFRQ(1) = for the scrolled section
ITMFRQ(2) = for the unscrolled section
ie; if ITMFRQ(1)=2, every second display line in the scrolled section of the DSA displays will have a time label attached to it. Note, however, that ITMFRQ settings affect only the display of time labels, every buffered record will have an accurate time label in ITMBUF. attached to it.

IROW = The top row in the workspace region of the terminal's screen to which the display regions are to be written. Display regions are described under NUMSEC.
IROW(1) = the top row of display region 1.
IROW(2) = the top row of display region 2.

LENTHY = The length, in rows, of the Y axis of the DSA displays in a given display region. Note that this sets the length of the Y axis and not the size of the display region. Thus, if the specified Y axis length is smaller than the size of the display region the X axis will be overwritten on display update.

LENTHY(1) = length of the Y axis in display region 1
LENTHY(2) = length of the Y axis in display region 2

ICH = Data to buffer connection numbers. Associates a given EEG data channel with a given EEG buffer channel. ie; if ICH(2)=3, the data acquired by the third A/D channel used for EEG sampling and the data stored in POWER3.DAT on the data disk will be buffered in LPW#BF(*,*,2).

for display type 2 (D5 to D6 or buffer mode 3);

```
INTEGER ICH(4)
COMMON/DSPLAY/ IDISPL,NUMSLO,ILINE,ICOLMN,XSIZE,YSIZE,ISTREC,NREC,INC,ICH
```

IDISPL = The number of the currently active display

NUMSLO = The number of channels of non-EEG data to be displayed. Note that all four non-EEG channels are read but only NUMSLO of these channels will be plotted.

ILINE = The bottom edge of the time plot. ie; If ILINE=30, row 30 of the terminal's workspace area will be the bottom row (highest row number) used for the time plot.

ICOLMN = The left edge of the time plot. ie; If ICOLMN=30, column 30 of the terminal's workspace area will be the leftmost column (lowest column number) used for the time plot.

XSIZE = The distance, in cm's, which the time plot will extend in the X direction. $10.8 < XSIZE < 21.6$

YSIZE = The distance, in cm's, which the time plot will extend in the Y direction. $7+NUMSLO/2 < YSIZE < 14.7$

ISTREC = The most recient record of non-EEG data to be buffered and displayed.

NREC = The number of records to be buffered and displayed.

INC = Record increment between consecutive buffered records.

ICH = Buffer to display connection numbers. Connects a given display buffer channel with a given plotting order.
ie; Data from non-EEG data channels which is stored on disk in the file ISLDAT.DAT is read into the display buffer. The data channel numbers are the same as the buffer channel numbers. The data buffered in the ICH(I)th buffer channel is assigned linetype I and is the Ith channel plotted. Thus, if ICH(2)=3, the second non-EEG data channel, which is buffered in the second buffer channel, is assigned linetype 3 and is the third channel plotted.

INPTRT parameters used in INPUT and INTERP
- used to prevent problems in overlaying INTERP (INPUT in root)

```
COMMON/INPTRT/ ICOMCH,ITAGQU,IDIR,IUPOFF,UPSAVE
```

```
      ICOMCH = The ADE for MOMA's command character
                 - usually set to 123 or "{".

      ITAGQU = Query on timed tag enable/disable flag.
               =0; a query is sent on receipt of a timed tag
               =1; "   "   " NOT "  "    "    " "   "   "

      IDIR   = Scrolling direction indicator
               =1;  scrolling operations move forward in time.
               =-1;     "           "         " backward "   "

      IUPOFF = Display update indicator
               =0; displays and buffers updated on foreground message receipt
               =1;     "        "    "    NOT  "     "          "        "      "

      UPSAVE = Temporary storage location for UPDIS when display update is
               turned off.  Inserted to prevent the loss of this value
               when INTERP is overlayed.
```

INPUTB character buffer used to buffer lines entered at the keyboard

```
LOGICAL*1 LINBUF(82)
COMMON/INPUTB/ ICHCNT,LINBUF
```

ICHCNT = The number of characters presently in the line input buffer.

LINBUF = The line input buffer.

IOBUF 256 word area used as a disk I/O buffer

```
INTEGER IREADB(256)
COMMON/IOBUF/ IREADB
```

IREADB = A 256 word area to be used as a read/write buffer for disk I/O.

IOCHAN contains channel information about the data files.
ie; permits the use of IREOPN for all disk I/O.

```
INTEGER ICHSAV(5,9)
COMMON/IOCHAN/ ICHSAV
```

ICHSAV = Used by FILACT to ISAVES the status of all the files on the "MOMA slave disk". This will then be used in all the disk I/O routines to IREOPN the files without the use of the USR routine.

```
MODES   flags controlling the mode of buffer and display updates

        for display type 1 (D1 to D4 or buffer modes 1 and 2)
   LOGICAL*1 LUPDIS(2,2)
   INTEGER IMODE(2)
   COMMON/MODES/ LUPDIS,IMODE,IFORM

        LUPDIS = Used in EEG4UP to control the update of the
                  data buffer and/or the display.
                 LUPDIS(1,1) = 1; update scrolled section of data buffer
                 LUPDIS(2,1) = 1;    "       "         "     "  display
                 LUPDIS(1,2) = 1;    "  unscrolled     "     "  data buffer
                 LUPDIS(2,2) = 1;    "       "         "     "  display

        IMODE  = Indicates the way in which data is to be put into
                  the display buffer.
                 = +2 or -2; data is overlayed onto the display buffer.
                 = +1 or -1;   "    "  inserted  "     "       "      "
                 = positive; data is placed at the top of the display buffer.
                 = negative;  "     "    "     "  " bottom "  "     "       "

        IFORM  = Indicates the output format of the lines sent from EEGDLN.
                 = 1; parallel DSA displays with NO font setting instructions
                      in the character strings sent.
                 = 2; Parallel DSA displays with font setting instructions
                      in the character strings sent.
                 = 3; Right side of screen contains a DSA display, left side is
                      used for long timed tags.  NO font setting instructions
                      in the character strings sent.
                 = 4; Right side of screen contains a DSA display, left side is
                      used for long timed tags.  Font setting instructions
                      in the character strings sent.

        for display type 2 (D5 to D6 or buffer mode 3)
   COMMON/MODES/ UPDIS

        UPDIS  = used by PLOTUP to permit or block entry to SLORED and SLOOUT.
                 = 0; SLORED and SLOOUT are NOT entered.  PLOTUP returns to
                      its calling routine (DISPLA) immediately on entry.
                 = anything else;  SLORED and SLOOUT are entered from PLOTUP.
```

NACT area where the record numbers of records in the unscrolled section of the last type 1 display. (saved by NACTSV)

```
INTEGER NACTNM(20)
COMMON/NACT/ NACTNM
```

NACTNM = Record numbers of the unscrolled section of the previous type 1 display. This array is loaded by NACTSV at the start of a display change and then loaded by NACTIN into the unscrolled section of the next type 1 display.

NONEEG labels for the non-EEG displays

```
LOGICAL*1 LABELS(32,4)
INTEGER LBLLNG(4)
COMMON/NONEEG/ LBLLNG,LABELS
```

LBLLNG = The length of the labels contained in LABELS.

LABELS = Labels that are to be used to identify the non-EEG data in both type 1 and type 2 displays. These labels can be up to 31 characters in length and are initialized in the routine LBLINI on system startup.

PNTSAV display buffer pointer save locations used by ISWTCH

```
COMMON/PNTSAV/ IPNT1,IPNT2
```

IPNT1 = Saves the position of the scrolled section pointer when ISWTCH switches to the unscrolled section.

IPNT2 = Saves the position of the unscrolled section pointer when ISWTCH switches to the scrolled section.

note; both these parameters should be set to 0 whenever the display is changed.

STATE scale parameters and parameters indicating the general state of MOMA

```
REAL SLOMLT(4),SLOADD(4),SLOTOP(4),SLOBOT(4)
COMMON/STATE/ MSGFLG,LOPCNT,ITINCR,PMAX,FMAX,SLOMLT,SLOADD,SLOTOP,SLOBOT
```

MSGFLG = raized by RCVCOM to signal the arrival of an update message from the foreground job (AQUIRE). This flag is dropped by DISPLA after the message is serviced.
=0; no message has been queued yet.
=1; a message has been queued.

LOPCNT = The number of MOMA cycles that have already occured. This parameter is used to index the data on the data disk.

ITINCR = not used

PMAX = Power magnatude corresponding to the maximum intensity character in the DSA.

FMAX = The frequency, in cycles per second, which corresponds to the end of the frequency axis in DSA plots. This parameter is only used for the determination of scaling labels.

SLOMLT = The multiplicative scaling factors used in ISLSCA for the four non-EEG data channels. ISLSCA is called prior to buffering the data for both type 1 and type 2 displays.

SLOADD = The additive scaling factors used in ISLSCA for the four non-EEG data channels. ISLSCA is called prior to buffering the data for both type 1 and type 2 displays.

SLOTOP = The upper limits used in BOUND for the four non-EEG data channels. BOUND is called prior to buffering the data for type two displays only (time plots).

SLOBOT = The lower limits used in BOUND for the four non-EEG data channels. BOUND is called prior to buffering the data for type two displays only (time plots).

USRTAG buffer to communicate timed tags to various routines

```
LOGICAL*1 LTAG(32)
COMMON/USRTAG/ ITGFLG,LTAG
```

ITGFLG = Flags the existance of an active timed tag in LTAG. When DISBUP incounters this flag, it inserts that tag into the display buffer along with the just received foreground message.
= -1; a timed tag is present.
= anything else; NO timed tag present.

LTAG = Contains timed tags of up to 31 characters.

DK:BAKCAL.TXT to TT:

MOMA BACKGROUND CALLING ROUTINES

ROUTINE	CALLING ROUTINE(S)
BFBLNK	BREAD
BOUND	SLORED
BREAD	NACTIN,SCROLL
COPY	TRNSFR
D*SET	DISPLA,INTERP
DD2SET	EEG4DD
DISBUP	EEG4UP
DISPLA	------
EEG4DD	DISPLA
EEG4UP	DISPLA
EEGDLN	FASTUP
ENC016	BREAD,DISBUP
FASTDD	EEG4DD
FASTUP	EEG4UP
FILACT	DISPLA
FNTS*	TVSET
INCBUF	BREAD,DISBUP,NACTSV,SCROLL,TBLANK,TGREAD,TRNSFR
INPUT	DISPLA
INTERP	INPUT
IQUERY	DISPLA,INTERP
ISLSCA	BREAD,DISBUP,SLORED
ISWTCH	EEG4UP,NACTIN,NACTSV,TRNSFR
LBLINI	DISPLA
MEXIT	INTERP
MOMKEY	TVSET
NACTIN	INTERP
NACTSV	INTERP
PLOTDD	DISPLA
PLOTUP	DISPLA
RCVCOM	DISPLA
RESTRT	DISPLA
SCROLL	DISPLA,INTERP
SLOOUT	PLOTUP
SLORED	PLOTUP
SLOWDD	EEG4DD
SLOWUP	EEG4UP
TAGSTO	DISPLA,INTERP
TBLANK	DISBUP,TGREAD
TINSRT	DISBUP,TGREAD
TGREAD	BREAD
TRNSFR	INTERP
TVSET	DISPLA,INTERP

GRAPHICS ROUTINES

AREA	PLOTDD
LINS1D	PLOT1D
LINTYP	LINS1D,RLINE
PLINE	XAXIS,YAXIS
PLOT1D	SLOOUT
PNTOUT	VECGEN
RLINE	PLINE,PLOTDD,SLOOUT,XAXIS,YAXIS
STRING	PLOTDD,SLOOUT,XAXIS,YAXIS
VECGEN	RLINE,LINS1D
XAXIS	PLOTDD
YAXIS	PLOTDD

MOMA BACKGROUND CALLED ROUTINES

(not includins SYSLIB routines)

ROUTINE	CALLED ROUTINE(S)
BFBLNK	------
BOUND	------
BREAD	BFBLNK,ENCO16,INCBUF,ISLSCA,TGREAD
COPY	------
D*SET	------
DD2SET	------
DISBUP	ENCO16,INCBUF,ISLSCA,TBLANK,TINSRT
DISPLA	D1SET,EEG4DD,EEG4UP,FILACT,INPUT,IQUERY,LBLINI
	PLOTDD,PLOTUP,RCVCOM,RESTRT,SCROLL,TAGSTO,TVSET
EEG4DD	DD2SET,FASTDD,SLOWDD
EEG4UP	DISBUP,FASTUP,ISWTCH,SLOWUP
EEGDLN	------
ENCO16	------
FASTDD	------
FASTUP	EEGDLN
FILACT	------
FNTS*	------
INCBUF	------
INPUT	INTERP
INTERP	D*SET,IQUERY,MEXIT,NACTIN,NACTSV,SCROLL,TAGSTO,TRNSFR,TVSET
IQUERY	------
ISLSCA	------
ISWTCH	------
LBLINI	------
MEXIT	------
MOMKEY	------
NACTIN	BREAD,ISWTCH
NACTSV	INCBUF,ISWTCH
PLOTDD	AREA,RLINE,STRING,XAXIS,YAXIS
PLOTUP	SLOOUT,SLORED
RCVCOM	------
RESTRT	------
SCROLL	BREAD,INCBUF
SLOOUT	PLOT1D,RLINE,STRING
SLORED	BOUND,ISLSCA
SLOWDD	------
SLOWUP	------
TAGSTO	------
TBLANK	INCBUF
TINSRT	------
TGREAD	INCBUF,TBLANK,TINSRT
TRNSFR	COPY,INCBUF,ISWTCH
TVSET	FNTS*,MOMKEY

GRAPHICS ROUTINES

AREA	------
LINS1D	LINTYP,VECGEN
LINTYP	------
PLINE	RLINE
PLOT1D	LINS1D
PNTOUT	------
RLINE	LINTYP,VECGEN
STRING	------
VECGEN	PNTOUT
XAXIS	PLINE,RLINE,STRING
YAXIS	PLINE,RLINE,STRING

DK:BAKTST.TXT to TT:

MOMA BACKGROUND TESTING ROUTINES

BLNKT = Tests the BFBLNK routine. Loads the text buffer with garbage, types it out, calls BFBLNK 20 times to blank the buffer, and types it out again.

ENCTST = Tests the ENC016 routine. Loads the power array with numbers from -6 to 66 and calls ENC016 with PMAX set to 60. Will result in the full range of gray levels being typed out and will saturate at 0 and 60.

IQUTST = Tests the IQUERY routine. Sends queries both with and without an appended "(Y or N)". Responding with "S<RETURN>" stop the program.

FNTTST = Tests all characters in character font 16. Types the ASCII characters from 32 to 100 on one line and types the font16 equivalents below. Uses FNTS* to initialize the character font.

FORSIM = Routine to run in the foreground while DISPLA is running in the background. Sends 4 tests patterns in a message of the same format as AQUIRE. If compiled using the /DEBUG option, it also stores these patterns on the slave disk.

PLTDDT = Checks out the data plotting routine. Calls FILACT, LBLINI, PLOTDD, and PLOTUP and indirectly calls SLOOUT, SLORED, BOUND, and ISLSCA to plot the non-EEG data stored on the data disk.

Files copied:

A2.8 Graphics Routines Documentation

DK:GPHEXP.TXT to TT:

GPHLIB GENERAL DESCRIPTION

note; the "GRAPHICS ROUTINE INTERCONNECTION" diagram should be viewed while reading the following text.

GPHLIB.OBJ contains a number of fortran graphics routines designed for the Tektronix 4025 graphics terminal. These routines make use of the GRA, LIN, STR, and VEV commands and provide the user with a data plotting and figure generation capability. It is assumed that the terminal's command character is backquote "`" and that the command lockout key is off. Time delays have been put into the routines so that every function can be used with the terminal at 9600 baud and no fill characters between lines. However, when a number of these functions are called in rapid succession, it may be necessary to insert extra time delays into the calling program.

Subroutines have been written to output four of the terminal's graphics commands; AREA sends the GRA command, LINTYP sends the LIN command, STRING sends the STR command, and VECGEN sends the VEC command. BORDER simply outlines the graphics area with a line of given linetype but does not make use of VECGEN. All other routines in the package directly or indirectly call these routines when the LIN, STR, and VEC commands are needed. Note that AREA must be the first routine called since it initializes various common block parameters and scale factors.

On the next level, a number of simple line generation routines were written. RLINE generates a line with given rectangular coordinates, PLINE generates a line with given polar coordinates, LINES connects the points contained in a given array, and LINS1D is a single dimension version of LINES. By single dimension I mean that the routine is given an array containing all the points of one dimension, an additive factor A, and a multiplicative factor B. The unspecified dimension is then obtained by assuming it conforms to the equation R(I) = B*I+A. The above routines use VECGEN and LINTYP.

Making use of the line generation routines are two classes of "higher level" routines; figure generation routines and plotting routines. The following two paragraphs describe these two catagories.

One main figure generation routine was written (FIGURE). This routine draws a NUM sided figure inscribed at a given angle within an undrawn circle of given radius. This routine is then used to generate arcs which subtend a given angle (CURVE) and NUM sides polygons (POLY). WARP extends the capability of these routines. When called, WARP alters the scaling factors which convert the calling arguments of VECGEN to pixels. Changing these factors has the effect of stretching or compressing the X and/or the Y dimension. Thus, for example, a circle will be converted to an oval and a square will be converted to a rectangle.

A number of routines for the generation of data plots have been written. XAXIS and YAXIS draw the axes for the plots in a format which provides for the labelling and scaling. TITLE places an underlined character string at the top of the Y axis and centered along the X axis. PLOT and PLOT1D can then be used to output data to this graph according to the axis scaling factors. PLOT outputs X,Y coordinates specified in an array and PLOT1D is given an array containing either the X data or the Y data and plots by linearly incrementing the other dimension (see the description of LINS1D). Scaling of the axes can be done in two ways. Firstly, absolute scale factors can be specified as arguments for the axis generation routines. Secondly, AUTOSC can be used to set the axis scaling factors to values which accomodate the data in a given array. For this routine to function properly, it must be called before either XAXIS or YAXIS because it alters the way in which the arguments to these routines are handled. The only other limitation on the calling order of plotting routines is that XAXIS and YAXIS must be called before TITLE, PLOT and/or PLOT1D are called.

DK:GPHRTN.TXT to TT:

GPHLIB ROUTINES DESCRIPTIONS

All the following routines are found in the object library GPHLIB.OBJ.

AREA Graphics region setting. Routine to specify an area on the terminal's screen into which all subsequent graphical figures are to be sent. This routine must be one of the first graphics routines called in order to get meaningful results.

AUTOSC Autoscaling. Sets scaling values for XAXIS and YAXIS according to the maximum and minimum of a given array.

BORDER Graphics region border. Draws a line of given linetype around the outside of the graphics region specified by AREA.

CURVE Curve drawing. Draws a curve of a given radius and angle centered at a given point on the screen. There is also an option to connect the ends of the curve to the centre, thus proucing a pie shaped figure.

FIGURE General figure drawing. Draws a NUM sided figure inscribed in an undrawn circle of given radius and origin. The first side is at a given angle from the horizontal and each side describes a given angle.

LINES Multiple line generation. Connects the points specified in a given array with a line of given linetype.

LINS1D Multiple single coordinate line generation. Given an array containing data for one coordinate, this routine plots a line which increments the other coordinate by a specified value.

LINTYP Linetype setting. Sends a linetype changing instruction to the computer if necessary.

PARSET Plot parameter setting. Used by AUTOSC to set the scaling parameters for subsequent plots.

PLINE Polar cooridinate line. Generates a line given a set of polar coordinates.

PLOT Data plotting. Plots the X,Y coordinates in an array according to the scaling parameter settings set by XAXIS, YAXIS, and/or AUTOSC.

PLOT1D Single dimension data plotting. Plots single dimensional data vs a given increment.

PNTOUT Coordinate output. Used by VECGEN to output a given X,Y coordinate to the terminal.

POLY Polygon generation. Generates a NUM sided polygon.

RLINE Rectangular coordinate line. Generates a line given a set of rectangular coordinates.

STRING Character string output. Outputs a character string to a given location in the graphics area.

TITLE Graph title. Produces an underlined title centred along the X axis and one line above the Y axis.

VECGEN Vector generation. Generates vectors connecting the points specified in a given array.

WARP Dimension warping. Streaches or compresses the X and the Y dimension by given factors for all subsequent figures. Using this feature, for example, draw ovals instead of circles or rectangles instead of squares.

XAXIS X axis generation. Generates an X axis at a given position in the graphics area.

YAXIS Y axis generation. Generates a Y axis at a given position in the graphics area.

DK:GPHCMN.TXT to TT:

GRAPHICS COMMON BLOCK TABLE

```
              /   / P / M /   /
     COMMON  /   / Y / I /   /
     BLOCK  / A / T / D / T /
           / E / N / E / O /
          / R / I / N / L /
ROUTINE  / A / L / O / P /
--------+---+---+---+---+
AREA    | X | X |   |   |
AUTOSC  |   |   |   | X |
BORDER  | X |   |   |   |
CURVE   |   |   |   |   |
FIGURE  |   |   |   |   |
LINES   |   |   |   |   |
LINS1D  |   |   | X |   |
LINTYP  |   | X |   |   |
PARSET  |   |   |   |   |
PLINE   |   |   |   |   |
PLOT    |   |   |   | X |
PLOT1D  |   |   |   | X |
PNTOUT  |   |   |   |   |
POLY    |   |   |   |   |
RLINE   |   |   |   |   |
STRING  | X |   |   |   |
TITLE   | X |   |   | X |
VECGEN  | X |   | X |   |
WARP    | X |   |   |   |
XAXIS   | X |   |   | X |
YAXIS   | X |   |   | X |
```

AREA   scale factors and graphics region dimension information

COMMON/AREA/ SCALX,SCALY,IDIMX,IDIMY

see note 1

SCALX = X dimension scale factor. The X coordinate of all arguments sent to VECGEN will be multiplied by SCALX prior to output. On startup, SCALX = 29.5 pixels/cm so the arguments for all routines will be in cms. However, using the WARP routine, it is possible to change this factor. When this happens, the X dimension is effectively stretched or compressed, thus distorting all figures drawn. (eg; circle become ovals and squares become rectangles)

SCALY = Y dimension scale factor. Does the same thing for the Y dimension as SCALX does for the X dimension. It is also initialized to 29.5 pixels/cm and can be changed by WARP.

IDIMX = The dimension, in pixels, of the X dimension of the currently active graphics region. Used in a number of the routines to insure that the specified arguments are within the bounds of the graphics region.

IDIMY = The dimension, in pixels, of the Y dimension of the currently active graphics region. Used in the same manner as IDIMX.

LINTYP saves the terminal's currently active linetype

```
COMMON/LINTYP/ LINOLD
```

see note 1

LINOLD = The currently active linetype of the terminal. Note that the "S" linetype has been called linetype 0 and the "P" linetype has been called linetype 9.

ONEDIM gives VECGEN information about single dimension line output

```
COMMON/ONEDIM/ IASSOC,RINC1D,STRT1D
```

see notes 2,4

When in the single dimension mode, VECGEN assumes that the other dimension conforms with the following equation;

```
     let N = the number of points to be output from VECGEN
             in the one dimension mode.
      R(I) = the points to be used in place of the unspecified
             dimension.
then R(I) = STRT1D + RINC1D*I     I = 0,1,2,...,N-1
```

note; the R(I) terms are scaled by scal* the same manner as are the parameters sent as arguments to VECGEN.

IASSOC > 0; VECGEN has been passed Y dimension values so it assumes that the X dimension follows the above linear equation.
< 0; VECGEN has been passed X dimension values so it assumes that the Y dimension follows the above linear equation.

RINC1D = Increment between the points of the unspecified dimension.

STRT1D = The starting point of the unspecified dimension.

PLOT Information used by the plotting routines.

```
COMMON/PLOT/ IFLAG,XONXMN,YONYMN,XLENG,YLENG,XMIN,XMAX,YMIN,YMAX,
# YONXMN,XONYMN
```

see notes 3,4

IFLAG = set in AUTOSC to indicate that it has preset values in the PLOT common block.
=-1; AUTOSC has set XMIN and XMAX (used in XAXIS).
=-2; AUTOSC has set YMIN and YMAX (used in YAXIS).
=-3; AUTOSC has set all of the above.

XONXMN = The X coordinate of the base of the X axis in cms from the lower left corner of the currently active graphics area.

YONYMN = The Y coordinate of the base of the Y axis in cms from the lower left corner of the currently active graphics area.

XLENG = The length, in cms, of the X axis.

YLENG = The length, in cms, of the Y axis.

XMIN = The X value which corresponds to the bottom of the X axis.

XMAX = The X value which corresponds to the top of the X axis.

YMIN = The Y value which correoponds to the bottom of the Y axis.

YMAX = The Y value which corresponds to the top of the Y axis.

YONXMN = The Y coordinate of the base of the X axis in cms from the lower left corner of the currently active graphics area.

XONYMN = The X coordinate of the base of the Y axis in cms from the lower seft corner of the currently active graphics area.

NOTES ON THE GRAPHICS COMMON BLOCKS

1) The AREA routine must be called before any of the other graphics routines are called. This is because AREA initializes the AREA common block and the LINTYP common block. The only exception to this rule is the AUTOSC-PARSET combination.

2) The parameters of the ONEDIM common block must be set prior to using VECGEN in a one dimensional mode (ITYPE=1). This setting is done in the LINS1D routine.

3) The XAXIS and YAXIS routines must be called prior to calling PLOT, PLOT1D, and/or TITLE. This is because the XAXIS and YAXIS routines set parameters in the PLOT common block.

4) When a parameters dimensions are given as cms, it is assumed that SCALX and SCALY have been set to 29.5 (pixels/cm). This is the value set in the AREA routine. If the WARP routine has been subsequently used to alter these settings, the dimension will no longer be in cms.

DK:GPHCAL.TXT to TT:

GPHLIB INTERNAL CALLED ROUTINES

(not including SYSLIB routines)

ROUTINE	CALLED ROUTINE(S)
AREA	------
AUTOSC	PARSET
BORDER	LINTYP
CURVE	FIGURE,PLINE
FIGURE	LINES,RLINE
LINES	LINTYP,VECGEN
LINS1D	LINTYP,VECGEN
LINTYP	------
PARSET	------
PLINE	RLINE
PLOT	LINES
PLOT1D	LINS1D
PNTOUT	------
POLY	FIGURE
RLINE	LINTYP,VECGEN
STRING	------
TITLE	PLINE,STRING
VECGEN	PNTOUT
WARP	------
XAXIS	PLINE,RLINE,STRING
YAXIS	PLINE,RLINE,STRING

GPHLIB INTERNAL CALLING ROUTINES

ROUTINE	CALLING ROUTINE(S)
AREA	------
AUTOSC	------
BORDER	------
CURVE	------
FIGURE	CURVE,POLY
LINES	FIGURE,PLOT
LINS1D	PLOT1D
LINTYP	BORDER,LINES,LINS1D,RLINE
PARSET	AUTOSC
PLINE	CURVE,TITLE,XAXIS,YAXIS
PLOT	------
PLOT1D	------
PNTOUT	VECGEN
POLY	------
RLINE	FIGURE,PLINE,XAXIS,YAXIS
STRING	TITLE,XAXIS,YAXIS
TITLE	------
VECGEN	LINES,LINS1D,RLINE
WARP	------
XAXIS	------
YAXIS	------

DK:GPHTST.TXT to TT:

GPHLIB TESTING ROUTINES

ARRLIT = Tests the RLINE routine. Directly calls AREA and RLINE.

ATST = Tests the effect of passing arguments to AREA that are too large. Directly calls AREA and BORDER.

ATOTST = Uses the data plotting routines to gererate a plot of a hyperbolic spiral (R*THETA=CONSTANT). Directly calls AREA, AUTOSC, PLOT, STRING, TITLE, XAXIS, and YAXIS.

BRDTST = Tests the BORDER routine. Directly calls AREA and BORDER.

FIGTST = Tests the figure generation functions. Directly calls AREA, BORDER, CURVE, POLY, STRING, and WARP.

GPHTST = Tests the "low level" graphics routines. Directly calls AREA, BORDER, LINES, PLINE, RLINE, and STRING.

LNTYPT = Tests the LINTYP routine. Directly calls LINTYP.

OVERT = Tests the buffer overflow limit on long VEC commands. Directly calls AREA, AUTOSC, PLOT1D, TITLE, XAXIS, and YAXIS.

PLT1DT = Tests the i dimensional plotting routines. Directly calls AREA, AUTOSC, PLOT1D, TITLE, XAXIS, and YAXIS.

STRTST = Tests the STRING command. Directly calls AREA and STRING.

TITTST = Tests the TITLE routine. Directly calls AREA, TITLE, XAXIS, and YAXIS.

VECTST = Tests the VECGEN routine. Directly calls AREA, VECGEN.

