

LARS Contract Report 112978

79-10165

CR-160092

Final Report

Vol. V. Computer Processing Support

by **James L. Kast**

Principal Investigator
D. A. Landgrebe

November 1978

"Made available under NASA sponsorship
in the interest of early and wide dis-
semination of Earth Resources Survey
Program information and without liability
for any use made thereof."

(E79-10165) COMPUTER PROCESSING SUPPORT, N79-21520
VOLUME 5 Final Report, 1 Dec. 1977: - 30
Nov. 1978 (Purdue Univ.) 157 p HC A08/MF
A01 CSCL 09B Unclass
G3/43 00165

Prepared for

National Aeronautics and Space Administration

Johnson Space Center
Earth Observation Division
Houston, Texas 77058

Contract No. NAS9-15466

Technical Monitor: J. D. Erickson/SF3

Submitted by

Laboratory for Applications of Remote Sensing

Purdue University

West Lafayette, Indiana 47907

1. Report No. 112978		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Computer Processing Support				5. Report Date November 1978	
				6. Performing Organization Code	
7. Author(s) James L. Kast				8. Performing Organization Report No. 112978	
9. Performing Organization Name and Address Laboratory for Applications of Remote Sensing Purdue University West Lafayette, Indiana 47906				10. Work Unit No.	
				11. Contract or Grant No. NAS9-15466	
				13. Type of Report and Period Covered Final Report 12/1/77-11/30/78	
12. Sponsoring Agency Name and Address J. D. Erickson/SF3 NASA/Johnson Space Center Houston, Texas 77058				14. Sponsoring Agency Code	
15. Supplementary Notes D. A. Landgrebe was LARS principal investigator.					
16. Abstract <p>Over the past seven years ten agencies and institutions scattered throughout the central and eastern sections of the United States have accessed and utilized the data processing facilities and software located at Purdue University's Laboratory for Applications of Remote Sensing (LARS) for technology transfer and research purposes. This network has been evaluated as both a cost- and time-effective means for training people in the use of quantitative remote sensing and for making new developments available to a wide range of users. During the past year, in cooperation with Johnson Space Center (JSC), work was begun to tie together, coordinate and evaluate the work performed by the key research groups supporting JSC's Earth Observations Division by expanding the LARS remote terminal network to serve JSC's research community. This report summarizes this activity and reviews the results with respect to the potential benefits of a shared data processing environment: reduced costs through centralization, immediate access to research software and data, a common operating environment, joint access to a modular baseline analysis system, and the opportunity for a single program of computer aided training.</p>					
17. Key Words (Suggested by Author(s)) Remote sensing, remote terminal network, technology transfer, data processing network, research, cost-effective, communication, training.			18. Distribution Statement		
19. Security Classif. (of this report) Unclassified		20. Security Classif (of this page) Unclassified		21. No. of Pages	22. Price*

TABLE OF CONTENTS

	<u>Page</u>
1. Introduction	1
1.1 Background	1
1.2 Project Objectives	3
1.3 Approach	4
1.4 Summary Schedule	5
2. Project Status	7
2.1 Computer Capability.	7
System Support	7
Remote Job Entry Station	7
Acquire HASP Capability for the Data 100	7
Upgrade Terminal Communications Software	8
Recommend IBM 2780 Replacement	9
Provide Shared SR&T System to SR&T Sites	11
Support Additional JSC Keyboard Terminals.	13
Establish JSC/LARS Tape Transfer Capabilities.	14
Upgrade Batch Machine Support	16
Provide Accounting by User Group	17
Initiate LARSYSP1 IPL System	18
Transfer IPL System Responsibility to JSC.	19
Create Timelimit per Interactive Session Software.	19
Secure Disk Storage.	20
Improve Efficiency of System Operation	21
Consulting Support	22
Establish Consulting Activities.	22
Provide System Information	24
Implement SR&T News Capability	25
EOD LARSYS Prompting Exec.	26
Make Statistical Packages Available.	27
Exchange Information	28
Investigate Computer Conferencing System	30
2.2 Data Base Management	31
Data Base Design and Implementation.	31
Design Historical Data Base.	31
Select Media and Format of Digital Data Base	32
Obtain Data Storage Space.	32
Receive Landsat Data Bases	32
Catalog Landsat Data in the Segment Catalog.	34
Data Management Software	34
Implement Tape Referencing Capability.	34
Implement Landsat Data Search Capability	35
Implement Data Base	35