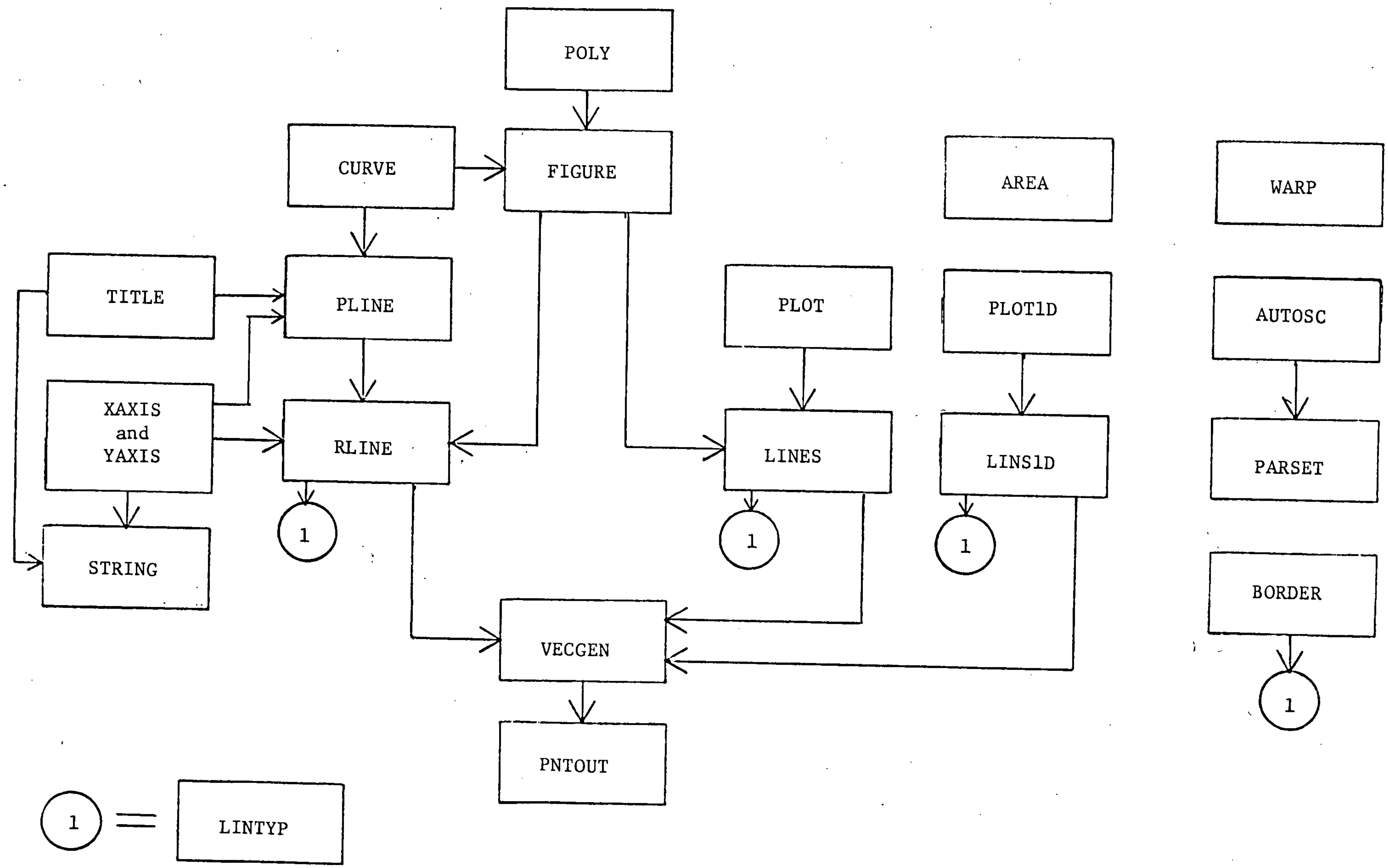
XAXTST = Tests the XAXIS routine. Directly calls AREA, XAXIS.

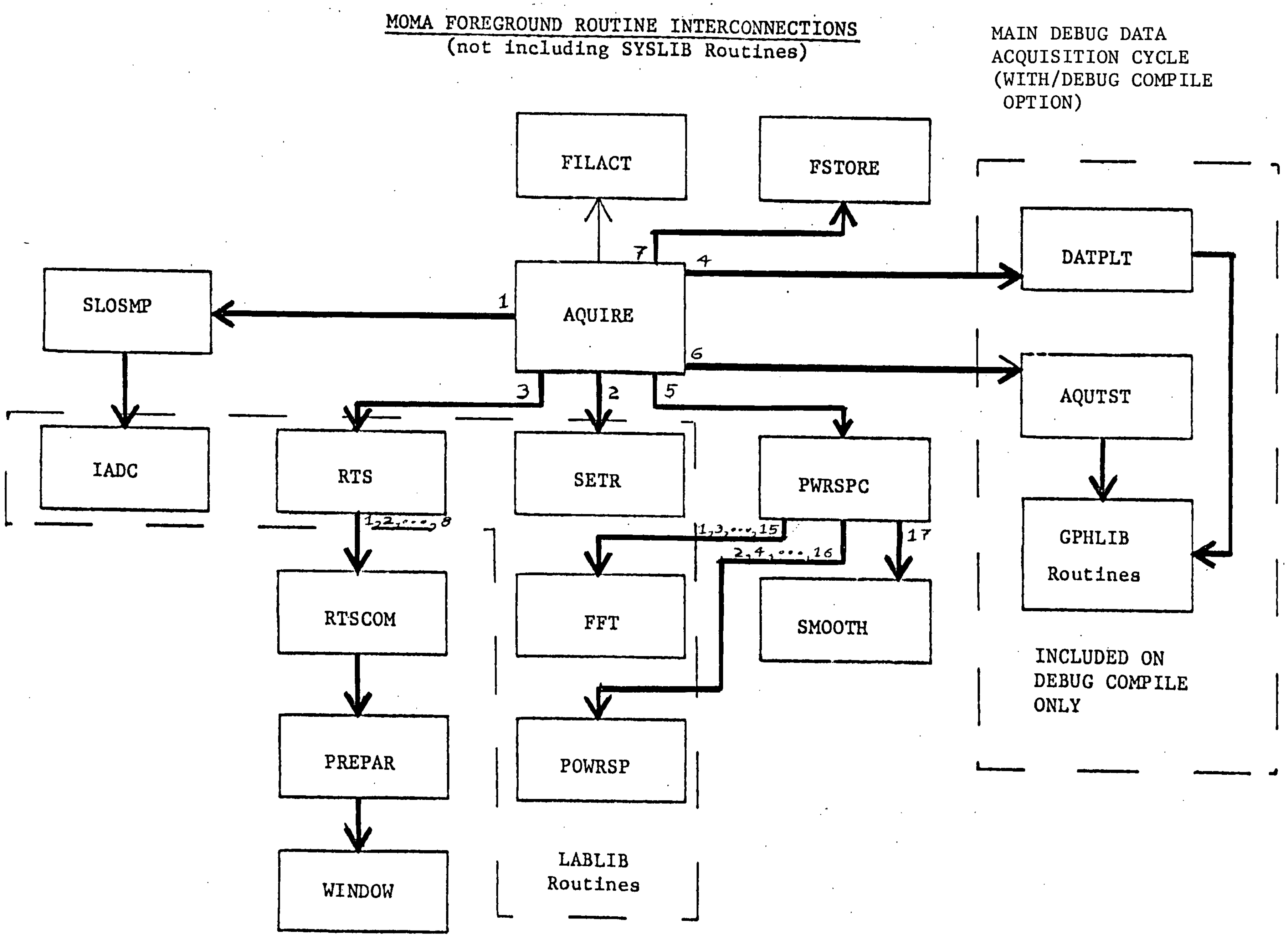
YAXTST = Tests the YAXIS routine. Directly calls AREA, YAXIS.

A2.9 Routine Interconnection Diagrams

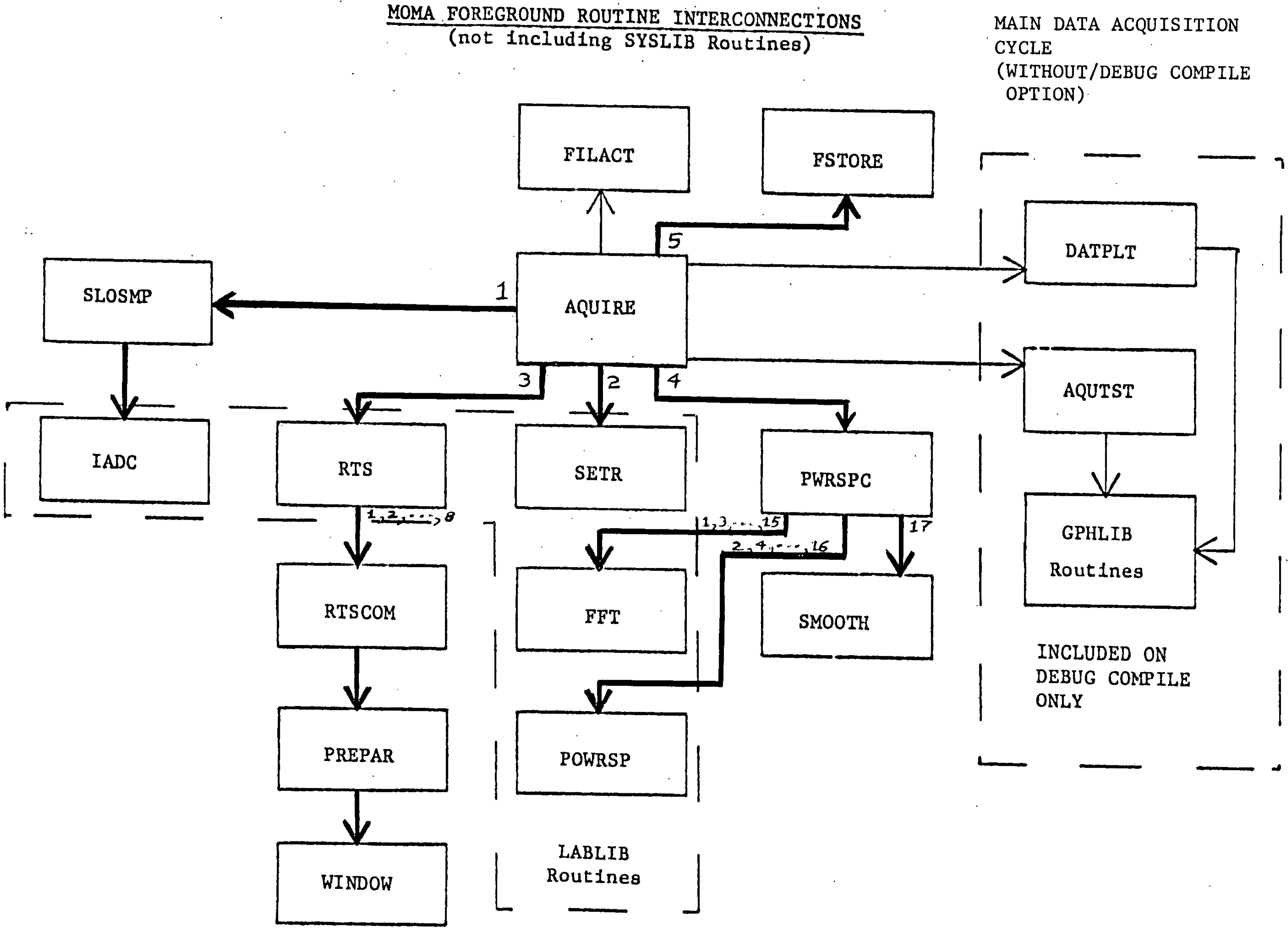
```
 Files copied:
DK:FLOW.TXT     to TT:
        MOMA ROUTINE INTERCONNECTION DIAGRAMS; DESCRIPTION
        ---------------------------------------------------

        The intent of routine interconnection diagrams is to graphically
present the flow of program control.  To produce these diagrams, the
following conventions were adopted;

     1) each block represents a given program or subprogram.
     2) the lines connecting the blocks represent paths which the flow
        of program control can follow.  These are called flow paths.
     3) Arrow on each flow path points from the calling routine to the
        called routine.

        In order to provide information about specific tasks in the system,
highlighted interconnection diagrams were produced.  THe following
conventions were addopted for these diagrams;

     1) The flow paths used during a specified task are highlighted in red.
        When photocopied, the red lines will come out as thick black lines.
     2) The numbers printed beside the highlighted paths near the calling
        routine denote the order in which the called routines are called.
        ie; if "2" is printed beside a highlighted flow path, the routine
        which that path points to is the second routine called by the
        calling routine.

Note that for MOMA background diagrams all tasks are started from DISPLA
and that for MOMA foreground diagrams all tasks are started from AQUIRE.
```

GRAPHICS ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

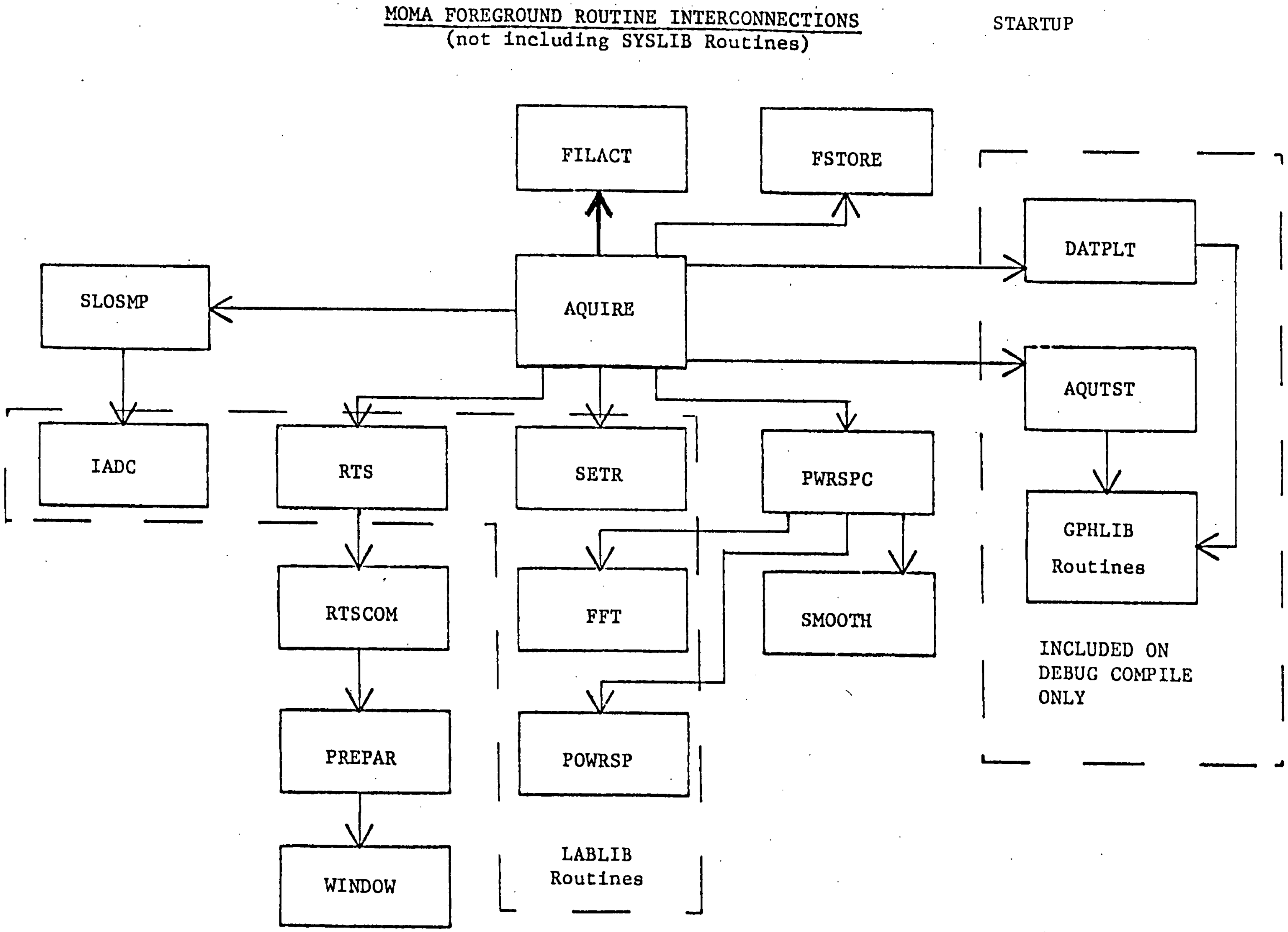



MOMA FOREGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB Routines)
MAIN DEBUG DATA
ACQUISITION CYCLE
(WITH/DEBUG COMPILE
OPTION)
FILACT
FSTORE
DATPLT
SLOSMP
AQUIRE
AQUTST
IADC
RTS
SETR
PWRSPC
GPHLIB
Routines
RTSCOM
FFT
SMOOTH
PREPAR
POWRSP
WINDOW
LABLIB
Routines
INCLUDED ON
DEBUG COMPILE
ONLY
1
2
3
4
5
6
7
1,2,...,8
1,3,...,15
2,4,...,16
17

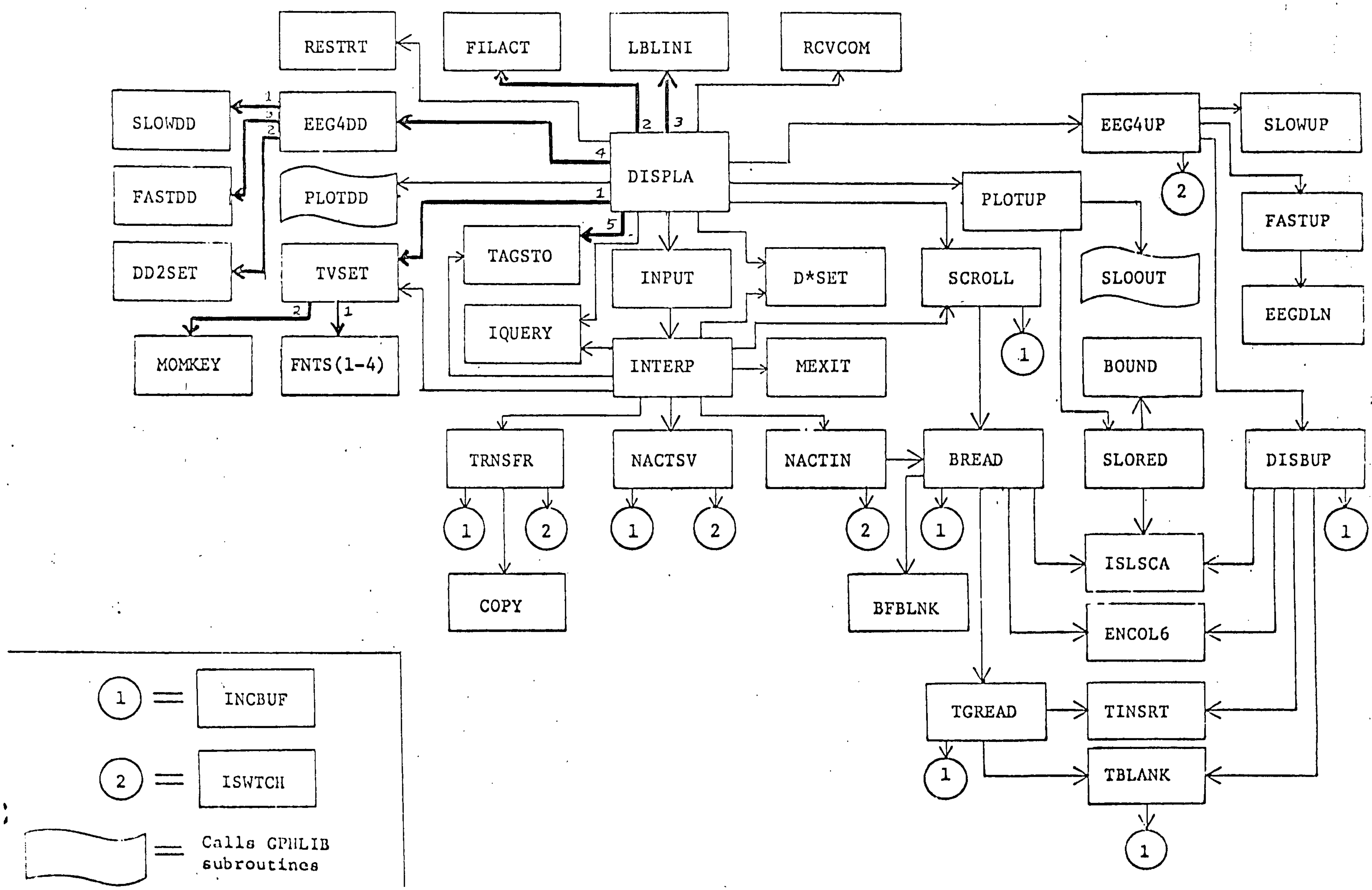
MOMA FOREGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB Routines)



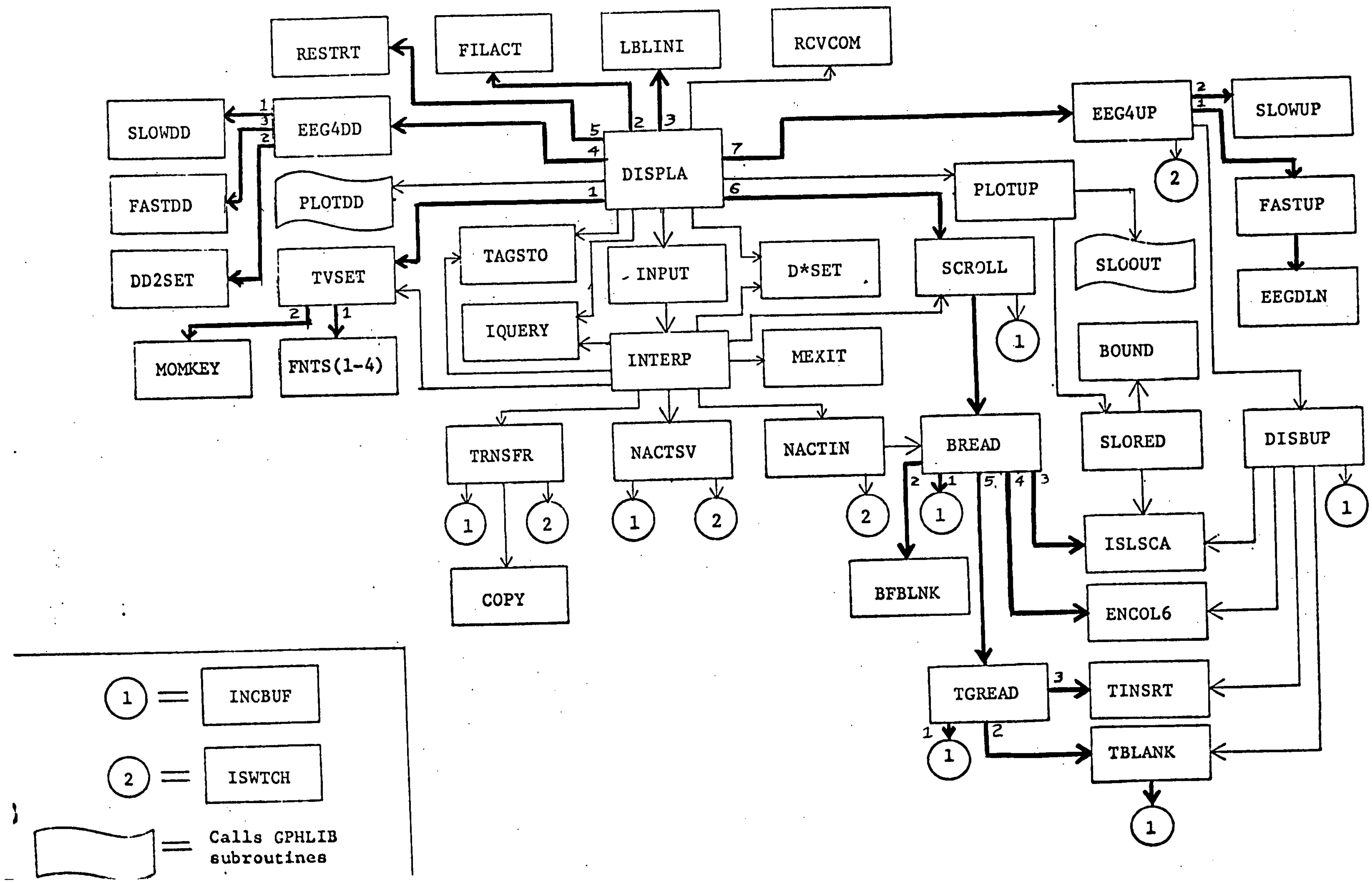
MOMA FOREGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB Routines)

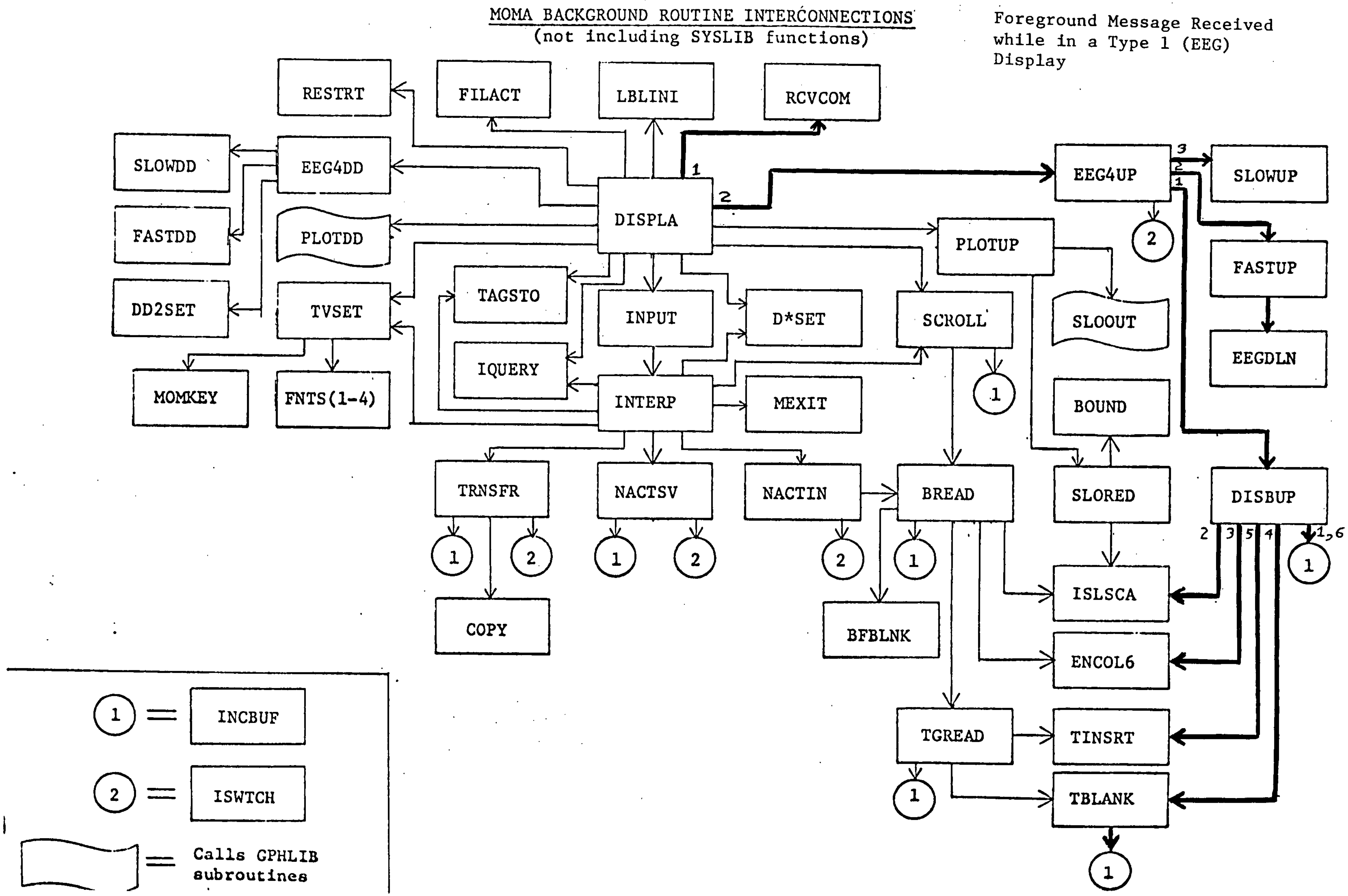


MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)



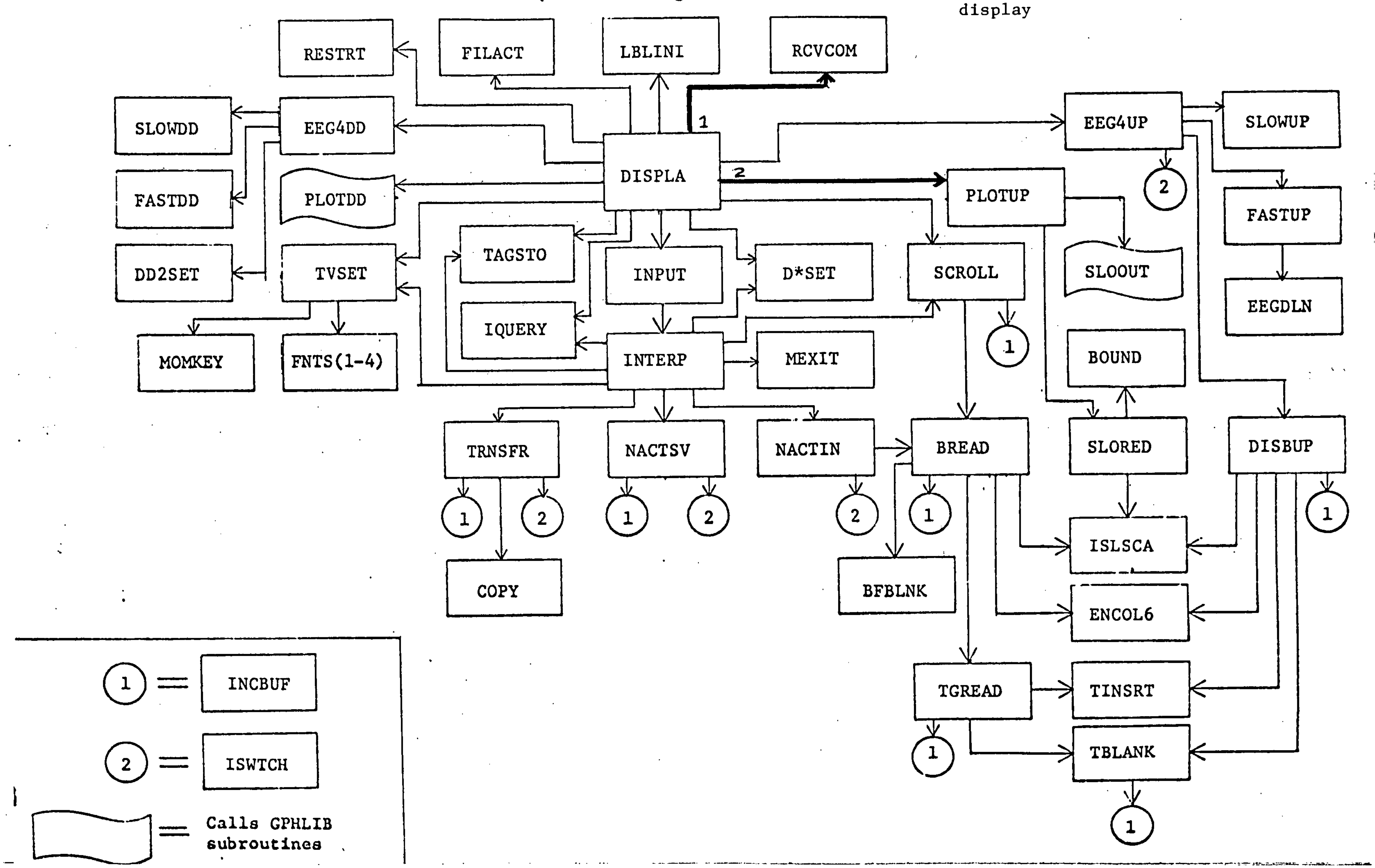
MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

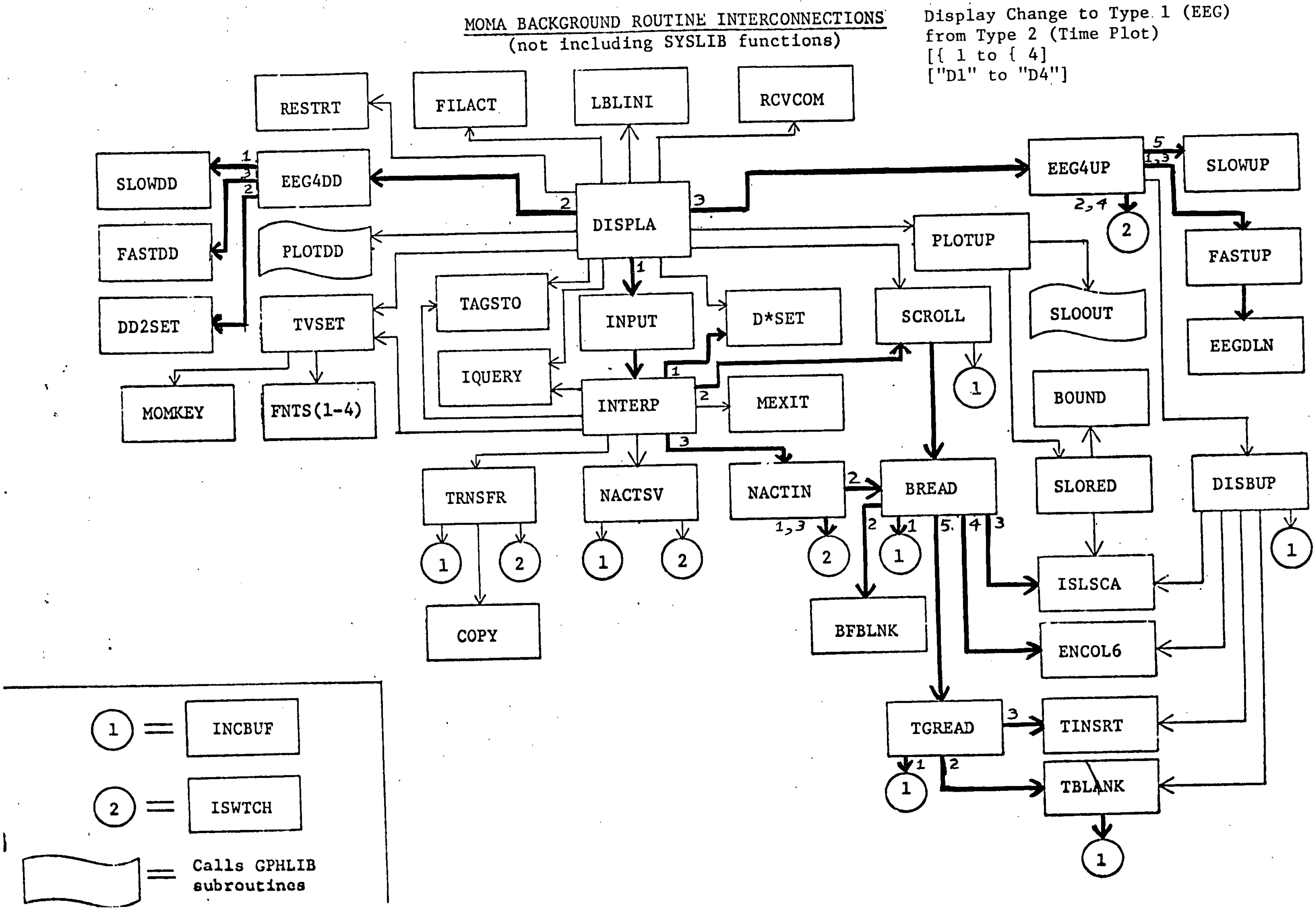



MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)
Foreground Message Received while in a Type 1 (EEG) Display
RESTRT
FILACT
LBLINI
RCVCOM
SLOWDD
EEG4DD
FASTDD
PLOTDD
DD2SET
TVSET
MOMKEY
FNTS(1-4)
DISPLA
TAGSTO
IQUERY
INPUT
INTERP
D*SET
MEXIT
EEG4UP
SLOWUP
FASTUP
EEGDLN
PLOTUP
SLOOUT
SCROLL
BOUND
TRNSFR
COPY
NACTSV
NACTIN
BREAD
BFBLNK
SLORED
DISBUP
ISLSCA
ENCOL6
TGREAD
TINSRT
TBLANK
INCBUF
ISWTCH
Calls GPHLIB subroutines

MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

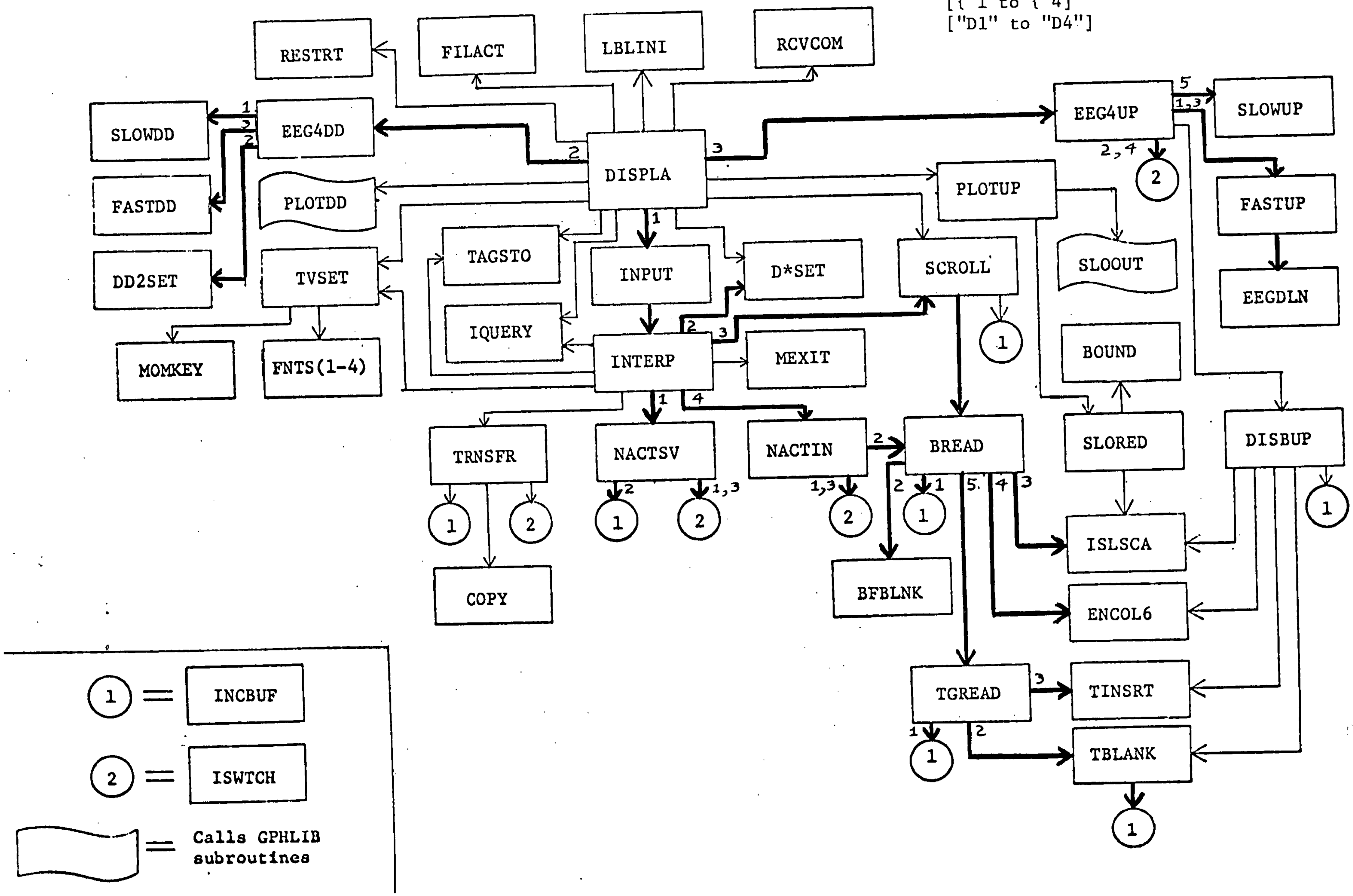
Foreground message received while in a type 2 (time plot) display

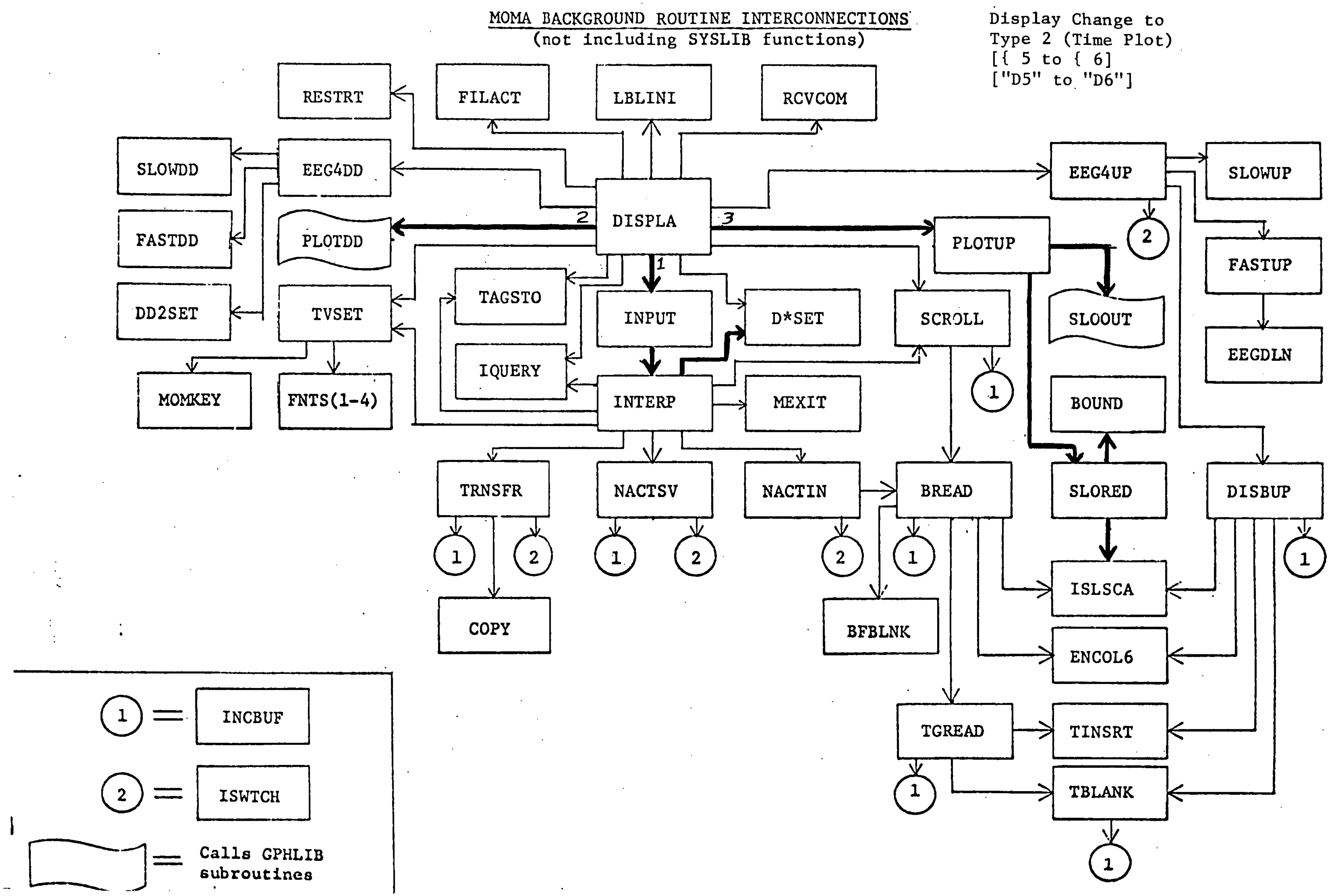



MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)
Display Change to Type 1 (EEG)
from Type 2 (Time Plot)
[{ 1 to { 4]
["D1" to "D4"]
RESTRT
FILACT
LBLINI
RCVCOM
SLOWDD
EEG4DD
DISPLA
EEG4UP
SLOWUP
FASTDD
PLOTDD
PLOTUP
FASTUP
DD2SET
TVSET
TAGSTO
INPUT
D*SET
SCROLL
SLOOUT
EEGDLN
IQUERY
INTERP
MEXIT
BOUND
MOMKEY
FNTS(1-4)
TRNSFR
NACTSV
NACTIN
BREAD
SLORED
DISBUP
COPY
BFBLNK
ISLSCA
ENCOL6
TGREAD
TINSRT
TBLANK
INCBUF
ISWTCH
Calls GPHLIB subroutines

MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

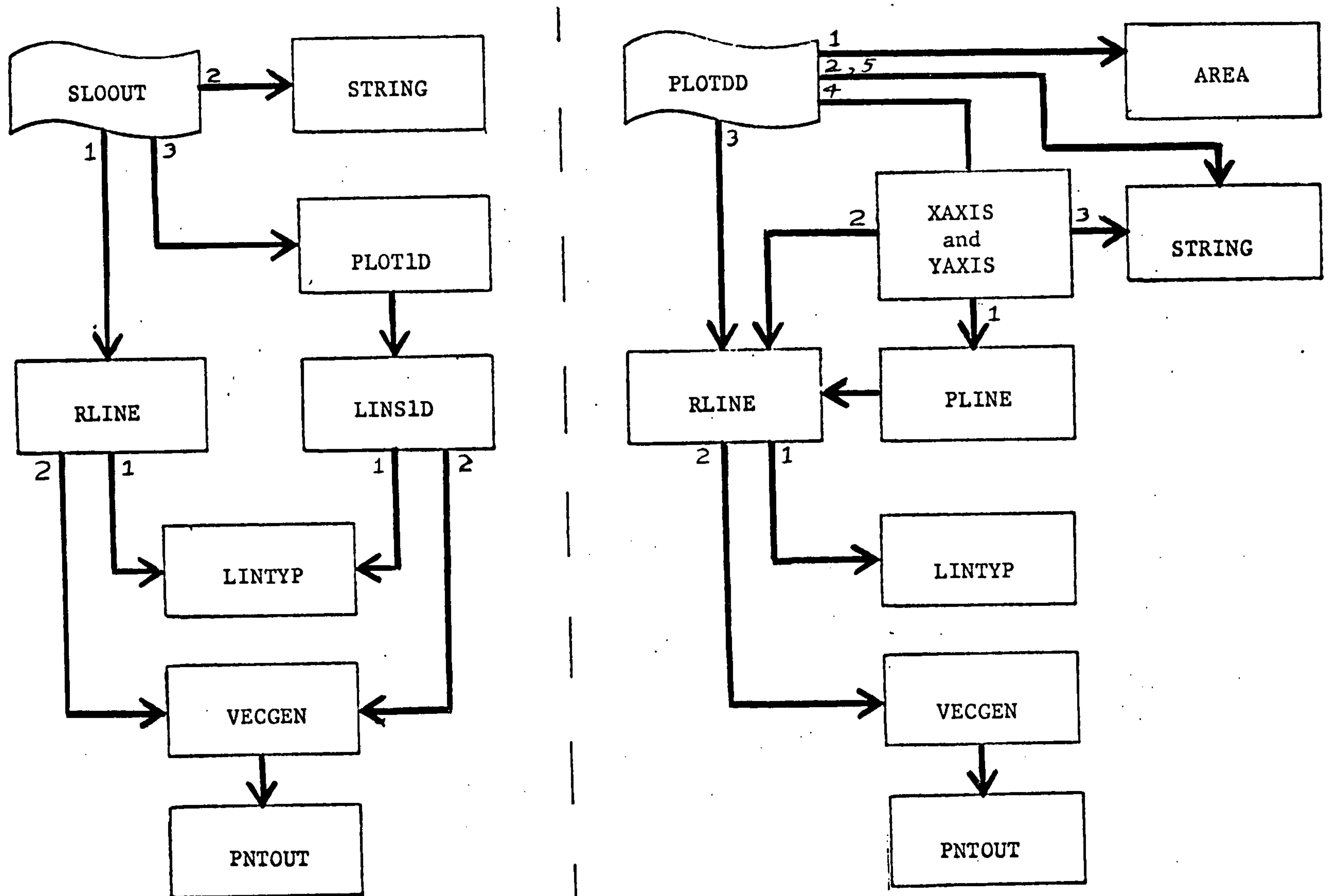
Display Change to Type 1 (EEG)
from Type 1 (EEG)
[{ 1 to { 4]
["D1" to "D4"]




MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)
Display Change to
Type 2 (Time Plot)
[{ 5 to { 6]
["D5" to "D6"]
RESTRT
FILACT
LBLINI
RCVCOM
SLOWDD
EEG4DD
DISPLA
EEG4UP
SLOWUP
FASTDD
PLOTDD
PLOTUP
FASTUP
DD2SET
TVSET
TAGSTO
INPUT
D*SET
SCROLL
SLOOUT
EEGDLN
IQUERY
MOMKEY
FNTS(1-4)
INTERP
MEXIT
BOUND
TRNSFR
NACTSV
NACTIN
BREAD
SLORED
DISBUP
ISLSCA
COPY
BFBLNK
ENCOL6
TGREAD
TINSRT
TBLANK
1 = INCBUF
2 = ISWTCH
= Calls GPHLIB subroutines

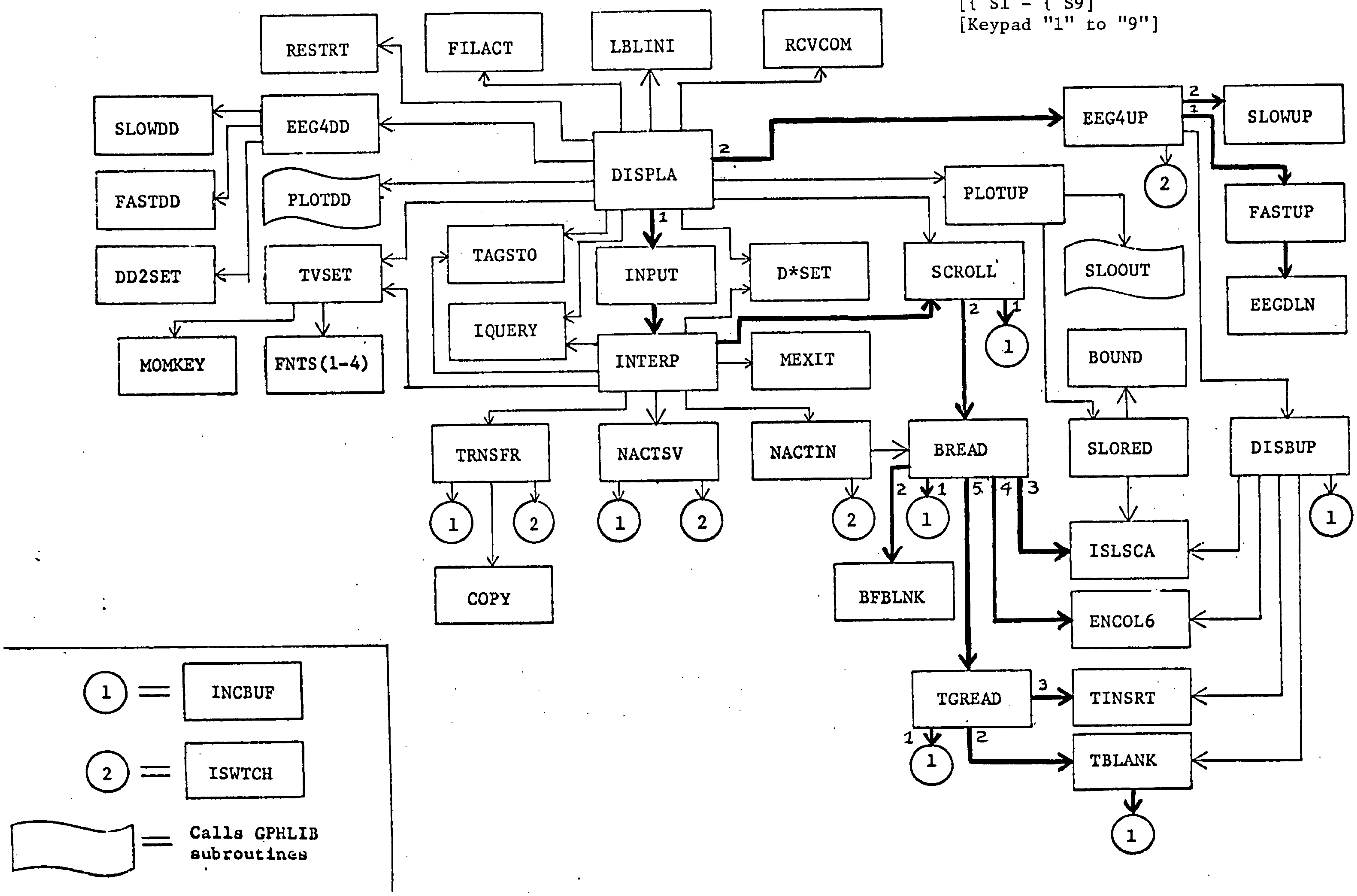
MOMA BACKGROUND GPHLIB CALLS
(not including SYSLIB functions)

2/2



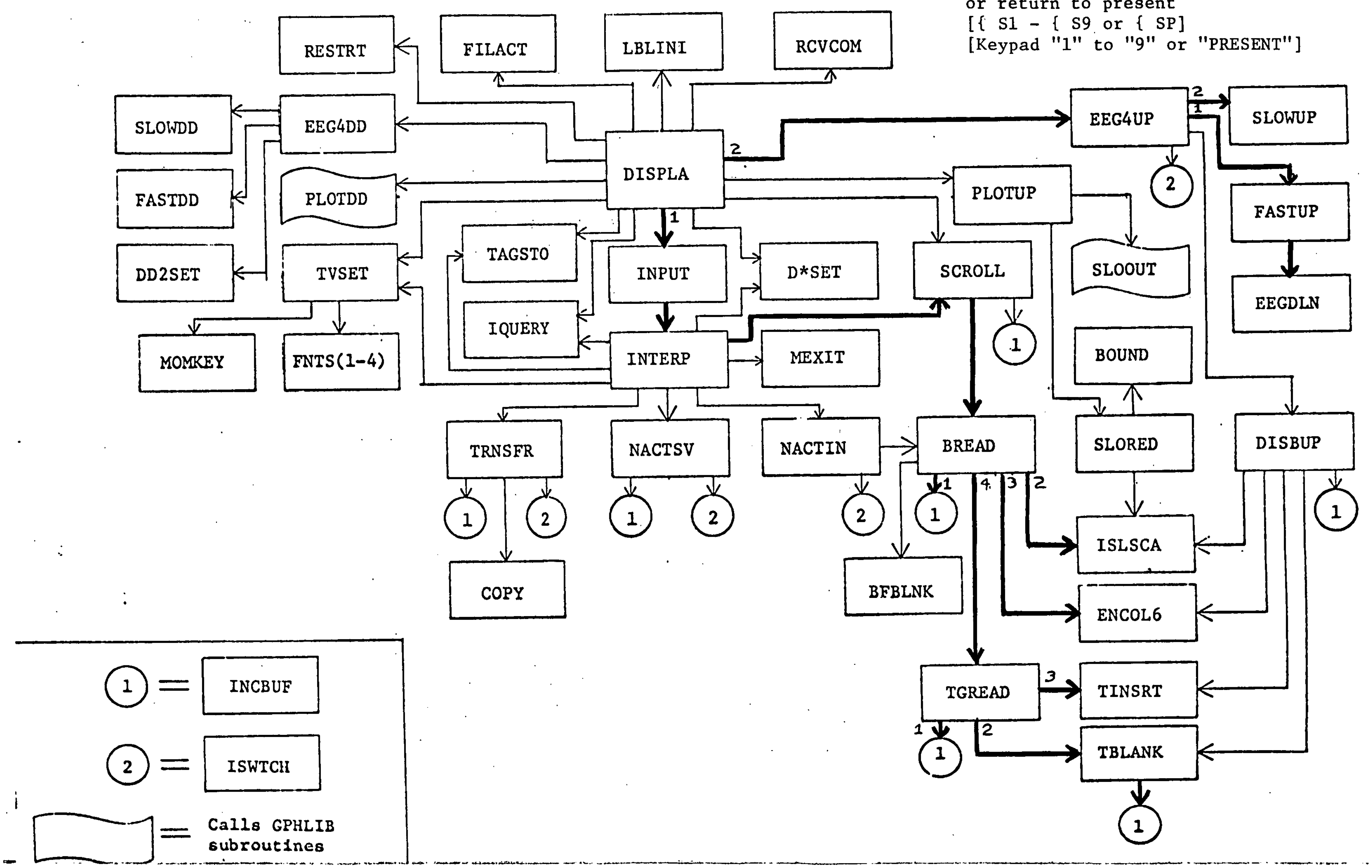
MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

Scroll Type 1 (EEG) Display
Back N records. N = 1 - 9
[{ S1 - { S9]
[Keypad "1" to "9"]