	<u>Page</u>
3. Results	37
3.1 Strategic Benefits of a Centralized SR&T Computer System	38
User Access	38
Centralization of Hardware	38
Centralization of Software	39
Information and Techniques Exchange	39
3.2 Tactical Reasons for Selecting Purdue as Host	41
Purdue's Experience	41
Improved Techniques Exchange Capability	41
Supplanting or Augmenting JSC On-Site Computations	41
Begin to Centralize SR&T Computing	41
Maintain Valuable Computational Capabilities	43
Increase Access to Shared Resources	43
Provide JSC Users an Interactive and Responsible Computational System	43
3.3 Specific Accomplishments	45
3.4 Evaluation	47
4. Recommendations	49
4.1 Coordinating SR&T Research Site Access to the LARS Machine	49
Remote Job Entry Station	50
Interactive Color Device	50
Alternative Processing Technique Comparisons	51
Programming Documentation and New Techniques Delivery Standards	51
Local Site Support Responsibilities	52
Establish Training Course	52
Coordination of ERDS and SR&T Systems Development	53
Establish a Research and Development Data Processing Network Steering Committee	53
4.2 Usage Projections	54

LIST OF FIGURES

	<u>Page</u>
Figure 1. Computer Processing Support Detailed Schedule	6
Figure 2. Monthly CPU Utilization Graph	42

LIST OF TABLES

Table 1. Resource Allocation.	24
---------------------------------------	----

LIST OF APPENDICES

	<u>Page</u>
Appendix A. System Loading Problems in a Virtual Machine Environment.....	55
Appendix B. LARSYSPl System Manual.....	60
Appendix C. User Time Limit Option.....	114
Appendix D. Guide for Sensible Computer Usage.....	117
Appendix E. Fortran H Compiler.....	125
Appendix F. LARS Consulting Team.....	127
Appendix G. Software Systems Available on the LARS Computer.....	129
Appendix H. LARS Computer System Configuration.....	132
Appendix I. Agenda for Developing a Modular LARSYSPl IPL System.....	135
Appendix J. Segment Catalog.....	137

Computer Processing Support*

1. INTRODUCTION

1.1 Background (Prior to CY79)

The Laboratory for Applications of Remote Sensing at Purdue University (Purdue/LARS) has developed and maintained an Earth Resources Data Processing System which is used by LARS personnel and remote users at Johnson Space Center's Earth Observations Division (JSC/EOD) and other locations. The implementation of LARSYS on a general purpose computer with time sharing and remote terminal capabilities increases the system's value for a large group of users. The resulting system potentially provides:

- * User access, at the users' locations, to remotely sensed data and processing capabilities,
- * Centralization and sharing of expensive portions of processing hardware at a cost advantage,
- * Centralization of software allowing flexibility in software maintenance, addition, and updating at a cost advantage over independent systems, and
- * Ease of training users and sharing experiences through standard data formats, terminology, and shared communication channels.

The Earth Observations Division is planning to install hardware for an Earth Resources Data System (ERDS) in the early 1980's timeframe. The ERDS system must be designed to support world-wide coverage for a multi-crop food and fiber program while allowing the processing flexibility necessary for a research and development environment. In addition, the system must be conceptualized and tested over a period expected to have a very limited earth resources budget.

Both hardware and software for the ERDS system must be modular for purposes of development and expansion. Subsystems should execute independently where possible. The most advanced, proven technology must be employed. The system should be effective and flexible with an easy to use, readily available, user interface.