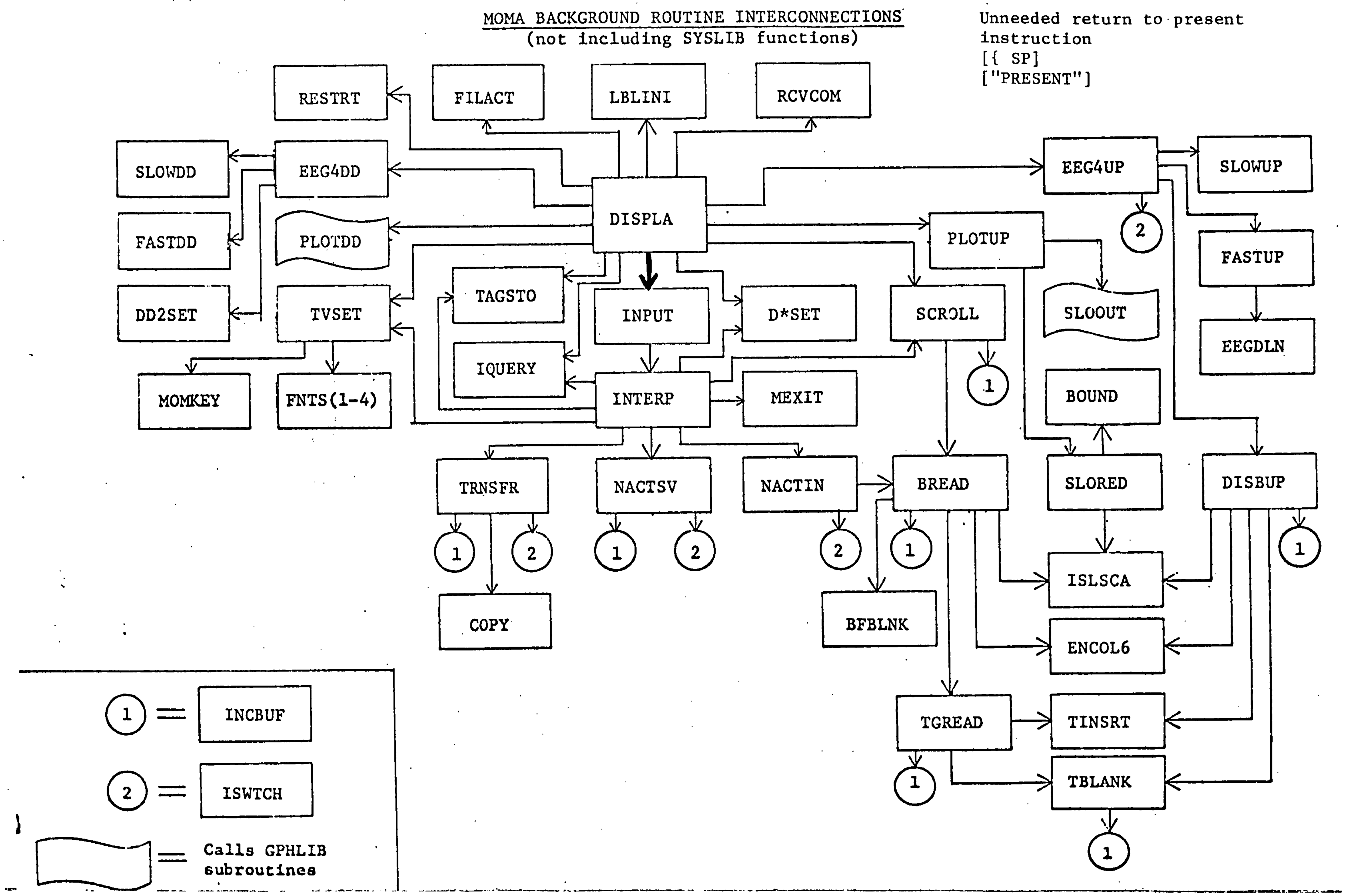


MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

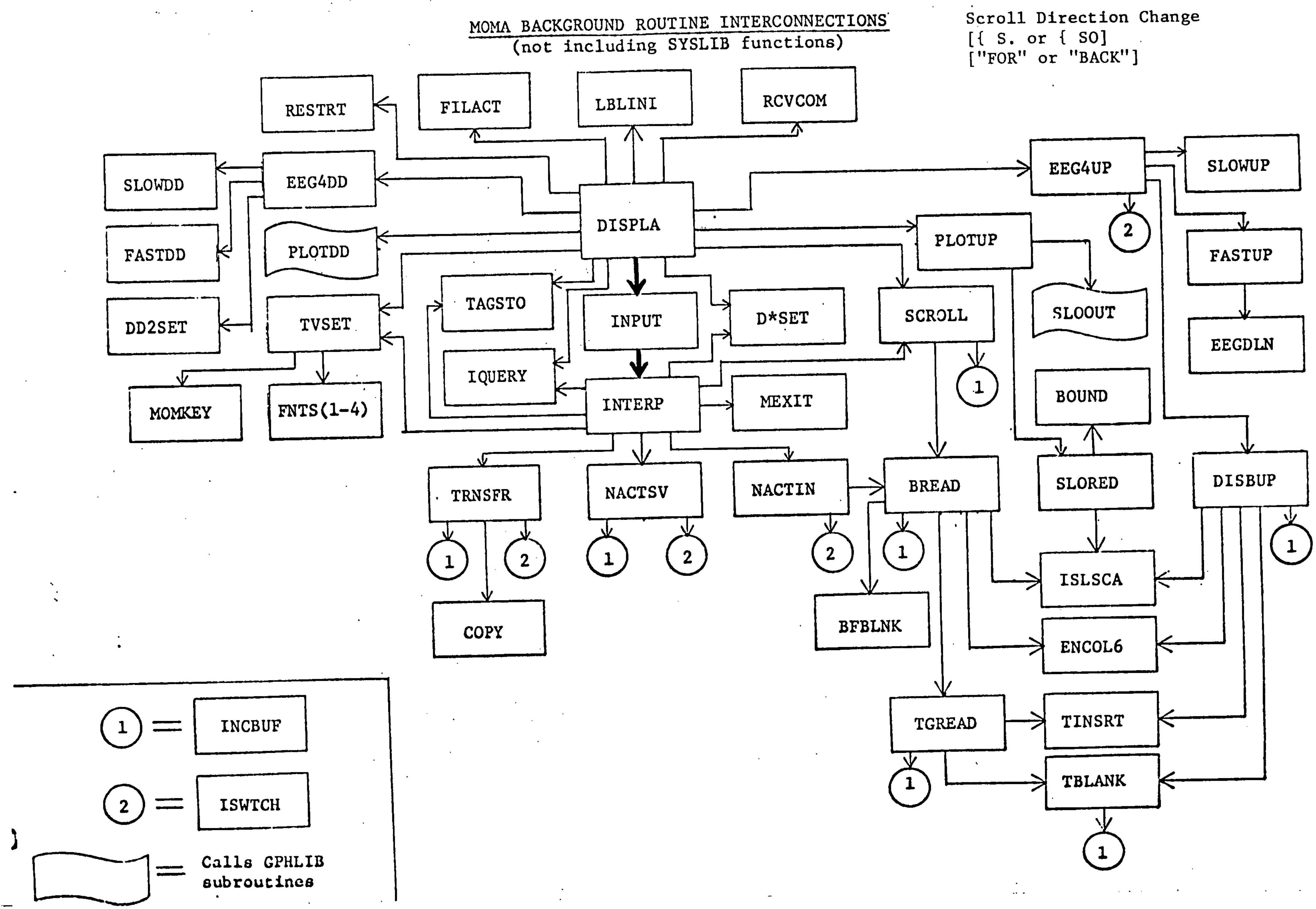
Scroll type 1 (EEG) Display
Forward N records. N = 1 - 9
or return to present
[{ S1 - { S9 or { SP]
[Keypad "1" to "9" or "PRESENT"]

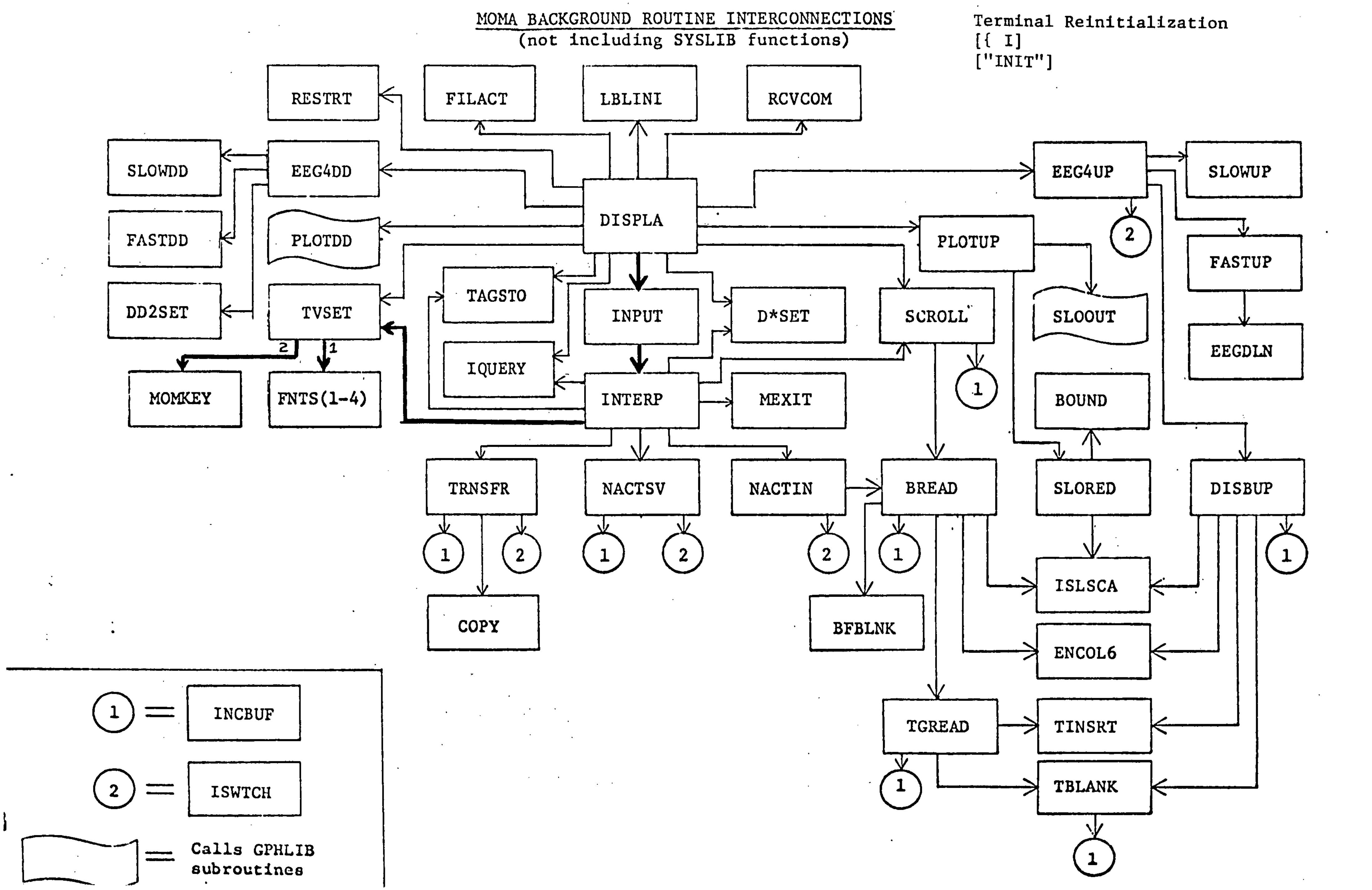


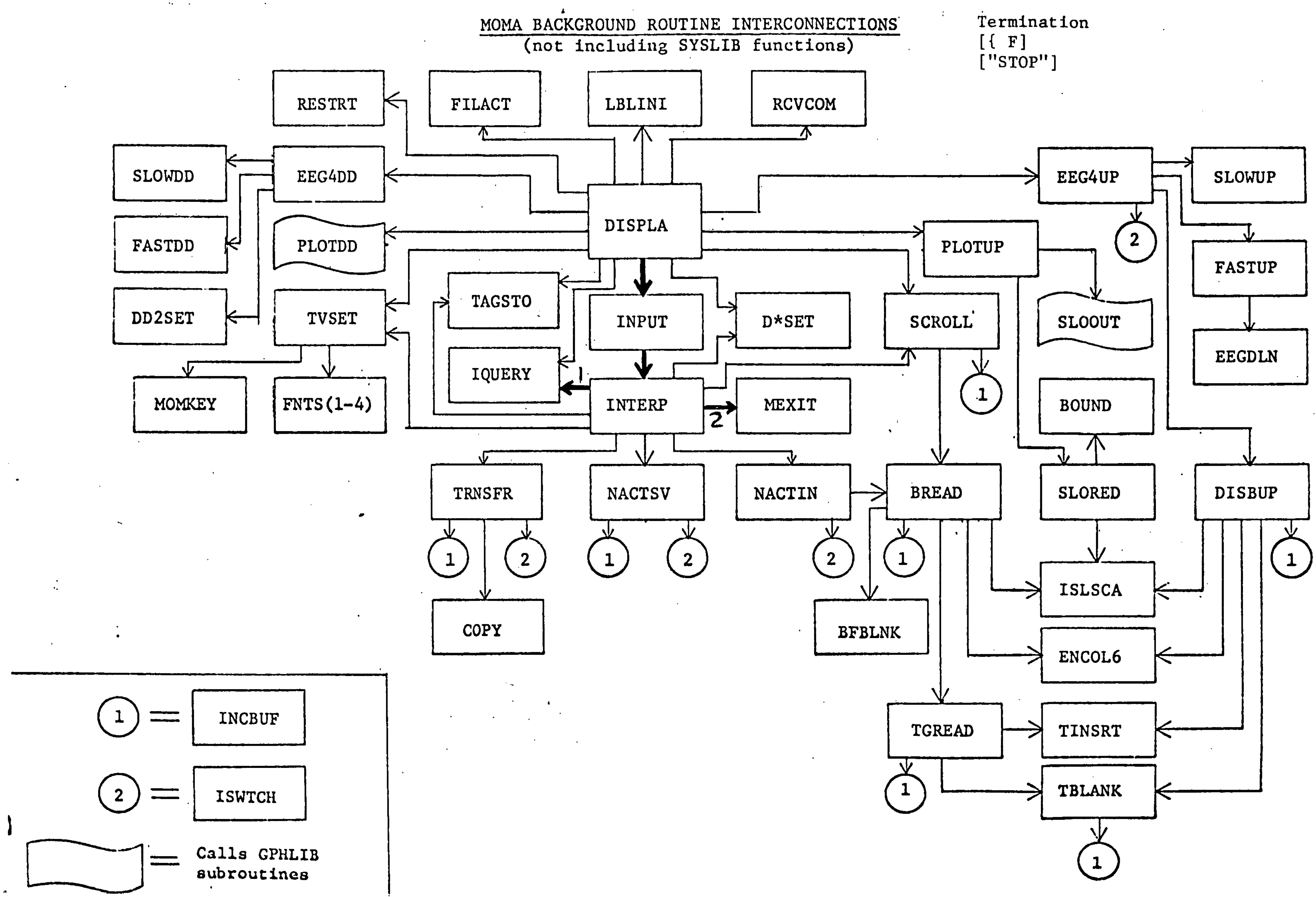
Unneeded return to present instruction
[{ SP]
["PRESENT"]



MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

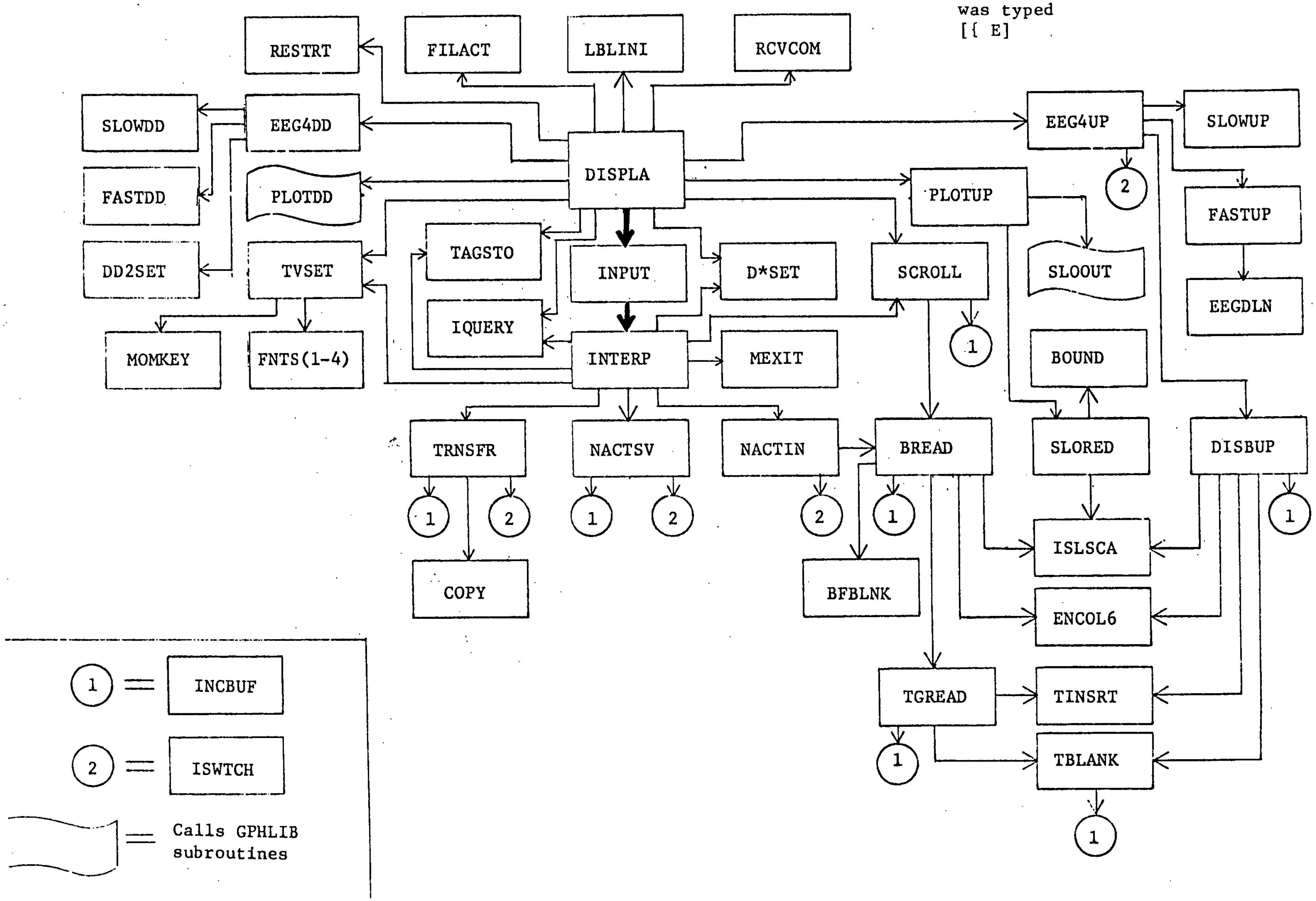



MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)
Terminal Reinitialization
[{ I]
["INIT"]
RESTRT
FILACT
LBLINI
RCVCOM
SLOWDD
EEG4DD
FASTDD
PLOTDD
DD2SET
TVSET
TAGSTO
IQUERY
DISPLA
INPUT
INTERP
D*SET
MEXIT
EEG4UP
SLOWUP
PLOTUP
SCROLL
SLOOUT
FASTUP
EEGDLN
BOUND
MOMKEY
FNTS(1-4)
TRNSFR
NACTSV
NACTIN
BREAD
SLORED
DISBUP
COPY
BFBLNK
ISLSCA
ENCOL6
TGREAD
TINSRT
TBLANK
1 = INCBUF
2 = ISWTCH
= Calls GPHLIB subroutines


MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)
Termination
[{ F]
["STOP"]
RESTRT
FILACT
LBLINI
RCVCOM
SLOWDD
EEG4DD
DISPLA
EEG4UP
SLOWUP
FASTDD
PLOTDD
PLOTUP
FASTUP
DD2SET
TVSET
TAGSTO
INPUT
D*SET
SCROLL
SLOOUT
EEGDLN
IQUERY
MOMKEY
FNTS(1-4)
INTERP
MEXIT
BOUND
TRNSFR
NACTSV
NACTIN
BREAD
SLORED
DISBUP
COPY
BFBLNK
ISLSCA
ENCOL6
TGREAD
TINSRT
TBLANK
1 = INCBUF
2 = ISWTCH
= Calls GPHLIB subroutines

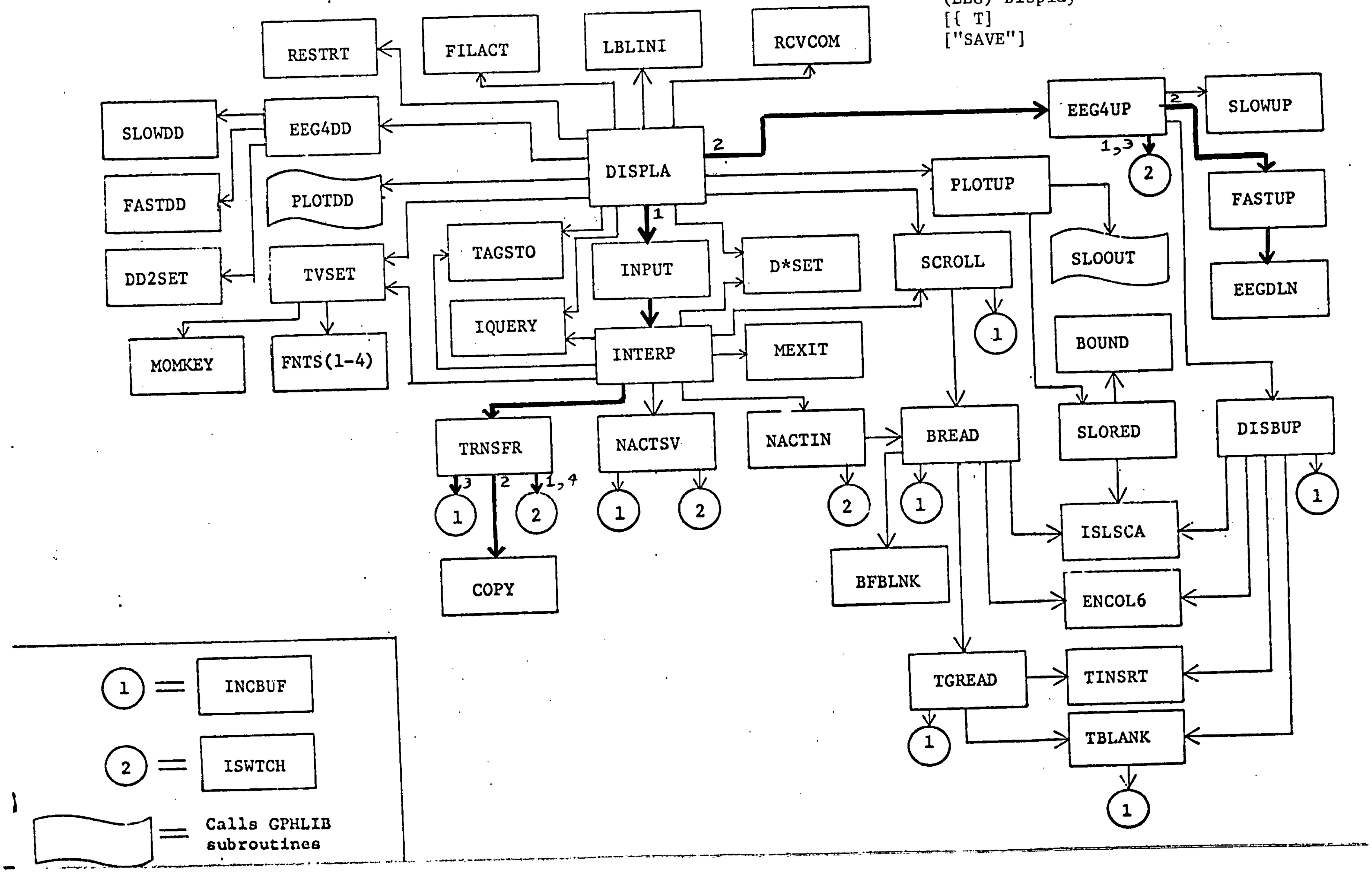
MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

Unshifted version of a shifted key instruction was typed [{ E]



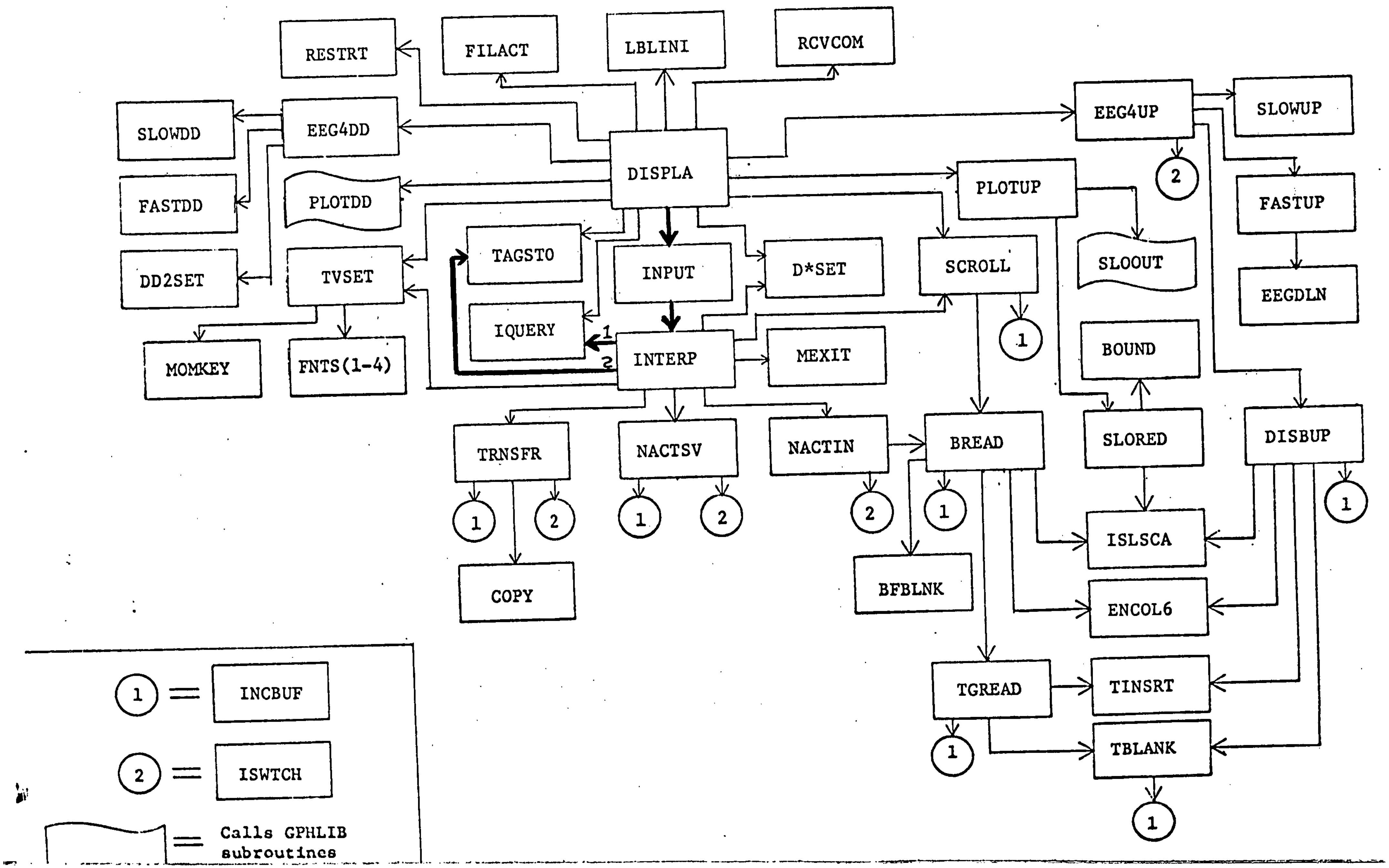
MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

Transfer Record from scrolled
to unscrolled section of type 1
(EEG) Display
[{ T]
["SAVE"]



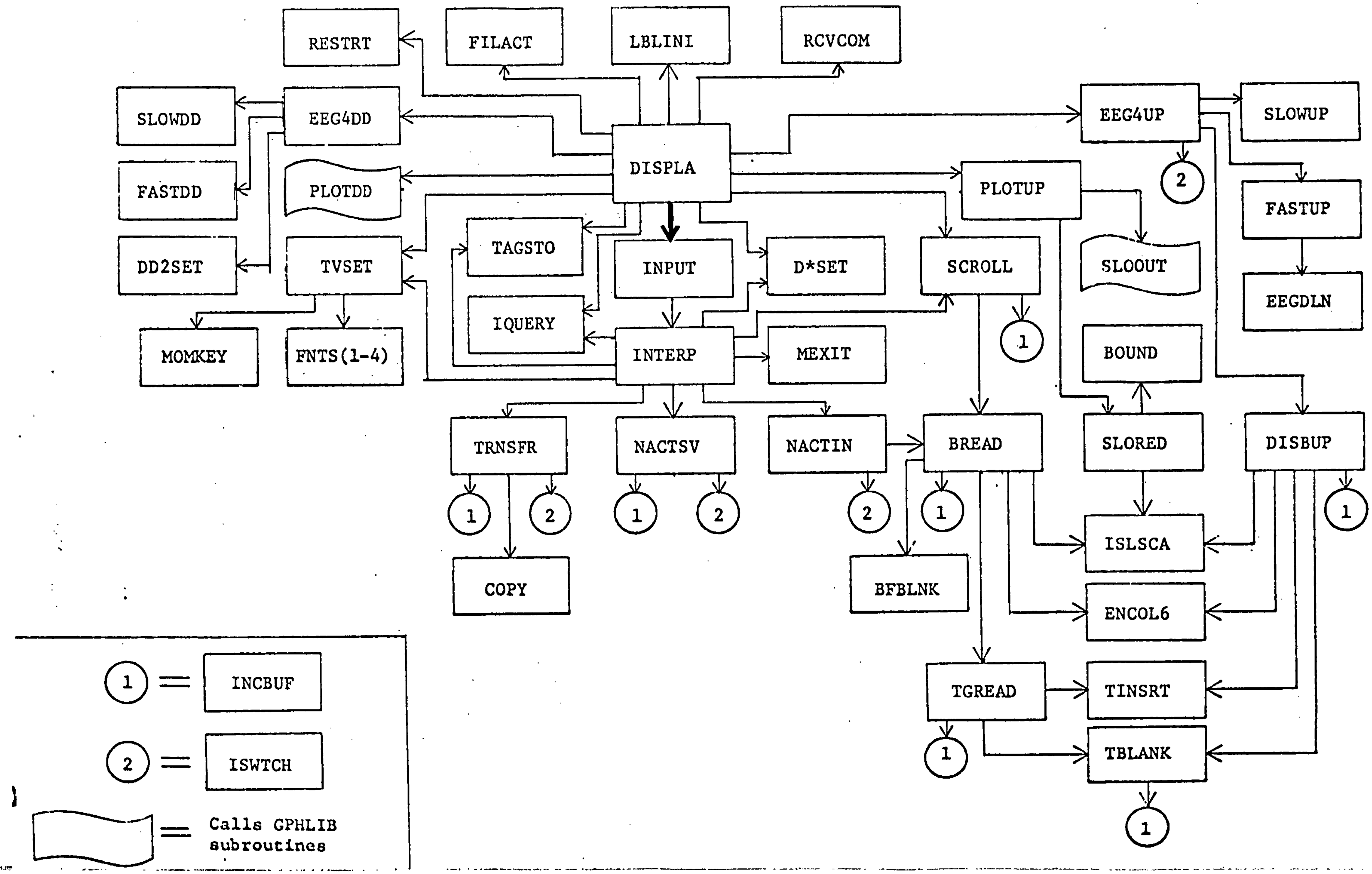
MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

Timed Tag Entry

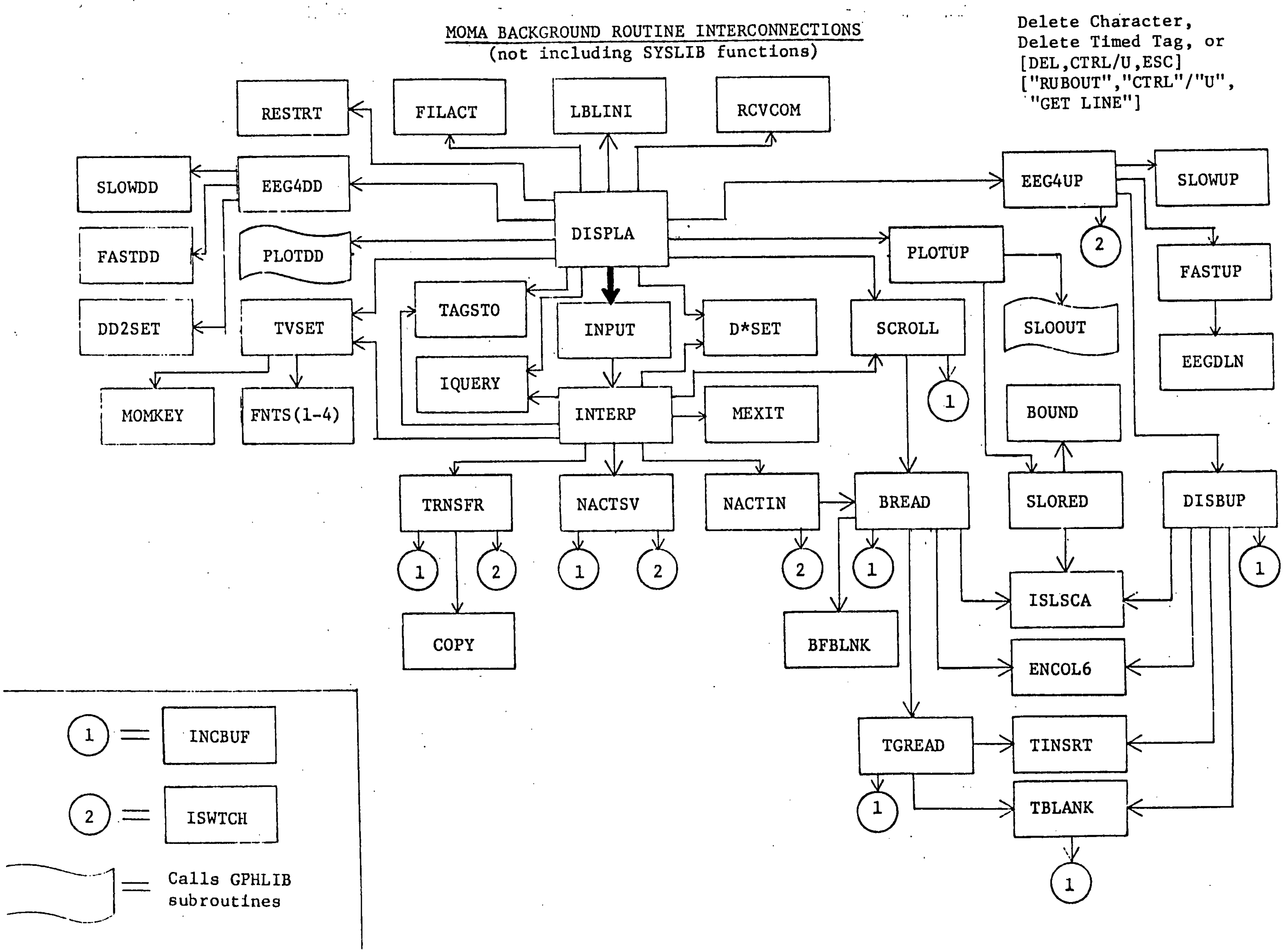


MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

Character Input

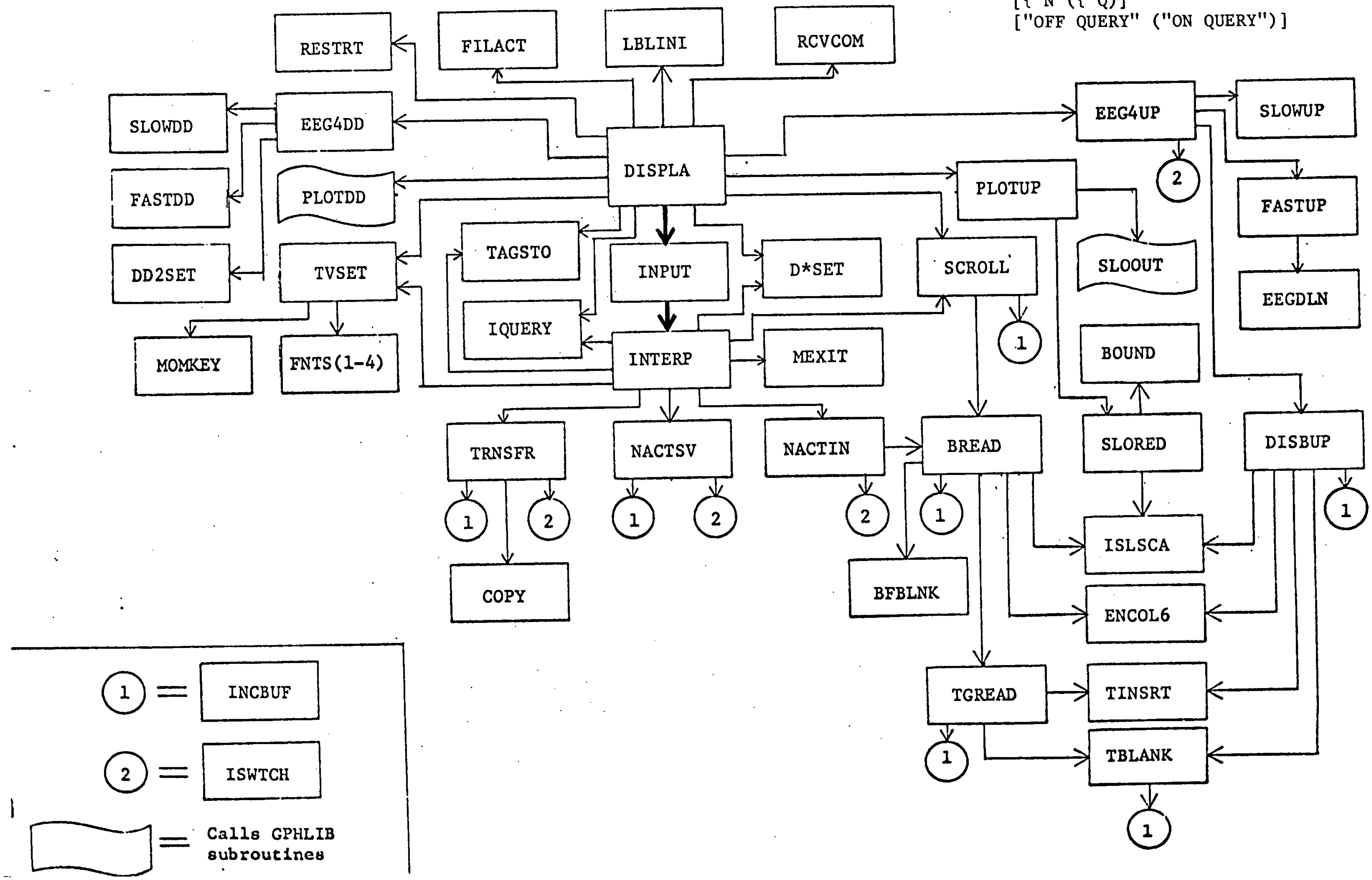


MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)



MOMA BACKGROUND ROUTINE INTERCONNECTIONS
(not including SYSLIB functions)

Deactivate (Activate)
Query on Timed Tag Entry
[{ N ({ Q)]
["OFF QUERY" ("ON QUERY")]



APPENDIX 3 MOMA SUMMARY AND USERS GUIDE

A3.1 MOMA Summary

```
TYPE SUMMOM.TXT
```

MOMA SUMMARY 1/2
------------ Sept 11/80

MOMA, or Mobile Operation Monitor for Anesthetists, is a computer-based perioperative patient monitoring system. This system can acquire, process, store, and display four channels of EEG data along with up to four channels of slowly varying non-EEG data.

A number of features have been incorporated into MOMA which should make it a useful addition to the operating room. Firstly, all critical components of the system reside within the operating room, thus simplifying startup procedures and minimizing the liklihood of external interference. Secondly, a number of display formats are accessible through a video screen so the user can choose the display which best suits him/her at a given point in time. The emphasis of these displays has been to make the trends in monitoring parameters readily apparent. Thirdly, these displays each incorporate a number of the monitoring parameters acquired by MOMA so the task of parameter comparison should be simplified. Fourthly, relevant comments can be entered to the computer at any point during the operation. These comments are then permanently stored and are incorporated into some of the displays. Fifthly, it is possible to move the time orientation of an EEG display back and forth in time. This ability to recall a display which appeared earlier in the operation should prove useful when attempting to determine long term trends in a patients state. Finally, an effort has been made to make these features easy to use. Most commands are generated by typing one or two clearly labeled predefined keys and documentation describing all phases of operation of the system has been provided.

A conclusion of a computer-based patient monitoring project performed here at VGH (EEGAL) was that presenting the EEG using a Dot Density representation of the compressed spectral array (DSA) showed promise as a patient monitoring tool. Consequently, the EEG is presented in this form in MOMA. MOMA also provides an additional feature. The DSA display for each EEG channel has been split into two independent sections. The first section, called the scrolled section, will be automatically updated throughout the operation. The other section, called the unscrolled section, will not be altered by the computer. At any point during the operation, the user can transfer a record from the top of the scrolled section to the top of the unscrolled section. Thus, when the scrolled section has been moved to a different point in time, a direct comparison between the record(s) saved in the unscrolled section and the contents of the scrolled section can be made. The DSA and the two section feature are explained more fully in the document "INTRODUCTION TO MOMA".

The focus of the work done on MOMA to this date has been to produce a framework on top of which a patient monitoring system could be developed. The features described above can be, and probably should be, altered so as to optimize MOMA's usefulness and useability. The framework will also permit the addition of new features. It is hoped that through a continuing process of device assessment and redesign, an "optimum" patient monitoring system will be produced.

The people presently involved with the design of the system have recognized a number of potential improvements to MOMA. Firstly, a computer with more memory and more processing power could speed up all phases of operation of the system and would make the system quieter. Secondly, with the incorporation of data acquisition hardware, it will be possible to elliminate the need for an EEG machine. Appropriate hardware has been ordered for both the above improvements.

Thirdly, with the addition of a printer, data can be periodically output throughout the operation. In addition to providing a hard copy backup in case of computer failure, the printer provides for the potential incorporation of a recordkeeping facility on MOMA. A number of printers are presently being considered for purchase. With the incorporation of these improvements, it is likely that the system will be repackaged. In the process of repackaging the system, things like large wheels, workspaces and drawers, standardized connector panels with noninterchangeable connectors, etc, will be included.

As a result of meetings with hospital personnel, four application areas for this system were identified. Two of these areas, (open heart surgery and carotid endarterectomies) involve the use of the system by anesthetists within the operating room. The other two areas (advanced brain rescussitation and the monitoring of head injury patients) involve the use of the device by EEG specialists outside of the operating room.

In conclusion, the system described above is the skeleton of a computer-based patient monitor. Through the involvement of many people with many areas of expertise, it is hoped that the bare bones will be be filled in in a way that will produce a clinically useful patient monitoring system.

A3.2 MOMA Users Guide

A3.2.1 Introduction to MOMA

INTRODUCTION TO MOMA

ABSTRACT

MOMA, or Mobile Operation Monitor for Anesthetists is a computer-based perioperative patient monitoring system. Four channels of EEG data are acquired and processed along with up to four channels of non-EEG data by a Digital Equipment Corporation LSI-11 based microcomputer system. This data can then be displayed on a Tektronix 4025 graphics terminal using a number of predefined display formats. The processed EEG data and the non-EEG data are also stored on a magnetic data storage medium (floppy disk).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYSTEM FEATURES

GENERAL CHARACTERISTICS

MOMA acquires and stores a number of physiological parameters. These parameters are then displayed simultaneously on a single display medium (graphics terminal). Using a multiparameter display format such as this, it should be possible to organize information in a way that will make parameter interrelationships readily apparent.

MOMA has a number of formats for displaying the information it acquires. By using simple keyed commands, the user can access any one of these displays at any point in time. This feature gives the user the luxury of choosing a display which he (she) feels best suits a particular situation. In addition, this feature provides a convenient means for directly comparing the efficacy of different display formats without necessitating the use of extra hardware.

All critical components of MOMA reside in the operating room. This will minimize the likelihood of interference with the system.from outside sources, simplify setup procedures, and allow it to be used in a wide variety of operating room layouts. At present, it is necessary that an EEG machine accompany MOMA in the operating room. However, hardware has been purchased so that it will be possible to place the entire system on a single mobile cart.

An effort has been made to minimize the time required to use the system. Towards this end, the amount of required interaction with MOMA has been minimized. When interaction is necessary, messages which describe appropriate courses of action are written on the terminal's screen. A bell is rung when these messages are sent so as to attract the user's attention.

The length of the required response is minimized, often only entailing a single key stroke.

In addition to minimizing required interaction, importance has been placed on the simplicity of use of MOMA's optional features. Thus, it is not necessary to type long command strings on the keyboard to effectively utilize the system. In fact, most of the commands can be sent by typing one or two predefined keys.

Fault tolerance is an important design consideration for patient monitoring systems. The following features have been incorporated into MOMA with this in mind. Firstly, to generate some of the commands, two keys must be pressed simultaneously. This should minimize the likelihood of accidental typing of those commands. Secondly, requests for certain critical functions such as program termination must be followed with an affirmative response to a query before they are performed. Thus the user has the opportunity to change his mind and/or to recover from an accidental request. Thirdly, if for any reason the terminal does not respond properly during the operation, it is possible to reinitialize it without stopping the program. Fourthly, if the program stops during the operation, it is possible to restart it using the standard start-up procedure. Data acquired prior to termination will not be lost. Finally, this document should serve to describe the systems features and proper use and will accompany the system within the operating room.

TIMED TAGS

Since many clincially significant events cannot easily be automatically acquired by a computer, a facility for entering relevant comments during the operation has been incorporated. These comments have been called "timed tags" and are entered through the terminal's keyboard. On entry, they will be associated with the next acquired data record, stored on floppy disk, and

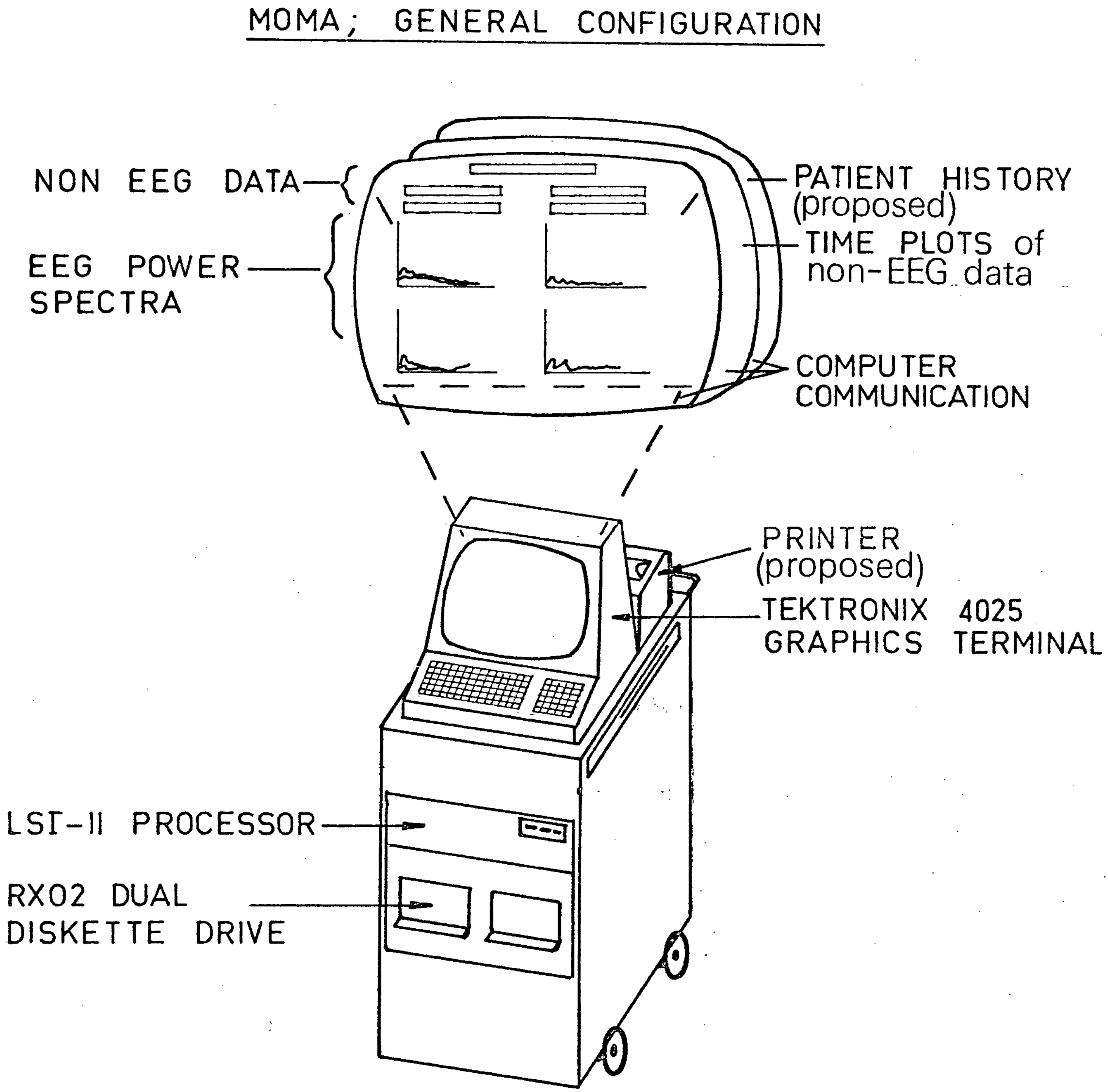


Figure 1: MOMA General Configuration

incorporated on some of the displays.

EEG REPRESENTATION

The usefulness of EEG as a routinely used monitoring parameter in the operating room has been limited by difficulties in obtaining, at a glance, clinically useful information. A density modulated compressed spectral array (DSA) has been used in MOMA to present the EEG in what is hoped to be a more useable fashion. This format has been used in other systems, including a previous patient monitoring project here at VGH (EEGAL). Details about this format will be described in more detail later in this document (display type 1).

A feature of MOMA which should further increase the useability of EEG for intraoperative patient monitoring is its two part DSA display. The DSA display for each EEG channel is separated into two sections. The information contained in one section is not affected by activity in the other. This feature can be used to perform direct comparisons of EEG records separated by long periods of time. For example, one section could be loaded with the baseline EEG and the other could be automatically updated throughout the operation. Using this scheme, it is possible to directly compare the EEG acquired during the operation with the baseline EEG. Details about this feature will be described later in this document.(display type 1).

Any set of 4 EEG channels can be used in MOMA. The choice of appropriate channels will be made according to the type of information desired. In a previous patient monitoring project here at VGH, (EEGAL) the following channels were used: F3-C3, C3-01, F4-C4, and C4-02. These channels correspond to the four quadrants of the head. To maintain compatability with this project, these channels should be used for initial trails of MOMA.

SCROLLING

It is not possible to continuously display all the EEG information acquired by MOMA during the operation in a useable format. However, using MOMA's scrolling feature, the user can access all previously acquired information. With this feature, the time orientation of and EEG display can be moved back and forth in time, thus it is possible to recall any EEG display which appeared earlier in the operation. Details about scrolling can be found later in this document (user interaction, keyboard configuration).

NON-EEG REPRESENTATION

Non-EEG data is displayed along with EEG data. However, the format of these displays (a digital display) does not effectively present the trends of the non-EEG data. For this purpose, a display type which is devoted totally to the presentation of these trends has been implemented. This type of display simply plots data magnitude vs time in a similar manner as is done manually on the anesthetic records. Details about this format can be found later in this document (display type 2).

As with the EEG channels, any set of up to four slowly varying non-EEG data signals can be acquired. However, since the inputs to MOMA's analogue to digital converter are not isolated from the computer, the sampled data signals must be isolated from the patient. Some of the monitoring devices routinely used in the operating room, such as the E for M VR6, have appropriately isolated analogue signals. As the requirements and capabilities of MOMA are more precisely defined, it is likely that special purpose hardware for the acquisition of non-EEG data will be incorporated.

FEATURE SUMMARY

- entire system within the operating room
- multiparameter displays
- quick access to a number of display formats
- simplicity of use
- fault tolerance
- timed tagging for the entrance and display of relevant comments
- two part DSA display of EEG data for long and short term trending
- scrolling of EEG displays
- displays for the trending of non-EEG data

USER INTERACTION

Input to MOMA from the user is accepted through the terminal's keyboard. This input can take 3 forms, commands, timed tags, and responses to queries from the computer. As stated previously, the format for responses is described in a message written on the terminal's screen. The following text described the use of the keyboard for the entry of commands and timed tags.

The timing of data acquisition, processing and storage is critical to the proper functioning of MOMA. These tasks have consequently been given a higher priority than user interaction, i.e., if a data processing task and a user interaction task are simultaneously requested, the data processing task will be serviced first. From the user's point of view, this will mean that there may be a time delay between the entry of a command and the servicing of that command by the computer. The delay should not be more than a few seconds and the entered request will not be altered.

COMMAND STRUCTURE

In order to minimize typing requirements, MOMA's command strings are two to three characters long. These strings always begin with the character "{" and are terminated with a carriage return "RETURN". Note that any string whose first character is "{" will be interpreted as a command, thus timed tags cannot start with that character. Table 1 summarizes the command strings and their related functions.

As will be described in the next section, a number of keys on the keyboard have been programmed to generate the command strings. It should be noted, however, that if the key programming appears to be faulty, it is still possible to generate the commands by typing the command strings. For example,

TABLE 1: MOMA COMMAND SUMMARY

KEY	COMMAND STRING	FUNCTION
SCROLLING:		
"FOR"	{S.	Set scrolling direction to forward
"BACK"	{SO	Set scrolling direction to backward
KEYPAD 1-9	{S1 to {S9	Scroll display N records
"PRESENT"	{SP	Return display to present time
TIMED TAGGING:		
ALPHANUMERIC	------	Timed tag entry
"RETURN"	------	End of timed tag
"RUBOUT"	------	Delete last character typed
CNTL/U	------	Delete partially entered timed tag
"GET LINE"	------	Reactivate last timed tag typed
"ON QUERY"	{Q	Reactivate query on timed tag entry
"OFF QUERY"	{N	Deactivate query on timed tag entry
DISPLAY CHANGE:		
"D1" to "D4"	{1 to {4	Change to EEG display
"D5" to "D6"	{5 to {6	Change to non-EEG display
SPECIAL FUNCTIONS:		
"INIT"	{I	Reinitialize terminal
"STOP"	{F	Stop MOMA programs
"SAVE"	{T	Transfer EEG record from scrolled to unscrolled section.
------	{E	Generates the message "use SHIFTED VERSION of that key" Programmed into the unshifted version of the shifted special function keys (D1-D6, INIT,STOP, SAVE).

the terminal can be reinitialized by typing "{I<RETURN>" if the programming of the "SAVE" key appears to be faulty.

KEYBOARD CONFIGURATION

The terminals programmable key feature has been used to free the user from the task of typing command strings. Using the configuration shown in Figure 2, any command can be generated by the typing of a single key. Additionally, the keys have been grouped according to their function on various sections of the keyboard to simplify the task of locating them.

All the special purpose keys except CTRL/U have been labelled. In the keypad and timed tagging sections, these labels are written on top of the key. In the display and special function section, these labels are placed on a keyboard overlay which can be placed on the keyboard prior to the operation (see Figure 2). If the overlay label is above the key, the shift key must be held down while that key is pressed. If the label is below the key, the shift key is not held down.

The terminal is initialized when the system is started. During this procedure, the internal modes of the terminal are set, the keys are programmed, and a character font for the DSA display is initialized. If at any point during the operation, these settings appear to have been altered, it is possible to reinitialize the terminal without disturbing the other MOMA functions using the "INIT" key. Displays will NOT be erased during this procedure. Note that if the key programming is faulty, the command string ("{I<RETURN>") may have to be entered.

The scrolling feature uses the keypad portion of the keyboard. Through this pad, the user can change the direction of scrolling by typing the "FOR" or the "BACK" keys. The actual scrolling is performed by the keypad numeric keys from 1 to 9. These keys will scroll the EEG display

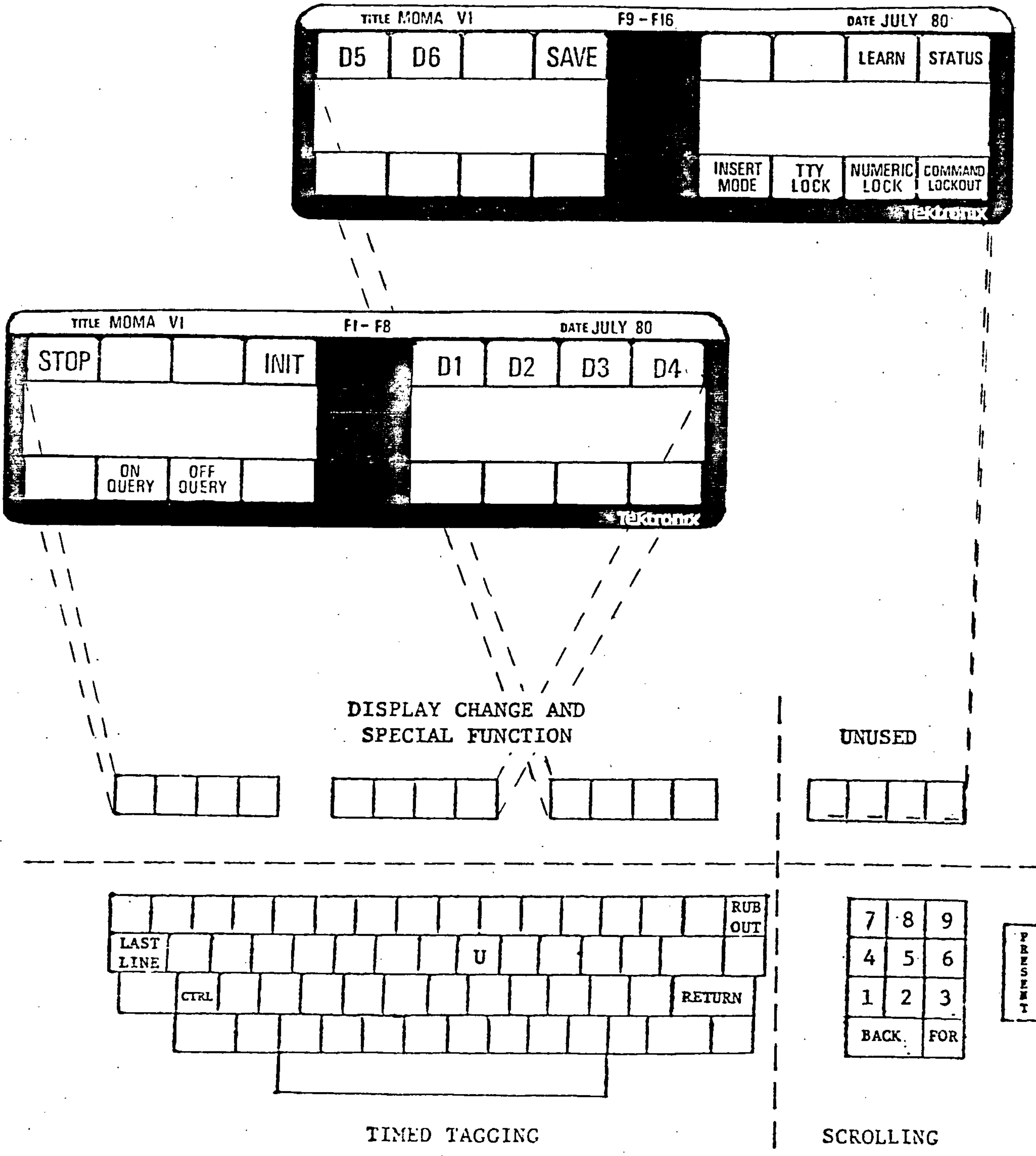


Figure 2: MOMA Keyboard Organization

1 to 9 records forward or backward in time, depending on the direction last chosen. While scrolled, the display is not automatically updated. When the "PRESENT" key is typed, the display will be returned to present time, automatic display update will be enabled, and the scrolling direction will be set to "BACK". Any attempt to scroll into the future or before the start of the operation will simply be ignored.

There are six displays presently implemented on MOMA. The keys which access these displays are labeled D1 to D6 on the keyboard overlays. Note that the "SHIFT" key must be held down while typing these keys. A description of the characteristics of these displays is presented later in this document (display type 1 and display type 2).

As has been mentioned previously, the EEG display from each channel is separated into two independent sections. The key labeled "SAVE" on the keyboard overlays allows the user to take a record from the top of one section (scrolled section) and insert it to the top of the other section (unscrolled section). At present, the unscrolled section will not be disturbed by activity in the scrolled section and cannot be loaded without using the "SAVE" function. Thus, by saving records in the unscrolled section and moving the scrolled section to another point in time, records separated by long time intervals can be directly compared. Finally, saved records are NOT lost on display change, i.e., a record saved while in one EEG display will appear in all other EEG displays and will not be lost if a non-EEG display is temporarily referenced. Note that the "SHIFT" key must be held down while the "SAVE" key is pressed. Details about the scrolled and unscrolled section can be found later in this document.(type 1 displays).

MOMA is stopped using the key labeled "STOP" on the keyboard overlays. The user is asked to confirm that the stop request is valid before the programs are terminated, so it is possible to recover from an accidental request. The writing of a period on the terminal's screen indicates the completion of this function. At this point the keyboard monitor is running and standard RT-11 instructions can be entered. If you are unfamiliar with the RT-11 operating system and/or you are finished with MOMA, you can now remove the disks from the RX02 drives, insert them in their covers, and turn off the computer's power switch. Note that the "SHIFT" key must be held down while the "STOP" key is pressed.

The keys allocated to a number of the above instructions are implemented on a shifted basis (D1-D6,SAVE,INIT,STOP). This was done to minimize the likelihood of accidental typing of those instructions. If the unshifted version of any of these keys is typed, a bell is rung and a message which reminds the user to use the "SHIFT" key is typed onto the terminal.

TIMED TAGS

Timed tags are entered through the alphanumeric portion of the keyboard by typing an appropriate comment followed with "RETURN". The only restriction on the form of these comments is that they cannot start with the character "{" and they cannot contain the character "`". The entered timed tag will be truncated to 32 characters before storage.