*The work in this report was done under task 2.4 Computer Processing Support

The LARS data processing facilities provide JSC with a test bed for ERDS techniques development:

- * Examples of modular software systems exist in the forms of LARSYS, LARSYSDV, LARSYSXP, EXOSYS, etc.
- * The capability for independent software subsystems has been developed.
- * EOD and LARS both possess proven and advanced processing techniques; drawing from the best techniques available at both organizations should allow the formation of optimal processing software for ERDS.

Early in the contract year, the decision was made to upgrade the LARS software/hardware facilities at JSC to provide Procedure 1 (P1) processing capabilities and increased terminal support. This upgrade was intended to:

- * Improve the capability for techniques exchange (such as P1 or ECHO) between the two organizations,
- * Relieve RT&E computational constraints by supplanting or augmenting on-site computing as JSC processing capabilities are implemented at LARS,
- * Reduce total costs by improving the efficiency of computer operations,
- * Maintain valuable computational capabilities supporting LARS and JSC research and development, and
- * Increase access to useful resources for both organizations (data, technology, processing systems, hardware, etc.).

Work toward providing a capability to mutually exchange remote sensing data processing techniques between NASA/JSC and Purdue/LARS had begun prior to this contract year. The state of the exchange efforts as of December 1, 1977 was documented in the "Final Report on Processing Techniques Development" of NASA Contract NAS9-14970, dated November 1977.

During December of 1977 the decision was made to significantly expand the scope of and resources available for Task 2.4 in order to:

- * Maintain a computational facility at Purdue/LARS,

- * Begin to actually centralize SR&T computing, and
- * Provide JSC/EOD users access to a more interactive and responsive computational system than was available at JSC.

At that time it was evident that expanded information exchange, hardware development and consulting activities would be required in order to support these objectives. Consequently an agreement was reached to increase personnel drawing on the funds available in the contract, while a cost extension to the contract was negotiated for later in the year.

1.2 Project Objectives

During May discussions were held with a representative (M. Trichel) of the Technical Contract Monitor and with the Operations Branch Chief (D. Hay) and others to establish long range objectives for the computer processing support task. At that time the following umbrella objective was identified:

The LARS computer should serve as the centralized prototype for ERDS; as such, as much candidate ERDS software as is reasonable should be integrated into the LARS system.

Adjunct to this objective were the following goals:

1. The current and historical SR&T data base (field measurements, current and historical imagery, and blind site ground truth) should be maintained at LARS at least through 1981.
2. Certain analysis system software which has been established as "standard" (P1, P2, others) should be maintained on the LARS computer.
3. LARS should support the active use of JSC system software, such as FLOCON, Accuracy Assessment, System Verification, etc. Data distribution and packing activities should be supported at LARS.
4. By 1981, the LARS system should be maintained as the key SR&T research computer.
6. The LARS computer should provide computation for on-going experiment support.

1.3 Approach

The goals and objectives for Computer Processing Support Task can be organized into Computer Capability Support and Data Base Management Support.

Computer Capability.

Computer capability support is comprised of those tasks and activities which are required to supply JSC/EOD and other SR&T sites a data processing environment designed for the support of remote sensing technology through a facility including computer and related hardware, software, procedures, training and support personnel. Specific computer capability support tasks may be categorized as either systems hardware and software related (Systems Support) or as assistance oriented (Consulting Support).

Systems Support. To provide the users access to the applications software listed in goals 2, 3 and 6 of Section 1.2 above, and to prepare to provide the other SR&T sites access to the LARS computer, certain system hardware related tasks are required. In addition, certain system software alterations are necessary for effective use of the JSC/EOD applications software.

Consulting Support. For the effective use of the LARS computer system, it was recognized that information would need to be provided in a dynamic manner about a wide range of subjects. In addition, examples of certain System features and information exchange sessions outlining the use of software systems and computer utilities were recognized as valuable. Consequently, personnel were assigned to disburse and/or secure information about software systems, computer resources, computer products, programming support and training activities. The tasks associated with System Consulting provide the information and assistance necessary to achieve goals 2, 3, 5 and 7 of Section 1.2.

Data Management.