After the timed tag is entered, the program will rewrite the line it received and ask the user to accept of reject it. If desired, the user can suppress this query by typing the key labeled "OFF QUERY" on the keyboard overlays. Timed tags will now be stored immediately on the typing of "RETURN". Query reactivation is performed using the "ON QUERY" key.

Three instructions have been implemented to aid in the manipulation of timed tags. Firstly, the "RUBOUT" key will delete the last character typed, thus permitting the correction of typing errors. Secondly, entering CTRL/U (holding down the "CTRL" key while typing the "U" key) will delete the currently active (partially completed) timed tag. Finally, the "LAST LINE" key will reactivate the last timed tag entered.

UNUSED KEYS

A number of the keys on the terminal's keyboard are not used in MOMA. In the alphanumeric and special function sections of the keyboard, these keys will have blank key caps and the typing of them will have no effect on MOMA.

The "BREAK" key (lower right corner of the alphanumeric section) should not be typed while the computer is running. Although typing it once will have no effect, typing it twice in rapid succession may stop the computer. Should this ever happen, one can restart the computer using the standard start-up procedure.

The lighted keys on the upper right corner of the keyboard are generally not used in MOMA. In fact, the NUMERIC LOCK and the COMMAND LOCKOUT keys must never be lighted if MOMA is to function properly. If either one of these keys is turned on, depress it once to turn it off. The "TTY LOCK" key may be found to be useful for some situations. When this key is lighted, all characters sent to the computer from the terminal are capitalized. This will not impair the computer's interpretation of commands. The "INSERT MODE" key will have no effect on MOMA.

TERMINAL SCREEN'S CONFIGURATION (see Figure 3)

The display medium for MOMA is a Tektronix 4025 raster scanned graphics terminal. As shown in Figure 3, the terminal's screen has been split up into 2 regions; the workspace scroll and the monitor scroll. The information contained in one scroll is independent of the information in the other scroll.

MOMA uses the workspace scroll for the display of relevant information. During start-up, this region will be loaded with checklists and start-up instructions. During the operation, the acquired data will be displayed in this region. For the displays presently implemented, this region will use the top 30 lines of the terminal's screen.

MOMA uses the monitor scroll for interaction with the user. When the computer sends a request for servicing, the message is typed into this region. These requests are accompanied by a bell to attract the user's attention. When the user enters an instruction or a timed tag, that entry is echoed in this region. For the displays presently impelemented, this region will use the bottom 4 lines of the terminal's screen.

DISPLAY FORMATS

At present there is six different displays implemented on MOMA. Four displays are devoted to the presentation of EEG data (type 1) and the remaining two are devoted to the presentation of non-EEG data (type 2).

The programs in this system are written in a way that should make the alteration of existing displays and/or the addition of new displays a simple matter. Through a continued process of display assessment and design it is hoped that an "optimum" format for information presentation will be produced.

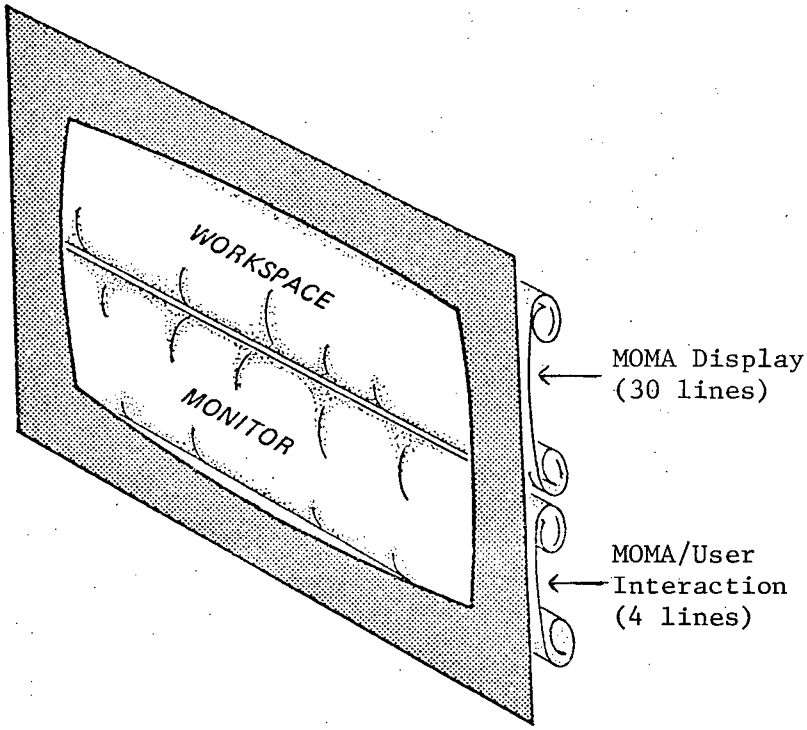


Figure 3: Terminal Screen's Configuration

DISPLAY TYPE 1 (display D1 to D4) (see Figure 4)

EEG

For type 1 displays, data from each channel of EEG can be displayed using a density modulated compressed spectral array (DSA) format. In this format, the intensity of a given character represents the magnitude of the EEG power spectrum at a given frequency and time (i.e., a high intensity character means that the EEG contained a large amount of that frequency at that time).

The horizontal axis of the EEG displays represents the frequency of the EEG signal. As the scaling numbers indicate, frequencies from 0 to 31 Hz (cycles per second) are represented.

The vertical axis of the EEG displays represents the time at which the EEG data on that row was acquired. A new row is added to the top of the display every 30 seconds throughout the operation. For convenience, the Hour and Minute of data acquisition are printed beside the vertical axis.

The display has been split into two sections, a scrolled and an unscrolled section. The following two paragraphs describe the characteristics of these sections.

The scrolled section can be actively changed by both the program and the user. When the display is presenting current information, this section is automatically updated as each new data record is acquired (every 30 seconds). Using the scrolling feature, the user can back up and recall a display which was presented earlier in the operation. The display will not be automatically updated until it is returned to present time. In this section, the time of data acquisition is printed beside the vertical axis on every second row. This section takes up the top rows of the display.

The unscrolled section can only be altered at the direction of the user. Using the record transfer function (the "SAVE" key) the user can take data from the top of the scrolled section and insert it to the top of the unscrolled section. This data will not be disturbed by future display update and scrolling activity. This ability to save a record should prove useful in performing direct comparisons between data records separated by long time internals. In this section, the time of data acquisition is printed beside the vertical axis on every row. This section takes up the bottom rows of the display.

NON-EEG

The acquired non-EEG data will be displayed digitally at the top of type 1 displays. These numbers will always be associated with the EEG record at the top of the scrolled section. They will be automatically updated when the scrolled section is updated and will change when the user uses the display scrolling feature.

TIMED TAGS

At any point during the operation the user can enter comments into MOMA via the terminal's keyboard. These comments will then be associated with the next acquired data record and stored on disk. In type 1 displays, these tags are printed to the right of the EEG scrolls.

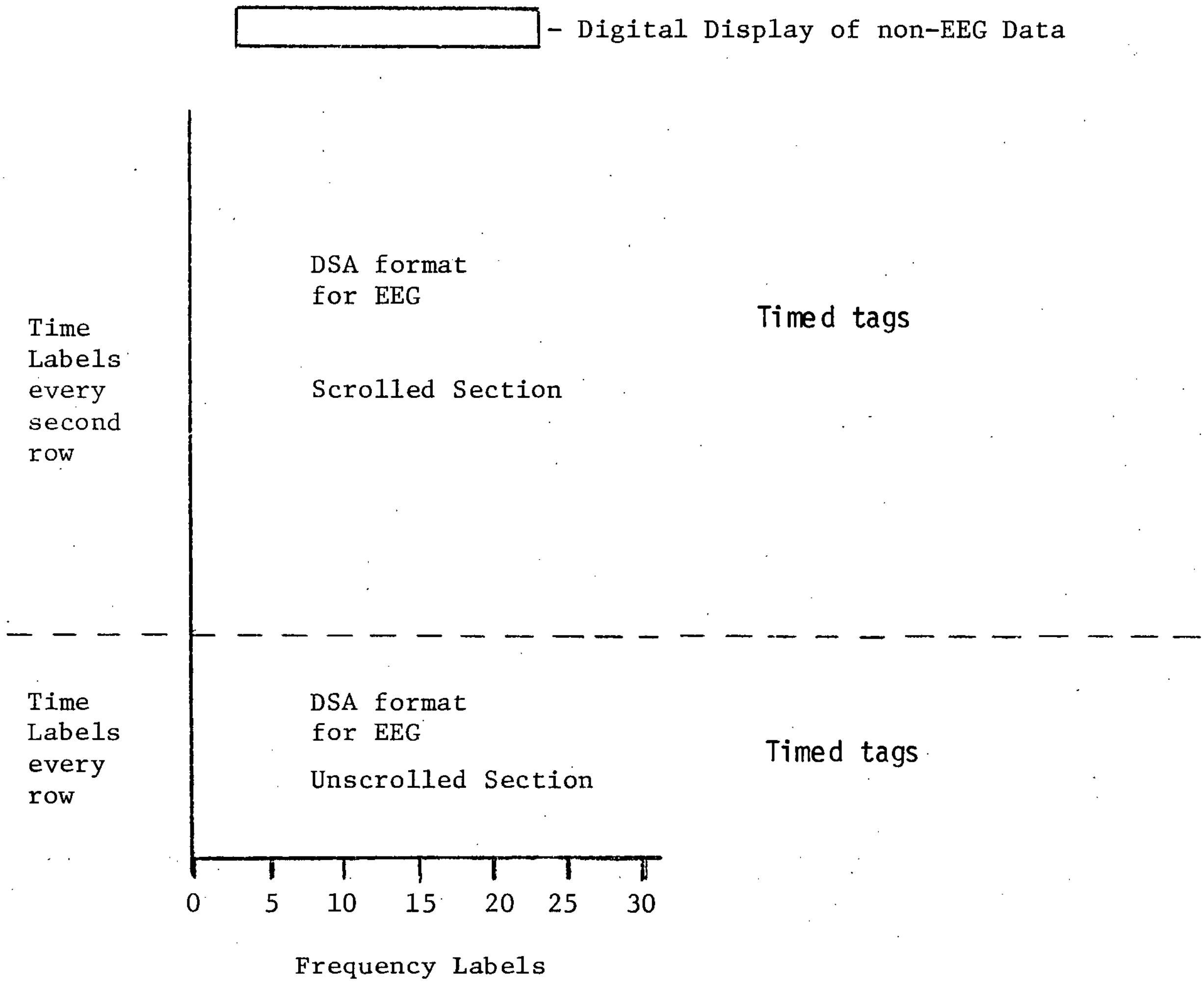


Figure 4: Display Type 1

TYPE 1 DISPLAY SUMMARY

DISPLAY UPDATE FREQUENCY....every 30 seconds

- not updated if the user has scrolled back in time

EEG DISPLAY FORMAT....DSA (power spectrum of EEG vs frequency and time)

DISPLAYED FREQUENCIES....0 to 31 Hz in 1 Hz increments

SCROLLED SECTION

- uses the scrolling feature to permit access to all EEG data
- automatically updated on the acquisition of new data
- takes up the top rows of the display
- time labels on every second row

UNSCROLLED SECTION

- accepts data transferred by the user from the scrolled section
- not automatically updated
- not affected by activity in the scrolled section
- takes up the bottom rows of the display
- time labels on every row

NON-EEG DISPLAY FORMAT....numeric typed at the top of the screen

- always associated with the top record of the scrolled section

TIMED TAGS....typed out beside the EEG display scrolls

DISPLAY TYPE 2 (display D5 to D6) (see Figure 5)

EEG...none

NON-EEG

The purpose of type 2 displays is to present the long term trends of up to four non-EEG parameters. Towards this end, the non-EEG data channels are plotted vs time on the terminal in a similar manner as is done by hand on standard anesthetic records.

The horizontal axis represents time. For scaling purposes, the times of acquisition of the most recent and the least recent records are typed at appropriate points below this axis.

The vertical axis represents magnitude of the data. Appropriate scaling factors and linetype definitions are printed above this axis.

It is expected that the resting state of MOMA will be a type 1 display. For this reason, it was not considered necessary to automatically update type 3 displays when new data is acquired. Consequently, these displays will only present data up to the time that they are entered.

TIMED TAGS...none

TYPE 2 DISPLAY SUMMARY

PLOTS NON EEG DATA vs TIME

- time is plotted on the horizontal axis
- data magnitude is plotted on the vertical axis
- up to 4 channels displayed simultaneously

TYPE TWO DISPLAYS ARE NOT UPDATED ON THE RECEIPT OF NEW DATA

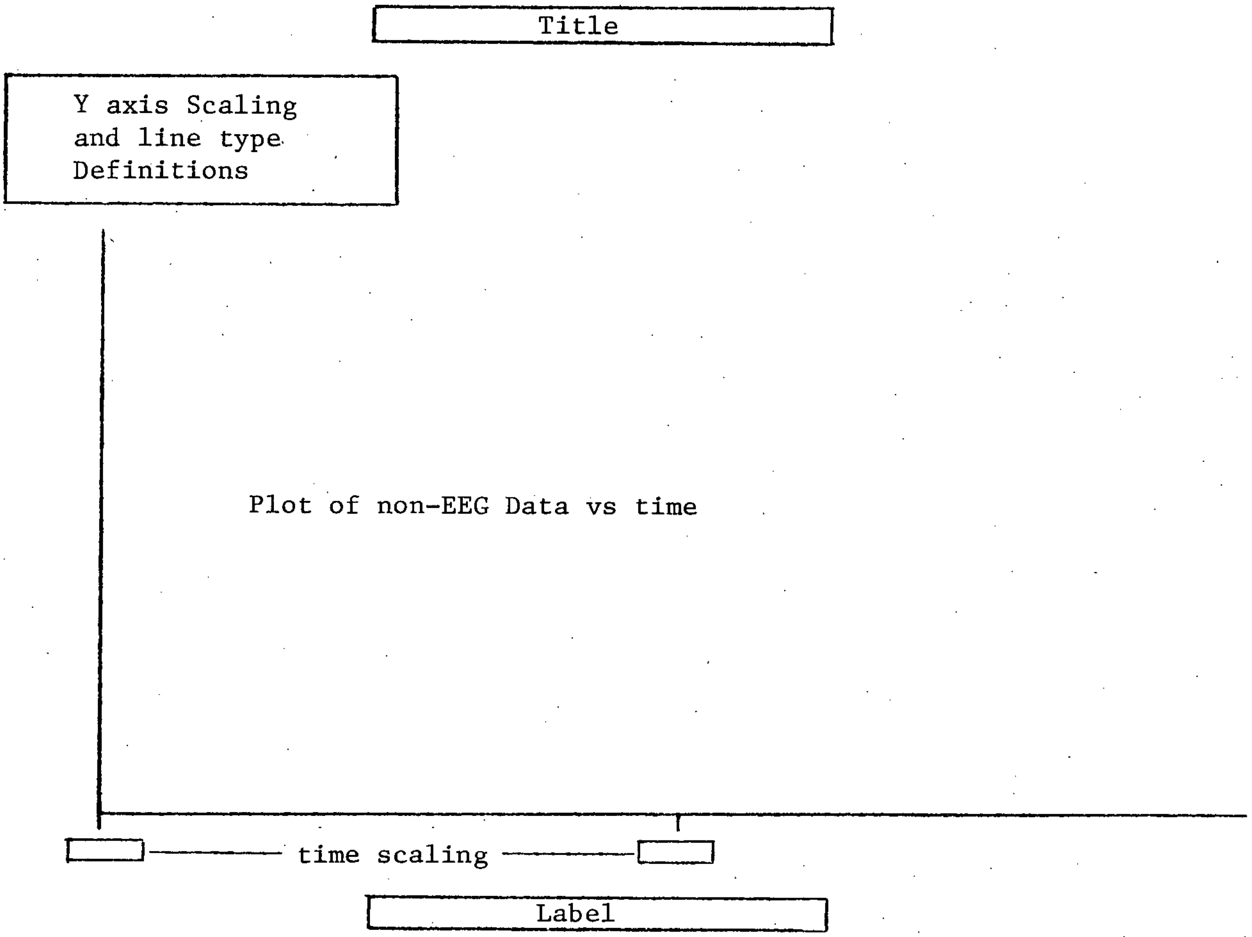


Figure 5: Display Type 2

A.3.2.2 Operation of MOMA

OPERATION OF MOMA

MOMA START-UP PROCEDURE

1) Insure that the "POWER" switch on the right side of the Tektronix 4025 terminal is ON. Note that brightness and contrast controls are located near this power switch (Figure 1).

2) LOCATE the "restart", "halt", and "aux on/off" switches on the front panel of the computer (see Figure 2).

3) Set the HALT switch to the UP (enable) position.

4) Set the AUX ON OFF switch to the UP (on) position.

5) REMOVE the "MOMA MASTER DISK" from its envelope and hold it with the label side up, label end towards you.

6) OPEN the LEFT-HAND disk DRIVE door by squeezing the black release mechanism in the center of the door and gently lifting.

7) Gently INSERT the master DISK as far as it will go. DO NOT FORCE

8) CLOSE THE DOOR by pulling it downwards until it clicks.

9) INSERT the "MOMA SLAVE DISK" in right-hand drive using the same procedure as was used for the master disk.

10) Lift and release the RESTART switch on the front panel of the computer.

11) WAIT. The computer will now send all further directions for start-up. A bell will be sounded when servicing is required.

MOMA TERMINATION PROCEDURE

1) ENTER the STOP INSTRUCTION by:

   a) holding down the "SHIFT" key and typing the "STOP" key.

   a) failing a), enter "{F<RETURN>".

2) WAIT until a period is typed onto the terminal's screen. Perform any procedures requested by the computer during this time.

3) OPEN the doors to BOTH DISK DRIVES by squeezing the black release mechanism in the center of the doors and gently lifting.

4) Gently REMOVE BOTH DISKS and insert them in their protective covers.

5) Set the "AUX ON OFF" to the DOWN (off) position. This switch is located on the computer control switch panel (see Figure 2).

6) DISCONNECT the data lines and the computer's power cord.

7) REMOVE the system from the operating theatre if necessary.

DEFAULT START-UP MODES

- Display number 1 (D1 or 4 EEG, short timed tags).
  - -empty unscrolled section.
  - -automatically updated scrolled section.
- Query generation on timed tag entry.

OPERATING CONSIDERATIONS

1) The lights on the "NUMERIC LOCK" key and the "COMMAND LOCKOUT" key (upper right corner of keyboard) should never be on. If they are, depress them once to turn them off.

2) Do not place objects on top of the terminal as this may impede proper cooling through the ventilation holes.

3) Do not depress the "BREAK" key (lower right corner of keyboard).

4) When handling diskettes:

   - do not bend
   - do not touch the exposed surface
   - keep dry
   - write on labels with soft tipped pens only
   - insert in protective envelope after use

Front of Terminal

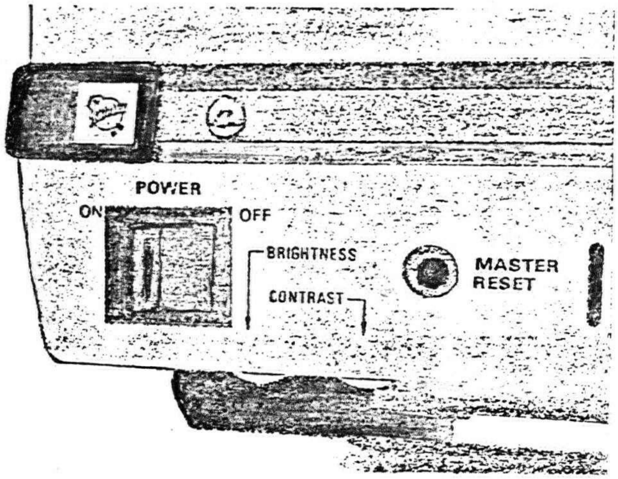


Figure 1: Terminal Control Switches

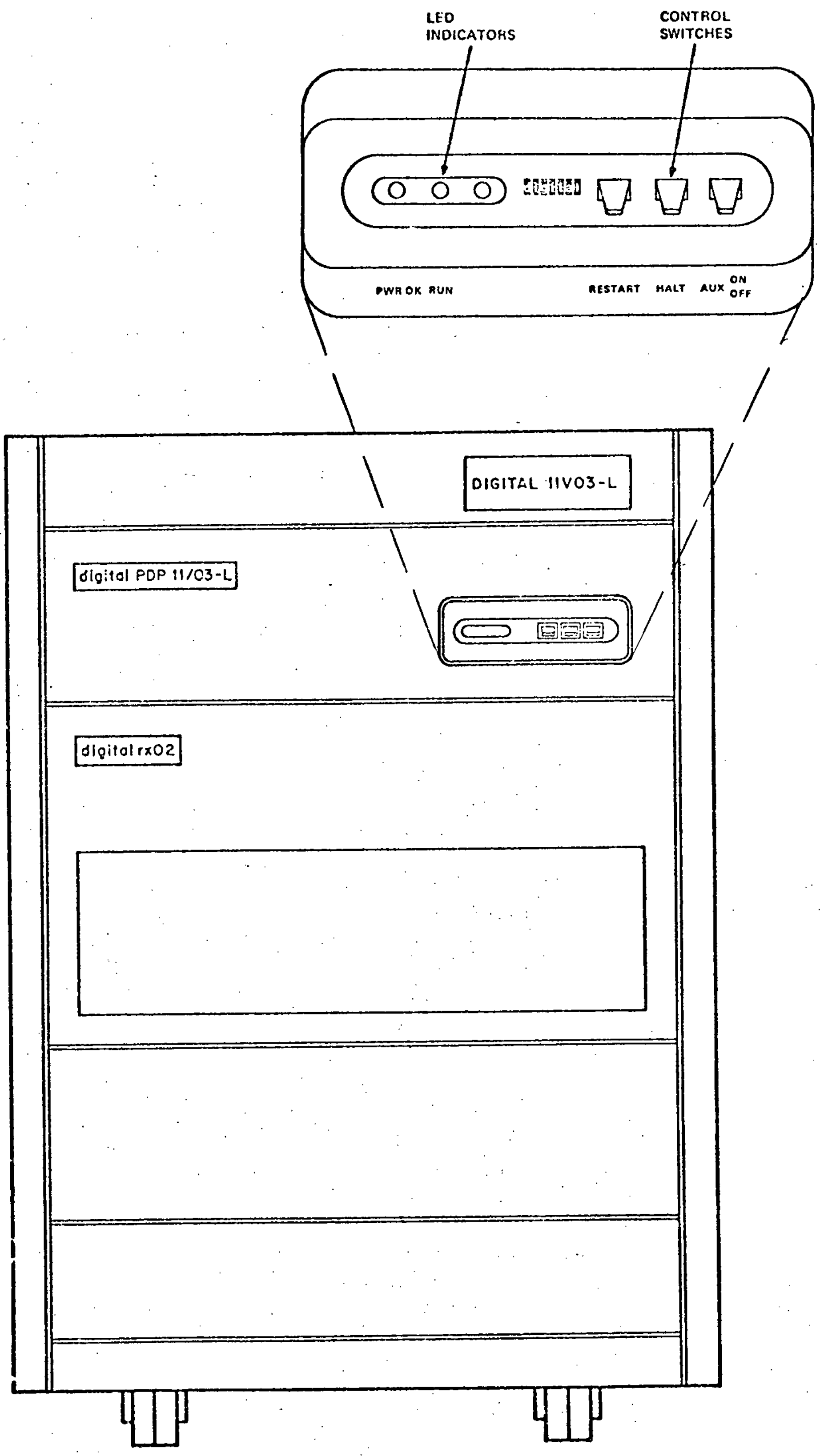
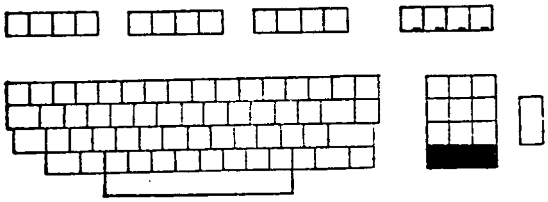


Figure 2: Computer Control Switches

MOMA KEYBOARD DETAILS

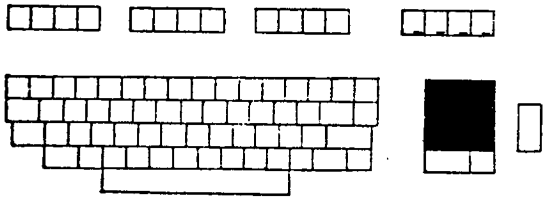
SCROLLING: (type 1 displays only)



DIRECTION CHANGE
("FOR" and "BACK")
({S. and {S0)

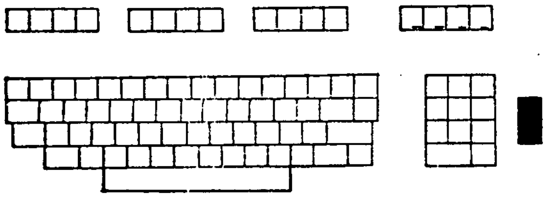
PURPOSE:
To set the direction of scrolling for type 1 displays to forward "FOR" (scroll towards future) or backward "BACK" (scroll towards past).

TIME JUMP
(keypad 1 to 9)
({S1 to {S9)



PURPOSE:
To move the scrolled section of type 1 displays 1 to 9 records forward or backward in time, depending on the last direction change.

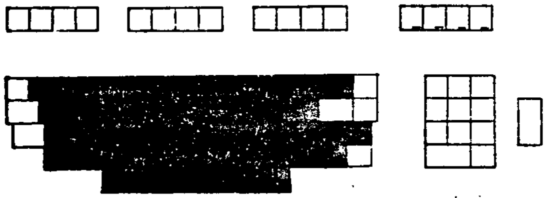
RETURN TO PRESENT
("PRESENT")
({SP)



PURPOSE:
To return the scrolled section of the type 1 displays to present time. It will read and display the most recently acquired data, restart automatic display updates, and set the scrolling direction to backward. Note that this key is the only key which will restart automatic display update.

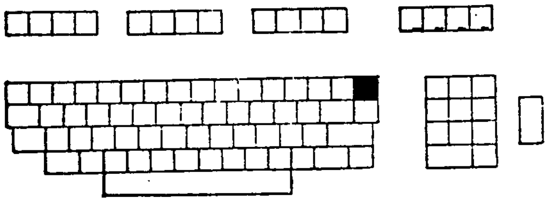
TIMED TAGS:

TAG ENTRY
(ALPHANUMERIC keys)



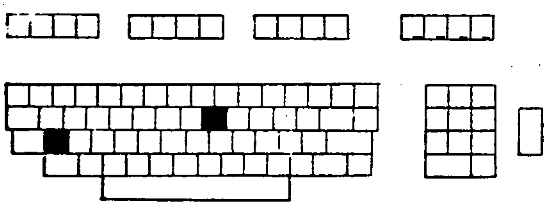
PURPOSE:
To enter relevant comments at any point during the operation. These comments are terminated with "RETURN".

CHARACTER DELETION
("RUBOUT")



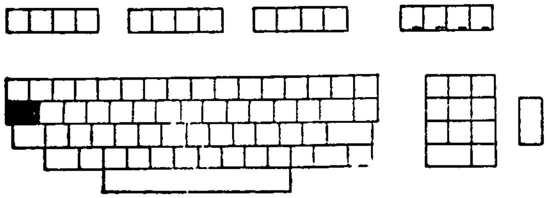
PURPOSE:
To rub out the character immediately preceeding the terminal's cursor.
If no character is there, this instruction is ignored.

LINE DELETION
(CNTL/U)



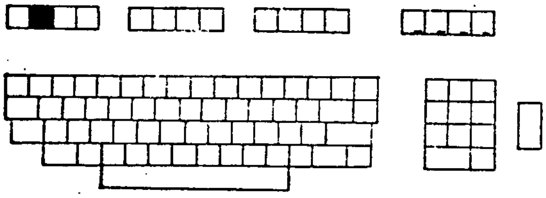
PURPOSE:
To delete a partially entered timed tag.
Note that the CTRL key must be held down while the "U" key is typed.

LINE REACTIVATION
("LAST LINE")



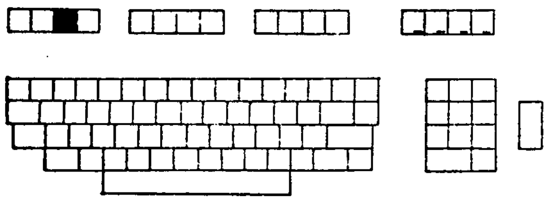
PURPOSE:
To reactivate the last character string entered.
If a partially completed timed tag is currently active no change will result.

QUERY ACTIVATION
("ON QUERY")
({Q)



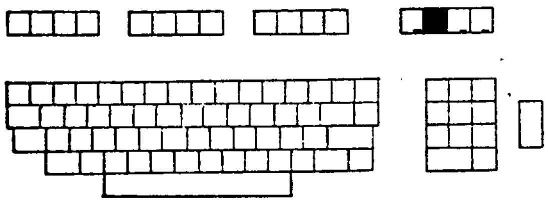
PURPOSE:
Activates the query on timed tag entry. i.e., the user will be asked to verify the validity of entered timed tags before they are stored.
This is the start-up mode.

QUERY DEACTIVATION
("OFF QUERY")
({N)



PURPOSE:
Deactivates the query on timed tag entry. i.e., timed tags will be stored immediately on entry.

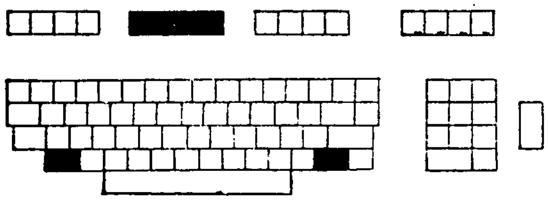
CAPITALIZATION
("TTY LOCK")



PURPOSE:
Capitalizes all alphabetic characters when lighted. Will not impede MOMA commands.