In order to apply the integrated analysis software becoming available on the LARS computer to useful analysis problems, data must be available for analysis. The volume, need for access, and diversity of the data present a significant data management problem. Since the field measurements

data base already resides at Purdue, and since (to fulfill goal 1) it will be necessary for the current and historical imagery data bases and the blind site ground truth data base to be implemented at Purdue, it makes sense for LARS to support future data distribution (goal 4).

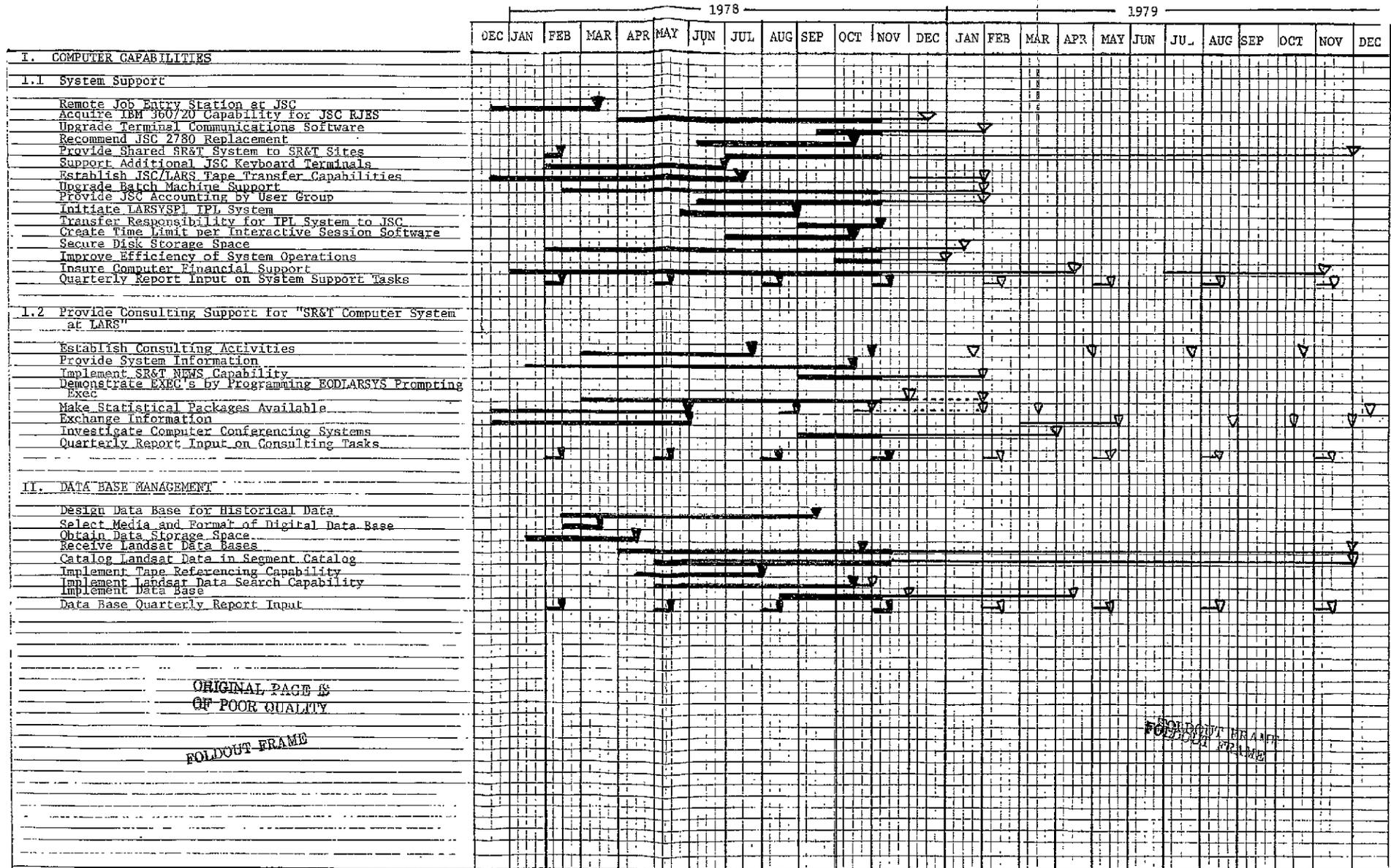
1.4 Summary Schedule

Figure 1 is the 25 month Summary Schedule for major Computer Processing Support tasks and is presented according to the organization outlined in Section 1.3. Since it is LARS' goal to be responsive to additional computational support needs as they arise, it is likely that additional tasks for the computer support activity will be identified and pursued in the future.