DISPLAY CHANGE:

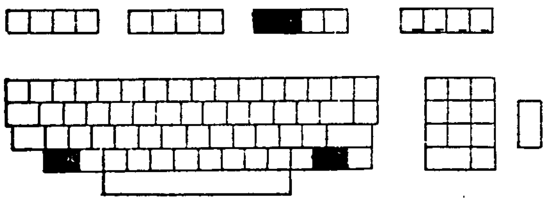
TYPE 1 DISPLAYS
(D1 to D4)
({1 to {4)



PURPOSE:
To change the display format to an EEG display which uses the two part DSA format. The start-up display is D1. Note that the "SHIFT" key be held down while the "D*" key is pressed.

D1 = four 10 line EEGs,
short timed tags
D2 = two 10 line EEGs,
long timed tags
D3 = two 20 line EEGs,
short timed tags
D4 = one 20 line EEG,
long timed tags

TYPE 2 DISPLAYS
(D5 and D6)
({5 to {6)



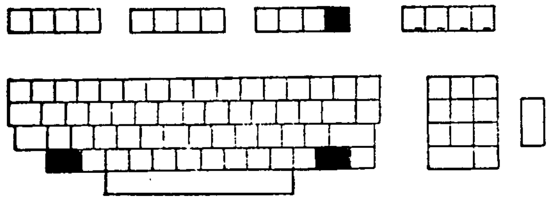
PURPOSE:
To change the display format to a display which plots non-EEG data vs time. Note that these displays are not automatically updated and cannot be scrolled. Note also that

the "SHIFT" key must be held down while the "D*" key is pressed.

D5 = the last 79 samples
D6 = from start of operation to present

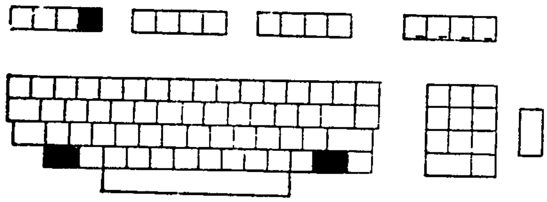
SPECIAL FUNCTIONS:

RECORD TRANSFER (type 1 only)
("SAVE")
({T)



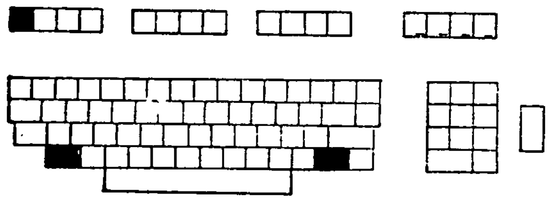
PURPOSE:
To take the record at the top of the scrolled section and inserts it to the top of the unscrolled section of type 1 displays. Note that the "SHIFT" key must be held down while the "SAVE" key is pressed.

TERMINAL REINITIALIZATION
("INIT")
({I)



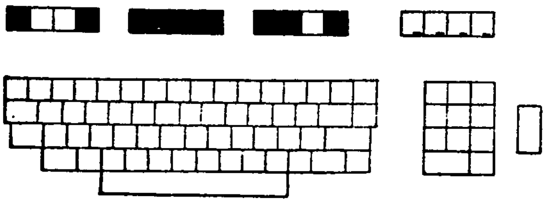
PURPOSE:
To reinitialize the modes and keyboard settings of the graphics terminal and to initialize the character font for DSA displays. Note that the "SHIFT" key must be held down while the "INIT" key is pressed.

TERMINATION
("STOP")
({F)



PURPOSE:
To stop the MOMA programs. Note that the "SHIFT" key must be held down while the "STOP" key is pressed.

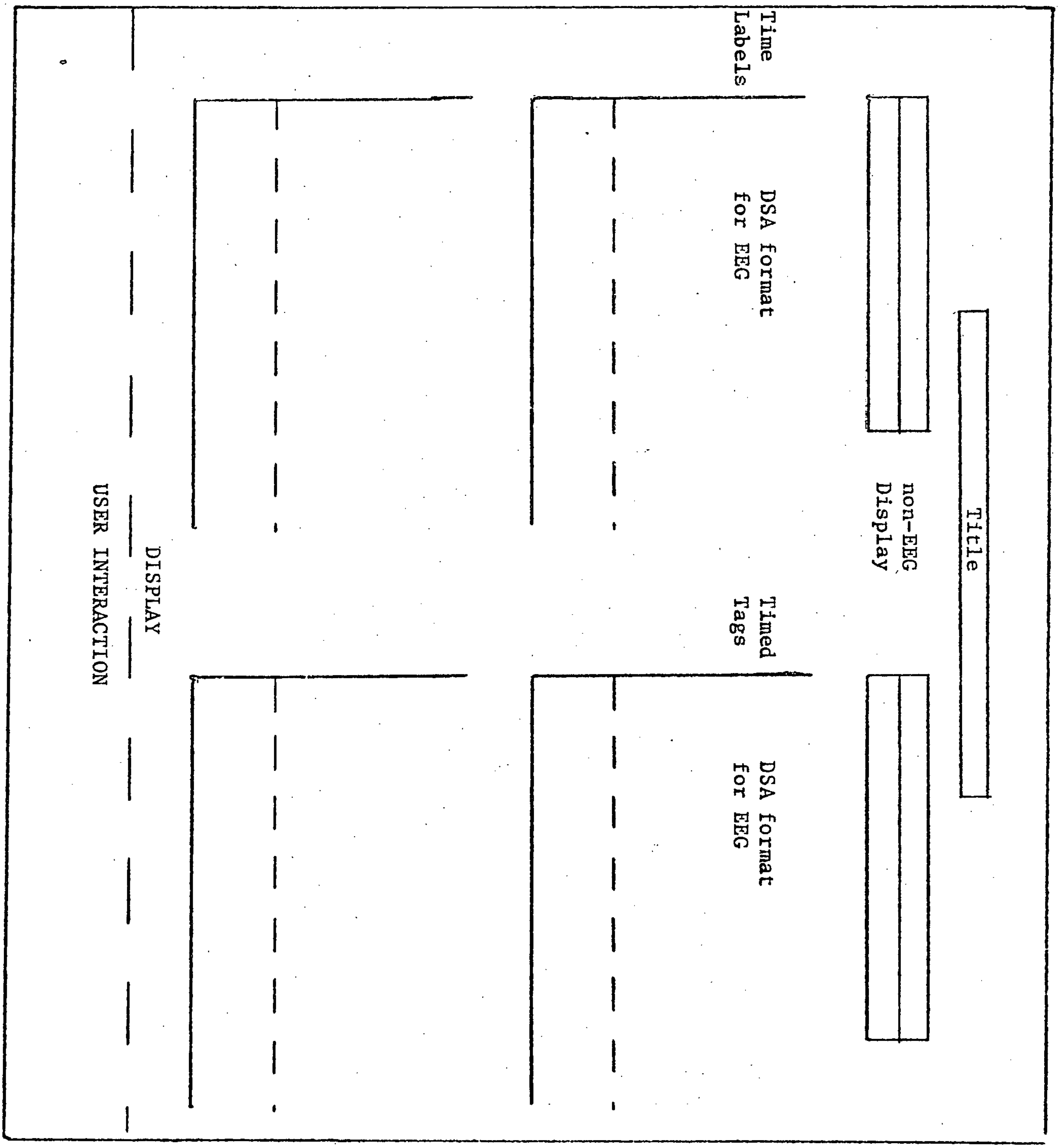
SHIFT KEY WARNING
(unshifted version of)
(D1-D6,INIT,SAVE,STOP)
({E)



PURPOSE:
To remind the user that the "SHIFT" key must be used to send the instruction from the last key typed.

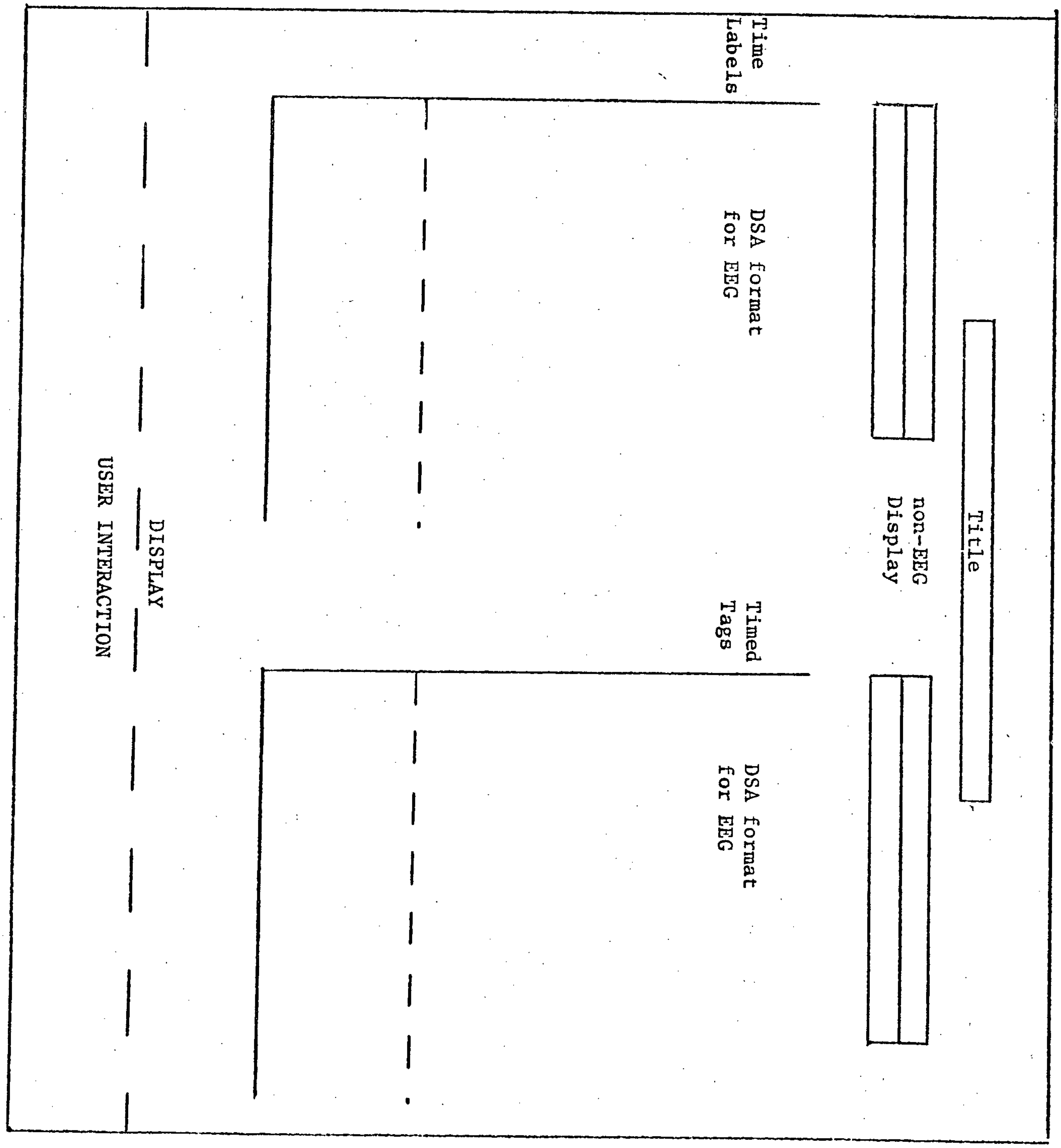
DISPLAY NUMBER 1 (D1)

-Type 1
-10 lines of DSA for four EEG channels
 -scrolled section = top seven rows
 -unscrolled section = bottom three rows
-Time and frequency labelling
-Timed tags truncated to 8 characters
-Digital Display of up to four non-EEG data channels



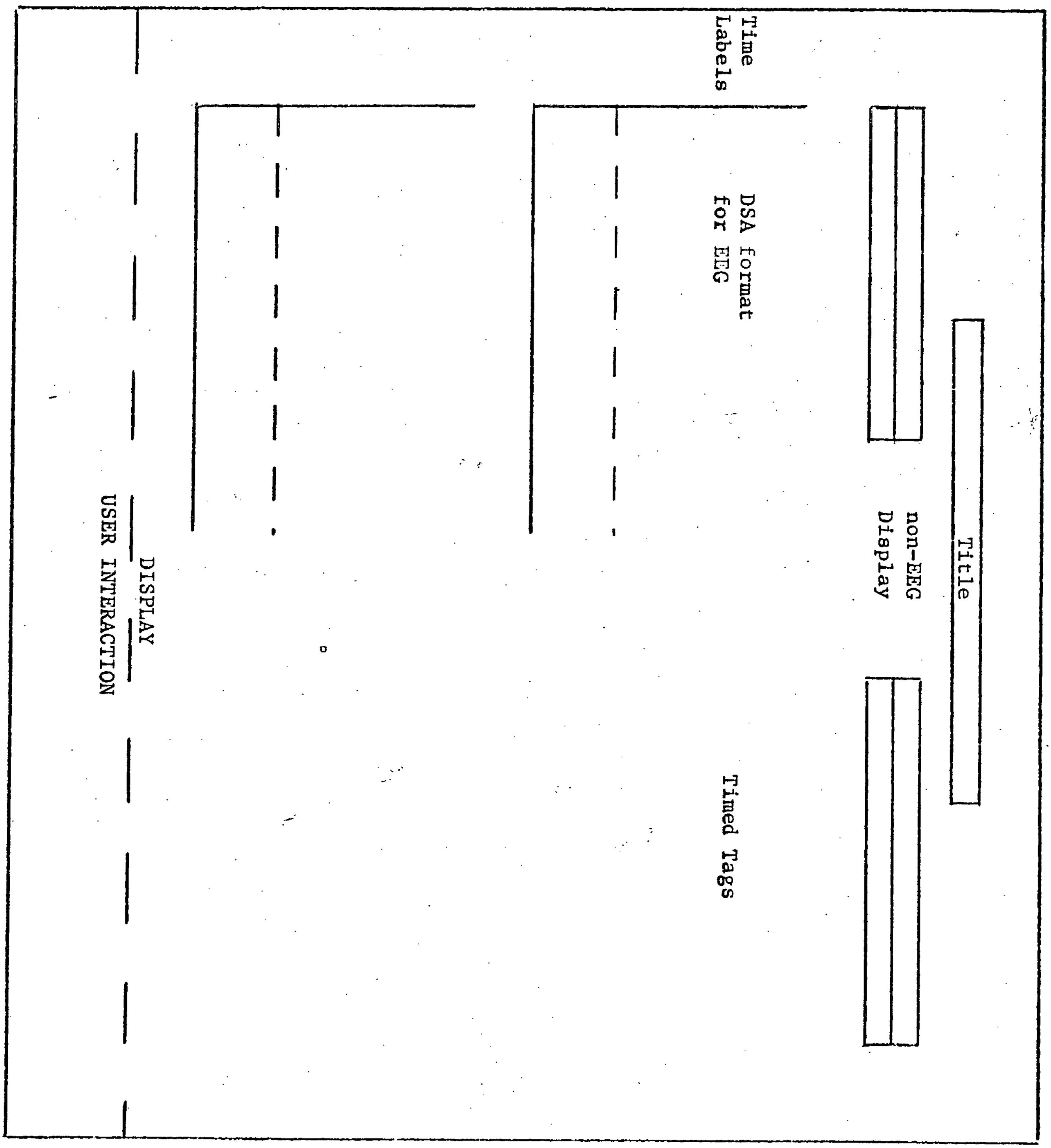
DISPLAY NUMBER 2 (D2)

-Type 1
-20 lines of DSA for two EEG channels
 -scrolled section = top 15 lines
 -unscrolled section = bottom 5 lines
-Timed tags truncated to 9 Characters
-Digital Display of up to four non-EEG Data channels



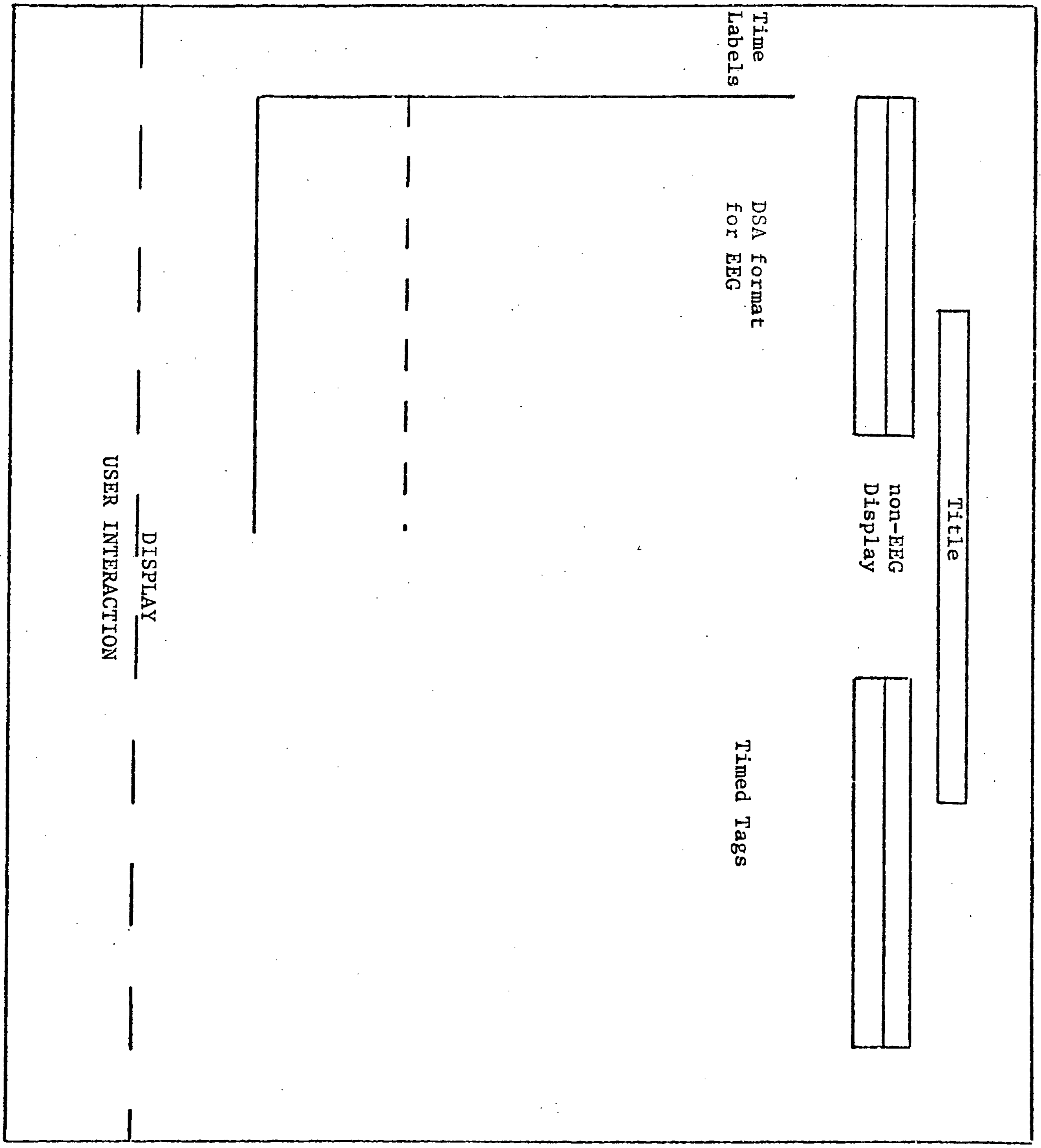
DISPLAY NUMBER 3 (D3)

-Type 1
-10 lines of DSA for two EEG channels
 -scrolled section = top 7 lines
 -unscrolled section = bottom 3 lines
-Timed tags truncated to 32 Characters
-Digital Display of up to four non-EEG Data channels



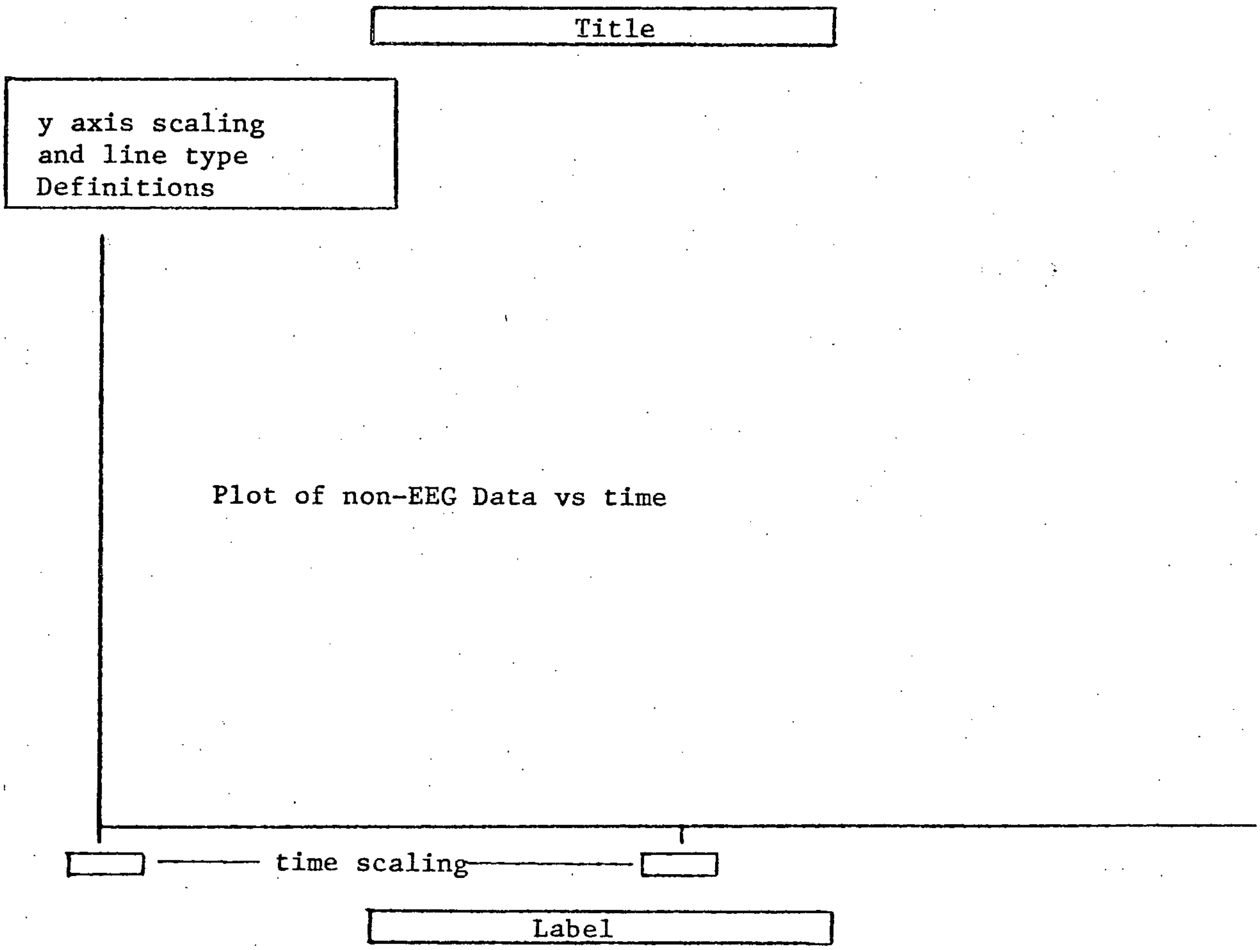
DISPLAY NUMBER 4 (D4)

-Type 1
-20 lines of DSA for one EEG channel
 -scrolled section = top 15 lines
 -unscrolled section = bottom 5 lines
-Timed Tags truncated to 32 Characters
-Digital Display of up to four non-EEG Data channels



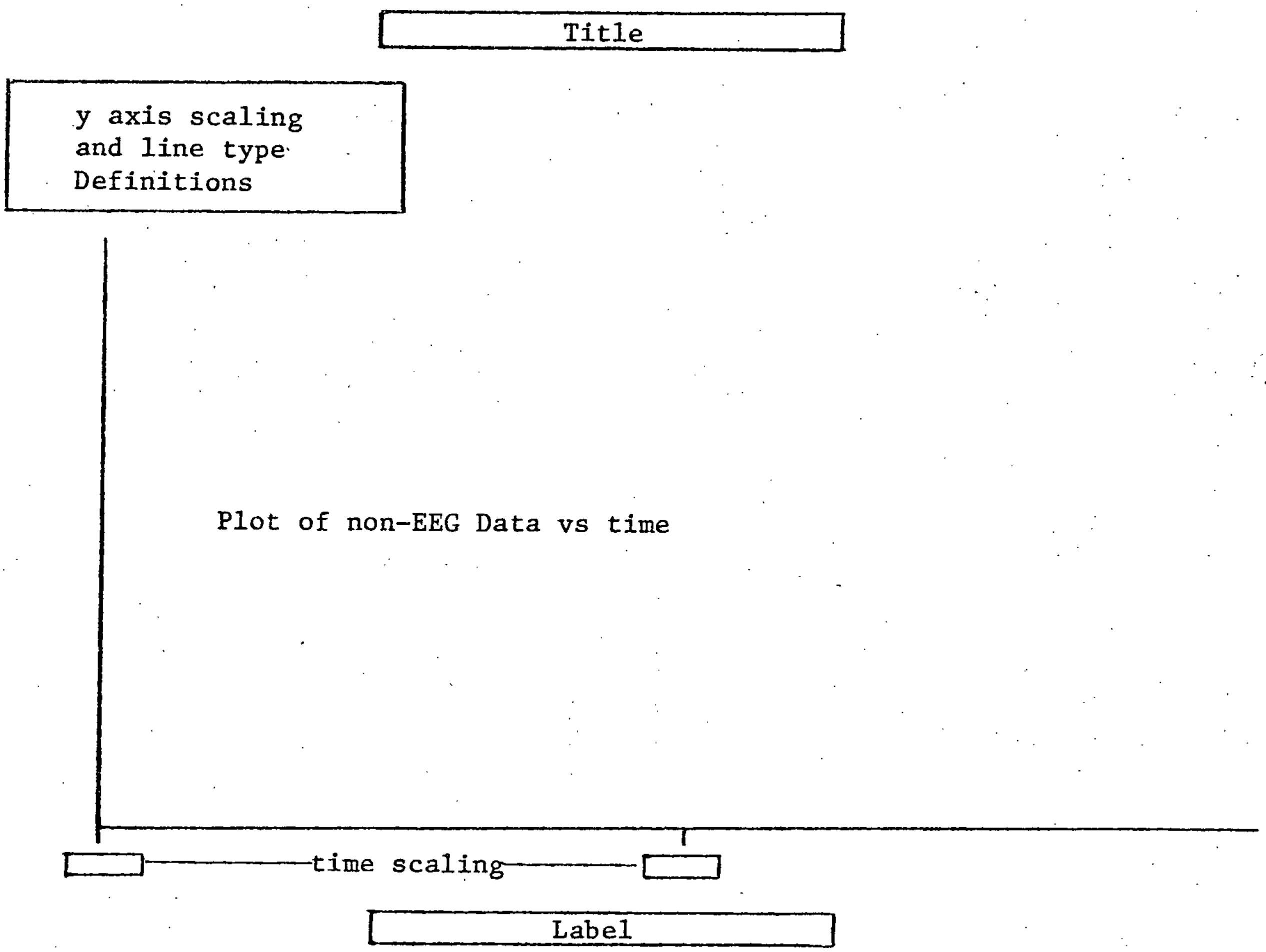
DISPLAY NUMBER 5

-Type 2
-Plot vs time of up to four non-EEG Data Channels
-Presentation of the last 79 samples



DISPLAY NUMBER 6

-Type 2
-Plot vs time of up to four non-EEG Data channels
-Presentation of data from the start of the operation to present



CHECKLIST IN CASE OF MOMA START FAILURE

CHECK FOR THE SYMPTOMS LISTED IN TABLE 1

???? IS THERE ANY APPARENT SYSTEM ACTIVITY OR NOISE ????

no: 1) insure that the power cord from the computer has been properly plugged into the hospital's power supply.

TRY TO START MOMA AGAIN, USING THE PROCEDURES DESCRIBED IN THIS DOCUMENT AND MAKING SURE THAT NO STEPS ARE MISSED

???? IS THE TERMINAL'S POWER SWITCH LIGHTED ????

no: 1) press it once to make sure it is fully on.

2) check that the power cord is connected from the back of the terminal to the computer.

???? IS THE TERMINAL ACTIVE ????

no: 1) check the power switch and power cord connections on terminal.

2) make sure the communications cord is connected to the computer.

3) rotate the contrast control dial near the terminal's power switch in a clockwise direction (roll towards terminal's front) (see Figure 1).

4) rotate the brightness control dial near the terminal's power switch in a counterclockwise direction (roll towards terminal's back) (see Figure 1).

5) - perform the standard baud rate setting procedure (see page 18).

- perform the standard command character setting procedure (see page 18).

IF THE TERMINAL IS NOT REACTING PROPERLY AND THE ABOVE PROCEDURES FAIL TO SOLVE THE PROBLEM, PUSH THE "MASTER RESET" BUTTON NEAR THE TERMINAL'S POWER SWITCH AND PERFORM THE FIFTH PROCEDURE OF QUESTION NUMBER 3.

CHECKLIST IN CASE OF INTRAOPERATIVE MALFUNCTION

CHECK FOR THE SYMPTOMS LISTED IN TABLE 2

TYPE "<RETURN>"

???? DOES THE QUALITY OF THE DSA DISPLAY APPEAR TO BE DETERIORATING ????

yes: 1) enter the "INIT" instruction by;

a) holding down the "SHIFT" key and typing the "INIT" key

b) failing a), type "{I<RETURN>".

???? DOES THE TERMINAL ALONE APPEAR TO BE THE PROBLEM ????

yes: 1) enter the INIT instruction as above

ATTEMPT A SYSTEM RESTART BY LIFTING THE "RESTART" SWITCH ON THE COMPUTER CONTROL SWITCH PANEL ON THE FRONT OF THE TERMINAL. NOTE THAT THIS IS THE SAME AS SYSTEM START-UP PROCEDURE STEP 10. YOU WILL NOW USE THE START FAILURE CHECKLIST IF NECESSARY.