FOLDOUT FRAME

TASK 2.4: COMPUTER PROCESSING SUPPORT DETAILED SCHEDULE
Computer Capabilities Milestone Chart

FOLDOUT FRAME



ORIGINAL PAGE IS
OF POOR QUALITY

FOLDOUT FRAME

FOLDOUT FRAME
FOLDOUT FRAME

Figure 1

2. PROJECT STATUS

2.1 Computer Capability

System Support.

Remote Job Entry Station at JSC. In order to support increased usage by personnel at Johnson Space Center (JSC), it was decided prior to the beginning of this contract year to add an additional Remote Job Entry Station (Data 100) at JSC's Earth Observations Division (EOD). During the current contract year, Purdue placed an order for a dedicated phone line capable of 9600 band communications for support of a Data 100 Remote Job Entry Station. Modems, multiplexors, and cables were necessary to interface the phone line with the communications controller at LARS and with the new Remote Job Entry Station (RJE) at Johnson Space Center. Software changes were made in the operating system to support the additional port required for the new RJE station. Hardware and software in the communications controller at LARS were altered to accept and recognize the new Remote Job Entry Station. It was also necessary to alter software in the IBM370 in order to allow users to identify and/or alter which remote site was to receive printer and punch output. When these subtasks were accomplished, it was possible to load the new Data 100 Remote Job Entry Station with a program which caused it to emulate and to access the LARS site as an IBM 2780 Remote Job Entry Station. This task was completed during March 1978.

Acquire IBM 360 Model 20 Capability for the Data 100. The Data 100 acquired by JSC has the capability of functioning as an IBM System 360 Model 20 (HASP) terminal as well as an IBM 2780 terminal. There are several reasons to select the HASP operating mode. For example, the HASP software supports data compression for repeated characters, allows the Job Entry Station to receive and send data concurrently, allows the operator of the Remote Job Entry station to communicate with other ID's on the computer and vice versa, and allows the operator at the RJE to interactively control printer and punch operations. In order to implement the Data 100 as a HASP work station, software in both the communications controller and in the IBM System 370's control program (CP) had to be altered extensively. The Data 100 at JSC began to operate in HASP mode during the month of May.

Upgrade Terminal Communications Software. The software system that supports both the Data 100 (JSCTEXAS) and the old IBM 2780 RJE station (Houston) is the IBM supplied Remote Spooling Communication System (RSCS). Though RSCS has been extensively modified to provide the needed capabilities the desired level of performance has not been achieved. Gains in throughput, capabilities (multiple printers/card punches, simultaneous sending/receiving of two or more output/input streams), reliability, and ease of use will be actively pursued. Throughput is partially a function of the host computer load conditions. When system load is high, the RJE stations may often have to wait for responses from RSCS. Limitations in the RSCS software caused remote sites to suffer from severe input/output backlogs during periods of heavy usage during August and September.

In an effort to improve the communications software, RASP (a replacement for RSCS provided by the IBM user's group, SHARE) was obtained and examined. Although RASP has several nice features, it requires numerous alterations to CP370 (alterations which must be repeated every time IBM makes a new release of the operating system) and retains many of the design flaws of RSCS (See Appendix A). It was therefore decided to implement Release 5 of RSCS and make efficiency alterations to it. RSCS Release 5 was implemented with some significant improvements during late September. The remainder of this subsection discusses further changes to RSCS which are currently being pursued.

Under heavy load conditions a major delay is in waiting to be dispatched, i.e., to get a time slice. Throughput under high load conditions will be improved by decreasing the amount of the two major resources required for RSCS operation - CPU time and virtual storage. Decreasing the amount of CPU time to transmit a file implicitly requires that the average CPU time per data transfer be reduced. When the average CPU time per data transfer is reduced, the average number of transfers per time slice is increased so that the total number of time slices required to transmit the file is decreased.

The operating system under which RSCS runs, VM/370, is a virtual machine operating system. VM/370 allows a large number of users to concurrently run virtual machines with memory sizes ranging from 4K (4096 bytes) to 16,384K

(16,777,216 bytes) even though the LARS computer has only 1024K of real memory. Each user's memory is divided in 4096 byte units called pages. The contents of a user's memory is stored on disk when there is not enough room in real memory. When a page is needed by the user's program, a paging operation is performed by the operating system to read the page into real memory. If all pages of real memory are in use, an additional paging operation must be performed to write a page from memory to disk in order to create a slot for the incoming page. Throughput improvements will be made to RSCS by decreasing the amount of memory that is not altered. These modifications will reduce the amount of paging that is necessary for RSCS operation. Paging is a very slow operation compared to the rate at which the CPU operates, and a user's program cannot be executed while waiting for a page. By reducing the number of paging operations, the amount of real time and CPU time required to execute a job is significantly reduced. It should be noted that decreases in CPU time and storage requirements will not only improve RSCS throughput, but will make more resources available for other users.

Recommend IBM 2780 Replacement. Since 1972 Johnson Space Center's Earth Observations Division has had access to the LARS computer through an IBM 2780 Remote Job Entry Station. During recent years, this terminal has frequently been down and in need of repair. Terminal technology has also experienced many improvements since 1972. These factors, coupled with the increased usage of the LARS computer by JSC/EOD, made the replacement of the IBM 2780 remote terminal attractive. The first step toward 2780 replacement is the examination of commercially available terminal hardware. Such an examination was pursued by staff located at JSC and was reviewed by staff at LARS. The commercially available hardware has been evaluated by personnel at JSC and the satisfactory alternative reviewed for their impact on the Purdue System by the Manager of Basic Systems at Purdue.

A trip was made to Houston to consult with JSC and LEC personnel about replacement of the IBM 2780 remote job entry system. Four classes of replacement systems were discussed.

1. Direct 2780 replacement systems.
2. Replacement 2780 systems with expanded capabilities.
3. Alternate communications systems.
4. General purpose computing systems.

The direct 2780 replacement systems were equipment supplied by IBM and Decision Data. These systems provide the same capabilities as the current system at approximately the same cost per month. However, they do not provide for expanded capabilities as needs increase in the future. The main advantage is that the replacement equipment would provide higher reliability.

Replacement Systems with Expanded Capabilities.

These systems are specialized programmable systems which can emulate the IBM 2780 protocol. In addition these systems can be coded to emulate other protocol such as the HASP workstation. These systems provide many more capabilities such as multiple devices including magnetic tape at 800 and 1600 bpi. Typical vendors are Data 100 and Harris Communication Systems. These systems can also operate at line speeds of up to 900 bpi.

Alternate Communications Systems.

These systems take an entirely different approach to the remote job entry communications problem. The hardware system is separated into two parts. The first connects directly to the IBM byte multiplexor channel and appears to the computer as a lineprinter, card reader, console, etc. The data sent to the present device is compressed into a transmission block and when the transmission block is full, it is sent to the remote control unit. There the block is decompressed and sent to the physical device. Note that all of the communication protocol and code is independent of the main frame. Also note that the system is inherently full duplex using an IBM SDLC or CCITT type protocol.

The manufacturers investigated were Paradyne and its PIX II system and DATA POINT. The Paradyne system had the additional advantage that remote magnetic tapes would be drives as if they were directly connected

to the main frame. Also, the Paradyne system would operate up to 96 KB lines.