STANDARD MODE-SETTING PROCEDURES

BAUD RATE SETTING:

1) If the light on the "STATUS/COMMAND LOCKOUT" key is on, type it once to turn it off. (upper right corner of keyboard)

2) Hold down the "SHIFT" key and type the "STATUS/COMMAND LOCKOUT" key. The system will type the following message on the terminal's screen:

   X X NNN

   X = some character

   N = some number

3) Type the second X followed with "BAU 9600<RETURN>". e.g., when message is U ` 335 enter "`BAU 9600<RETURN>".

COMMAND CHARACTER SETTING:

1) Perform steps 1 and 2 from above

2) - If the second X is "`", do nothing.
   - If the second X is NOT "`", type the second X followed the "COM `<RETURN>".
     e.g., when message is U ! 335 enter "!COM `<RETURN>´"

NOTE: "`" = BACKQUOTE or SHIFT/BACKSLASH
(third row up, third key in from right side of alphanumeric keys)

TABLE 1: CONDITIONS WHICH PREVENT START-UP

CONDITION	SYMPTOM	SOLUTION
HALT SWITCH DOWN	- after lifting "RESTART" the following message is immediately written on terminal's screen. 173000 @	- raise halt switch
NO MASTER DISK INSERTED IN RIGHT-HAND DRIVE	- after lifting "RESTART" and waiting about 7 seconds the following message is written on the terminal's screen. 173712 @	- insert master disk in left-hand drive
UNBOOTABLE DISK INSERTED IN RIGHT-HAND DRIVE (e.g., slave disk)	- after lifting "RESTART" and waiting about 4 seconds the following message is written on the terminal's screen. ?BOOT-F-NO boot on volume	- replace disk in left-hand drive with MOMA MASTER DISK
BOOTABLE BUT INCORRECT DISK INSERTED IN RIGHT-HAND DRIVE	- system types a period "." after a few seconds. - messages unrelated to MOMA are typed onto the terminal's screen.	- replace disk in left-hand drive with MOMA MASTER DISK
NO SLAVE DISK OR UNCLOSED DISK DOOR	- SEVEN CLANKING SOUNDS followed with the writing of the following message on the terminal's screen. ?MON-F-Dir io err 004360	- insert MOMA SLAVE DISK in right-hand drive - close door to disk drive
ILLEGAL SLAVE DISK	- After completion of most of the start-up procedure the following message is written onto the terminal's screen. 137270 @	- replace disk in right-hand drive with MOMA SLAVE DISK

Continued on next page...

CONDITION	SYMPTOM	SOLUTION
COMMAND LOCKOUT KEY LIGHT ON OR SYSTEM COMMAND CHARACTER IS ILLEGAL	- character strings starting with the character "`" are imbedded in the messages written on the terminal's screen - no bells are heard - the screen is not erased between messages	- if the light on the COMMAND LOCKOUT KEY is on type it once to turn it off - failing the above, perform the standard command character setting procedure (see page 18).
INCORRECT TERMINAL BAUDRATE	- after lifting "RESTART" a string of unexpected characters is typed onto the terminal's screen - the computer makes a clanking sound whenever a key is typed	- perform the standard baud rate setting procedure (see page 18).

TABLE 2: CONDITIONS WHICH PREVENT PROPER OPERATION

CONDITION	SYMPTOM	SOLUTION
NUMERIC LOCK KEY LIGHT ON	- when the keys programmed to produce scrolling are typed numbers are produced.	- if the NUMERIC LOCK key light is on, type it once to turn it off.
COMMAND LOCKOUT KEY LIGHT ON OR SYSTEM COMMAND CHARACTER IS ILLEGAL	- many unexpected characters typed into the interaction scroll. - or display scrolled off the screen by many unexpected characters in the display scroll.	- if the COMMAND LOCKOUT key light is on, type it once to turn it off - failing the above, perform the standard command character setting routine (see page 18).
UNANSWERED SYSTEM REQUESTS	- after being unanswered for over 30 seconds, there will be no data acquisition or display update.	- answer the request
UNDEFINED CHARACTER FONT	- unexpected characters typed into DSA plots.	- reinitialize the terminal using the INIT instruction
UNPROGRAMMED SPECIAL FUNCTION KEYS	- special function keys not performing their intended function	- reinitialize the terminal using the INIT instruction

APPENDIX 4 ASSESSMENT QUESTIONNAIRES

A4.1 System Assessment Questionnaire

TYPE DEMMOM.FRM

MOMA DEMONSTRATION QUESTIONNAIRE 1/4

1) **BIOGRAPHICAL DATA:** DATE ____________________

SYSTEM DEMONSTRATOR ____

ANESTHETIST ____

2) **CHOICE OF PARAMETERS**

For all questions;

a) Disregard difficulties in obtaining signals appropriate for acquisition by a computer.

b) Indicate when an answer applies only to specific procedure(s).

2.1) List, in order of importance, six parameters which you feel are essential for monitoring patients during

Open Heart Surgery	and	Carotid Endarterectomies.
______________________		______________________
______________________		______________________
______________________		______________________
______________________		______________________
______________________		______________________
______________________		______________________

2.2) List any parameters with which you feel that trend information is not adequately presented using existing instrumentation.

____

____

____

____

2.3) List any parameters which you feel would be more useful either during of after an operation if they were recorded more frequently

____

____

____

____

3) MOMA ASSESSMENT 2/4

3.1) Give your initial impression of the potential usefulness of the following features of MOMA:

	0 = do not know	poor				excellent
The multiple display capability	0	1	2	3	4	5
The scrolling feature of the EEG displays	0	1	2	3	4	5
The two section EEG display	0	1	2	3	4	5
The time plots	0	1	2	3	4	5
The timed tagging feature	0	1	2	3	4	5
Simplicity of operation	0	1	2	3	4	5
Keyboard configuration	0	1	2	3	4	5
Documentation	0	1	2	3	4	5
Overall assessment	0	1	2	3	4	5

3.2) Can you suggest any changes which would improve any of the above ratings.

3.3) Will the combination of the scrolling feature and the two section display be sufficient for performing the comparisons necessary for using the EEG as a monitoring parameter in the operating room. ---------

If not, what is lacking.

3.4) Specify any ambiguities or characteristics unexplained in the documentation provided with this system.

3.5) Specify any specific commands or features not present on the existing monitor which would improve its general usefulness.

3.6) Do you feel that the use of MOMA will take a disproportionate amount of your time away from more essential tasks. ----------

4) GENERAL COMMENTS

A4.2: Software Assessment Questionnaire

TYPE SOFTQU.FRM

1/2

MOMA SOFTWARE GENERAL ASSESSMENT FORM

1) BIOGRAPHICAL DATA DATE ___________________

ASSESSOR ____

1.1) Are you familiar with the following things:

	no	yes
FORTRAN IV	______	______
RT-11	______	______
The objective of MOMA	______	______

1.2) Have you read the following documents:

MOMA SUMMARY	______	______
MOMA USERS GUIDE	______	______

2) GIVE YOUR ASSESSMENT OF THE OVERALL QUALITY OF THE FOLLOWING MODULES:
(this assessment includes overall program flexibility, usefulness)
(and completeness of the documentation, and appropriateness)
(of the programs for their intended tasks)

0 = do not know

		poor				excellent
DOCHLP.TXT	0	1	2	3	4	5
COMRTN.TXT	0	1	2	3	4	5
modules described in COMRTN.TXT	0	1	2	3	4	5
SRT*.TXT	0	1	2	3	4	5
modules described in SRT*.TXT	0	1	2	3	4	5
FOR*.TXT	0	1	2	3	4	5
modules described in FOR*.TXT	0	1	2	3	4	5
BAK*.TXT	0	1	2	3	4	5
modules described in BAK*.TXT	0	1	2	3	4	5
GPH*.TXT	0	1	2	3	4	5
modules described in GPH*.TXT	0	1	2	3	4	5
FLOW.TXT	0	1	2	3	4	5
modules described in FLOW.TXT	0	1	2	3	4	5
VISXXX.DOC	0	1	2	3	4	5
modules described in VISXXX.DOC	0	1	2	3	4	5

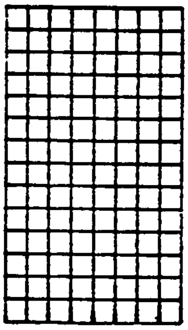
3) Do you feel that the software described above will be useful for the development of an effective computer-based patient monitoring system. ________

4) GENERAL COMMENTS

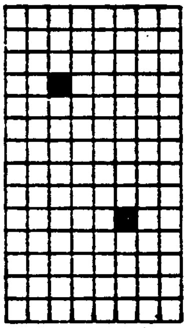
Appendix 5

Dot matrix patterns for character font 16

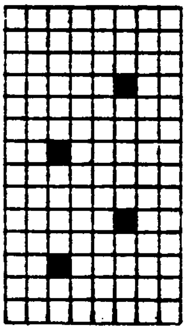
33
!



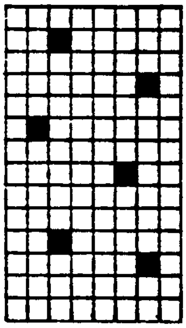
34
"



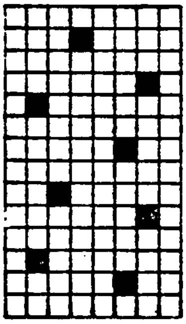
35
#



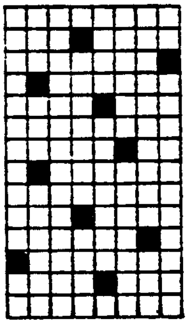
36
$



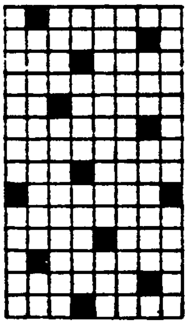
37
%



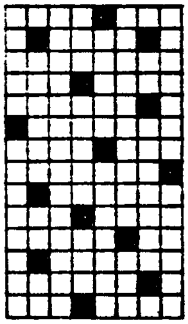
38
&



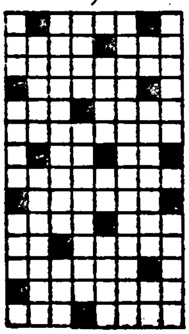
39
'



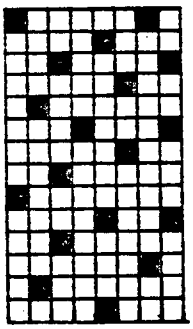
40
(



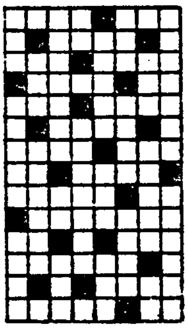
41
)



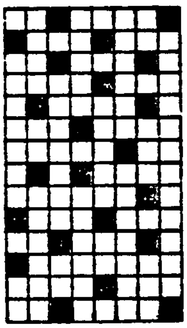
42
*



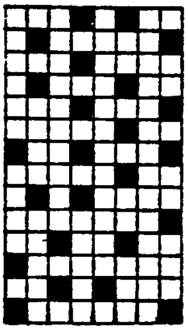
43
+



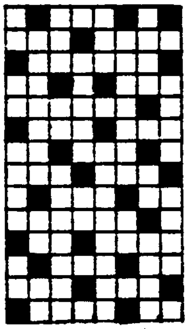
44
,



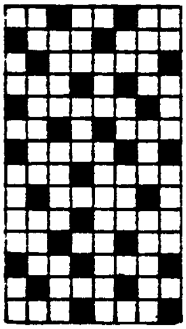
45
–



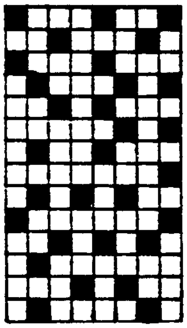
46
.



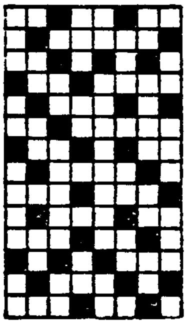
47
1



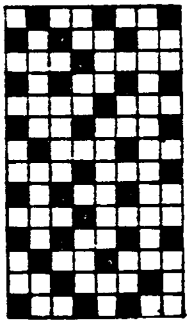
48
0



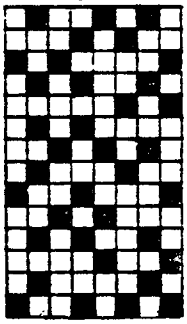
49
1



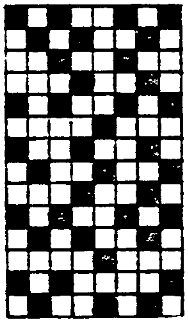
50
2



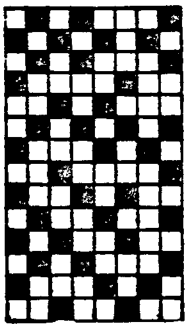
51
3



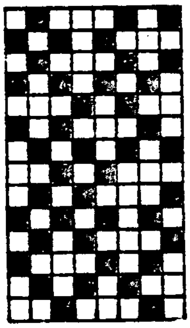
52
4



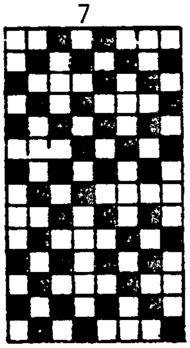
53
5



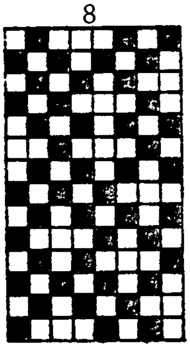
54
6



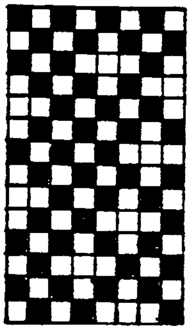
55
7



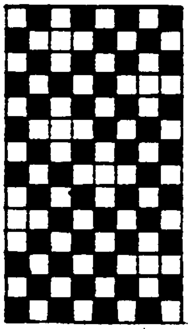
56
8



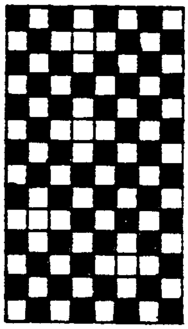
57
9



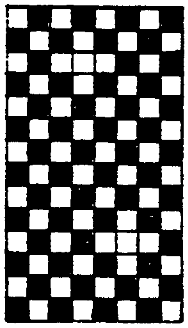
58
:



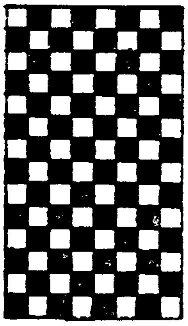
59
;



60
<

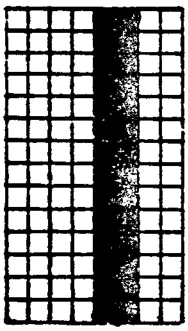


61
=

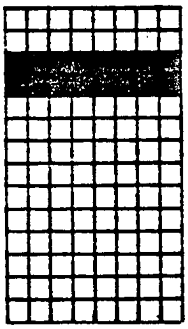


$$\begin{matrix} 62 & \leftrightarrow & 89 \\ > & \leftrightarrow & \mathrm{Y} \end{matrix} = \text{the ones complement of } \begin{matrix} 60 & \leftrightarrow & 33 \\ < & \leftrightarrow & ! \end{matrix}$$

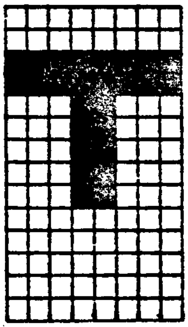
97
a

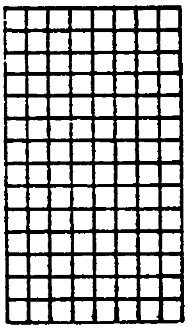


98
b



99
c





REFERENCES

[1] E.I. Eger II, "Monitoring the Depth of Anesthesia", Monitoring in Anesthesia, L.J. Saidman, N.T. Smith (editors), John Wiley & Sons, New York, 1978, pp. 1-15.

[2] J.B. Cooper, R.S. Newbower et al., "Preventable Anesthesia Mishaps: A Study of Human Factors", Anesthesiology, vol. 49, 1978, pp. 399-406.

[3] A. Boba, "Essays on Future Trends in Anesthesia", Anesthesiology and Resuscitation, Springer-Verlag Berlin, Heidelberg, New York, 1972, pp. 1-21.

[4] S.C. Cullen, "Reliance on Cybernetics", Anesthesiology, vol. 51, No. 1, July 1979, pp. 97-98.

[5] S.A. Glantz, "Computers in Clinical Medicine: A Critique", Computer IEEE, May 1978, pp. 68-77.

[6] B. Smalhout, "The Importance of Monitoring in Anesthesia", Acta Anesthesiologica Belgica, vol. 29, No. 1, 1978, pp. 45-69.

[7] J.J. Osborn et al., "Managing the Data from Respiratory Measurements", Medical Instrumentation, vol. 13, No. 6, November-December 1979, pp. 330-336.

[8] R.A. de Asla, C.W. Bryan-Brown, "The Clinically Valuable Variable", Medical Instrumentation, vol. 13, No. 6, November-December, 1979, pp. 325-326.

[9] N.P. Cass, D.G. Lampard, W.A. Brown, and J.R. Coles, "Computer Controlled Muscle Relaxation: A Comparison of 4 Muscle Relaxants in the Sheep", Anesthesia and Intensive Care, vol. 4, No. 1, February 1976, pp. 16-22.

[10] D.G. Lampard, J.R. Coles, W.A. Brown, "Electronic Digital Computer Control of Ventilation and Anesthesia", Anesthesia and Intensive Care, vol. 1, No. 5, August, 1973, pp. 382-392.

[11] W.W. Mapleson, "From Clover to Computer", Anaesthesia, vol. 34, Blackwell Scientific Publications, 1979, pp. 163-172.

[12] D. Daub, R. Vom Hövel, G. Kalff, R. Repges, "A Process Control Computer for Controlling Anaesthesia", Recent Progress in Anaesthesiology and Resuscitation, A. Arias et al., American Elsevior Publishing Co. Inc., New York, 1974, pp. 662-665.

[13] H. Lutz, "Anaesthesia Record System with Mark Page Reader", Recent Progress in Anaesthesiology and Resuscitation, A. Arias et al., American Elsevior Publishing Co. Inc. New York, 1974, pp. 647-651.

[14] D.D.C. Howat, "Why keep Anaesthetic Records", Recent Progress in Anaesthesiology and Resuscitation, A. Arias et al., American Elsevior Publishing Co. Inc., New York, 1974, pp. 640-642.

[15] M. Hilberman, "Monitoring in the Operating Room: Current Techniques and Future Requirements", Medical Instrumentation, vol. 11, No. 5, September-October, 1977, pp. 283-287.

[16] R.M. Zollinger, J.F. Kreul, J.L. Schneider, "Man-Made versus Computer-Generated Anesthesia Records", Journal of Surgical Research, vol. 22, 1977, pp. 419-424.

[17] M.J. Shaffer, P.R. Kaiser et al., "Manual Record keeping and Statistical Records for the Operating Room", Medical Instrumentation, vol. 12, No. 3, May-June, 1978, pp. 192-197.

[18] E.B. Cox and W.E. Stanley, "Schema-Driven Time-Oriented Record System on a Minicomputer", Computers and Biomedical Research, vol. 12, 1979, pp. 503-516.

[19] B.M. Tompkins, "A Computer Assisted Preanaesthesia Interview: Value of a Computer-Generated Summary of Patients Historical Information in the Preanaesthesia Visit", Anesthesia and Analgesia, vol. 59, No. 1, January 1980, pp. 3-10.

[20] A.W. Paulsen et al., "A Computerized Operating Room Monitoring System Employing X-Y Plotter Hard Copy Output at the Anaesthesia Machine", Advances in Anesthesiology and Monitoring, R.S. Newbower, AAMI 15th Annual Meeting, San Francisco, April 13-17, 1980, p. 161.

[21] F.A. Gibbs, E.L. Gibbs, W.G. Lennox, "Effect on the EEG of Certain Drugs which Influence Nervous Activity", Arch. Intern. Mod., vol. 60, July, 1937, pp. 154-166.

[22] L.J. Rogers, "EEG Frequency Band Correlates of Depth of Surgical Anesthesia", Internal Paper, Vancouver General Hospital, 1976.

[23] J.A. McEwen, "Estimation of the Level of Anesthesia During Surgery by Automatic EEG Recognition", Ph.D. Thesis, UBC, 1975.

[24] D.G. McDowall, "Monitoring the Brain", Anesthesiology, vol. 45, No. 2, August, 1976, pp. 117-134.

[25] T.A. Salerno et al., "Cerebral Surveillance during Cardiac Surgery", The Canadian Journal of Surgery, vol. 22, No. 4, July 1979, pp. 325-326.

[26] T.M. Sundt et al., "Monitoring Techniques for Carotid Endarterectomy", Clinical Neurosurgery, vol. 22, The Williams Wilkins Co., Baltimore, 1974, pp. 199-212.

[27] H.G. Beebe et al., "Comparison of Carotid Artery Stump Pressure and EEG Monitoring in Carotid Endarterectomy", The American Surgeon, vol. 99, No. 10, October 1978, pp. 655-660.

[28] G. Rolly, "2 Cases of Burns Caused by Misuse of Coagulation Unit and Monitoring", Acta Anaesthesiologica Belgica, vol. 29, No. 3, 1978, pp. 313-317.

[29] J. Spierdijk et al., "Burns Caused by Monitoring in Anesthesia", Acta Anaesthesiologica Belgica, vol. 29, No. 3, 1978, pp. 305-313.

[30] Canadian Standards Association, CSA Standard C 22.2, No. 125, 1979.

[31] K. Uyttendaele, "Monitoring Instrumentation, Isolated Inputs, Electrosurgery Filtering, Burns Protection: What does it mean?" Acta Anaesthesiologica Belgica, vol. 29, No. 3, 1978, pp. 317-330.

[32] J.G. Webster, "Medical Instrumentation Application and Design", Houghton Mifflin Co., Boston, 1978.

[33] J.B. Cooper, R.S. Newbower, "The Anesthesia Machine - an Accident Waiting to Happen", Human Factors in Health Care, R.M. Picket, T.G. Triggs (editors), Lexington Books, D.C. Heath & Co., 1975, pp. 345-358.

[34] J.B. Cooper, R.S. Newbower, "A New Anaesthesia Delivery System", Anesthesiology, vol. 49, 1978, pp. 310-318.

[35] R. Forest, Y. Lamarche, "An Anesthesia Circuit Monitor", Anesthesiology, vol. 50, No. 5, April, 1979, pp. 356-359.

[36] D.H. Petty, E.A. Ernst, G. Blackwell, "Special Purpose Timer to Aid in Closed Circuit Anesthesia Delivery?, IEEE Trans. on Biomedical Engineering, vol. BME-25, No. 8, August, 1979, pp. 493-495.

[37] D. Maynard, P.F. Prior, D.F. Scott, "Device for Continuous Monitoring of Cerebral Activity in Resuscitated Patients", British Medical Journal, vol. 29, November 1969, pp. 545-546.

[38] M.A. Branthwaite, "Prevention of Neurological Damage During Open Heart Surgery", Thorax, vol. 30, No. 3, June, 1975, pp. 258-261.

[39] R.F. Cucchiara, F.W. Sharbrough et al., "The Electroencephalographic Filter-Processor as an Indicator of Cerebral Ischema during Carotid Endarterectomy", Anesthesiology, vol. 51, No. 1, July, 1979, pp. 77-79.

[40] M.S. Schwartz, M.P. Colven, P.F. Prior, L. Strunin, B.R. Simpson, E.M.J. Weaver, D.F. Scott, "The Cerebral Function Monitor", Anaesthesia, vol. 28, 1973, pp. 611-618.

[41] G.A. Volgyesi, "A Brain Function Monitor for Use During Anaesthesia, Preliminary Report", Canadian Anaesth. Soc. Journal, vol. 25, No. 5, September, 1978, pp. 427-430.

[42] R.A. Fleming, N.T. Smith, "An Inexpensive Device for Analysing and Monitoring the Electroencephalogram", Anesthesiology, vol. 50, 1979, pp. 456-469.

[43] J.D. Frost Jr., D.E. Hillman, P. Kellaway, "Automatic Interpretation of EEG Analysis of Background Activity", Computers and Biomedical Research, vol. 13, 1980, pp. 242-257.

[44] J.S. Brodkey et al., "The Role of Computers in Neurosurgery", Clinical Neurosurgery, Williams & Wilkins Co., Baltimore, 1975, pp. 432-442.

[45] R.A.F. Pronk, "Peri and Post operative Computer Assisted Patient Monitoring", WHO Project, Ref. No. EU/NETH/HMD/099/RB/78/1, Report M-1978-1, September, 1978.

[46] E.A. Gofton, "EEGAL-Mk-4 Integrating EEG Monitoring With the Clinical Anaesthetic Record", Dept. of Anaesthesia, Vancouver General Hospital, 1977.

[47] R.A.F. Pronk et al., "The Use of EEG for Patient Monitoring During Open Heart Surgery", Computers in Cardiology, IEEE Computer Society, 1975, pp. 77-82.

[48] H.H. Jasper, "Report of the Committee on Methods of Clinical Examination in EEG", Electroenceph. & Clinical Neurophysiol., vol. 10, 1958, pp. 370-375.

[49] W. Gerch et al., "A Kullback Leibler-Nearest Neighbor Rule Classification of EEGs: The EEG Population Screening Problem, An Anesthesia Level EEG Classification Application", Computers and Biomedical Research, vol. 13, 1980, pp. 283-296.

[50] J.W. Cooley, J.W. Tukey, "An Algorithm for the Machine Computation of Complex Fourier Series", Math. Comput., vol. 19, April, 1965, pp. 297-301.

[51] P.D. Welsh, "The Use of FFT for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms", IEEE Trans. Audio & Electroacoust., vol. AU-15, June, 1967, pp. 70-73.

[52] C.K. Yuen, "A Comparison of 5 Methods for Computing the Power Spectrum of a Random Process Using Data Segmentation", Proceedings of IEEE, June, 1977, pp. 984-986.

[53] G. Dumermuth, E. Keller, "EEG Spectral Analysis by Means of FFT", Automation of Clinical EEG, P. Kellaway, I. Petersen (editors), Raven Press, New York, 1973, pp. 145-159.

[54] D. Childers (editor), "Modern Spectrum Analysis", IEEE Press, New York, 1978.

[55] C.M. Rader, "An Improved Algorithm for High Speed Autocorrelation with Applications to Spectral Estimation", IEEE Trans. on Audio and Electroacoustics, vol. AU-18, No. 4, December, 1970, pp. 439-441.

[56] A. Peled, B. Liu, "Digital Signal Processing Theory Design and Implementation", John Wiley & Sons, 1976, pp. 156-168.

[57] D. Childers and R. Durling, "Digital Filtering and Signal Processing", West Publishing Co., 1975.

[58] R.G. Bickford, T.W. Billinger, N.I. Fleming, L. Stewart, "The Compressed Spectral Array (CSA), A Pictoral EEG", Proceedings, San Diego Biomed. Symposium, 1972, pp. 365-371.

[59] A. Bricolo, et al., "Clinical Application of CSA in Long Term EEG Monitoring of Comatose Patients", EEG and Clin. Neurophys., vol. 45, 1978, pp. 211-225.

[60] R. Burgess et al., "Cerebral Monitoring by Compressed Spectral Array", New Zealand Medical Journal, vol. 86, No. 601, December 14, 1977, pp. 521-523.

[61] R.A. Fleming, N.T. Smith, "Density Modulation - A Technique for the Display of Three-Variable Data in Patient Monitoring", Anesthesiology, vol. 50, No. 6, June, 1979, pp. 543-546.

[62] W.J. Levy, H.M. Shapiro, G. Maruchak, E. Meathe, "Automated EEG Processing for Intraoperative Monitoring: A Comparison of Techniques", Anesthesiology, vol. 53, September 1980, pp. 223-236.

[63] I. Overington, "Vision and Acquisition", Crane Russak & Co. Inc., New York, 1976, pp. 236-240.

[64] J.L. Demer et al., "An Esophageal Multiprobe for Temperature, ECG, and Heart and Lung Sounds Measurement", IEEE Trans. on Biomed. Engg., vol. BME-25, No. 4, July 1978, pp. 377-380.

[65] J.P. Karis et al., "A Quantitative Neuromuscular Blockade Monitor", Anesthesia and Analgesia, vol. 59, No. 4, April, 1980, pp. 308-310.

[66] R. Purschke et al., "On-Line Monitoring of Cardiac Output, Arterial Blood Pressure, and Heartrate during and after Cardiac Surgery Using a New Computer (COC)", Recent Progress in Anesthesiology and Resuscitation, A. Arias et al (editors), 1974, pp. 631-634.

[67] J.M. Coe et al., "Computer Assisted Heparin Monitoring", American Journal of Clinical Pathology, vol. 72, No. 1, July, 1979, pp. 74-76.

[68] P. Lauwers et al., "Computer Assisted Monitoring in Intensive Medicine", Acta Anesthesiologica Belgica, vol. 27, No. 3, 1978, pp. 287-305.

[69] J. Ackmann, "A Computer System for Neurosurgical Patient Monitoring", Computer Programs in Biomedicine, vol. 10, 1979, pp. 81-88.

[70] G. Dumermuth, R. Dinkelman, "EEG Data Acquisition and Preprocessing by Microcomputer Satellite System", Computer Programs in Biomedicine, vol. 10, 1979, pp. 197-208.

[71] M.G. Rosen et al., "Use of Computers in the Labor and Delivery Suite: An Overview", American Journal of Obstetrics and Gynecology, vol. 132, No. 6, November 15, 1978, pp. 589-594.

[72] G.N.C. Kenny, C. Schmulian, "Computer Assisted Learning in the Teaching of Anaesthesia", Anaesthesia, vol. 34, 1979, pp. 159-162.