The Data Point system operates in a slightly different manner. The local unit connects directly to the byte multiplexor channel and emulates a reader or printer as in the PIX II system. However, all of the print file is spooled to disk before it is transmitted to the remote site. Similarly, the remote reader file is spooled to the local disk unit before it is transmitted to the main frame.

General Purpose Computers.

These systems are medium to large minicomputers such as the DEC PDP II series, the HP3000 and the Data General Eclipse. These systems can be programmed to emulate any of the basic protocols used by the RSCS system, (2780 and HASP workstation). However, unless demonstrably correct emulation programs are supplied by the vendor, the JSC and/or LEC personnel must supply the expertise to write and debug the protocol drives. However, if the drives are available or can be written easily, these systems would provide a local editing and program execution capability which would extend the EOD task capabilities.

The choice of systems was narrowed to a Data 100 system which paralleled the existing Data 100 at JSC and a Paradyne PIX II system. The probable choice will be a Data 100 M76 system with a peripheral switch to move the punch and tape drive between the two Data 100 systems.

Provide Shared SR&T System to SR&T Sites. One of the goals listed in Section 1.2 for this project is for the LARS computer to serve as a primary SR&T research computer after 1981. Many possible advantages would accrue if the SR&T community shared data processing facilities for research and

development:

- * Reduced total operating cost by improving the efficiency of operations through synergy,
- * Improved inter-organization communication of problems and technical advances,
- * Maintenance of valuable capabilities supporting remote sensing research and development,
- * Increased access to such useful resources as data, hardware and software by selected SR&T research sites,
- * A common system background for comparative tests of algorithms.

The first step was to identify a list of candidate SR&T sites which might benefit from access to the LARS computer. Once this was done, computer ID's were placed on the LARS system for each of these sites and information about how to access the LARS system via a dial-up keyboard terminal was made available for distribution to these sites by JSC as needed. These activities make access to the LARS computer by remote SR&T sites possible on a limited basis. Texas A&M University and Fort Lewis College began to make use of the LARS system during the third quarter. Documentation on the use of the LARS system has also been sent to the Environmental Research Institute of Michigan (ERIM).

In order for remote sites to make more extensive use of the LARS computer, more extensive communication systems, computer hardware and support software are necessary. To minimize the expense of such hardware and to identify the possible benefits of the LARS system to each of the candidate SR&T research sites, LARS personnel are contacting the list of candidate sites, discussing what equipment is used and available at the remote sites, and evaluating the suitability of each site's equipment as a remote terminal to the LARS system.

Various communications services and networks will also be investigated so that the cost of transmitting data between remote sites and the LARS site may be minimized. JSC personnel have transmitted information about TELENET to LARS. The availability and expense of satellite communications are being explored. General Telephone and Electronics Corporation made a presentation at LARS during late August outlining system considerations,

communication capabilities and costs of satellite communication systems. Once these investigations are complete, equipment needs for each individual SR&T research site will be identified and, for those sites which a terminal link to the LARS system seems feasible and beneficial, individual plans will be submitted to JSC as separate proposals.

Training programs in the use of the LARS system and the specialized remote sensing software available on the LARS system will be developed in conjunction with JSC. Upon approval of each individual remote site proposal, equipment will be ordered, the newly developed training program will be administered to the new users, and equipment will be installed at the prospective site.

Support Additional JSC Keyboard Terminals. The expanded JSC usage of the LARS computer called for not only an additional Remote Job Entry Station (discussed above), but also expanded use or access through additional keyboard terminals. The first step in supporting additional keyboard terminals was to formulate a terminal expansion plan. This plan identified how hardware at LARS would have to be altered, the number of additional JSC terminals to be supported, and the time schedule for the implementation of the additional terminals. This plan revealed that the most cost effective way to support the additional JSC keyboard terminals called for the replacement of a portion of the communications controller, a line interface base (LIB), which could support only IBM 2741 terminals, with a new LIB capable of supporting a variety of keyboard input terminals, but no IBM 2741 terminals. LARS ordered the new line interface base, keyboard terminals to replace Purdue's 2741's, and certain other hardware components needed by the communications controller. Software in the communications controller and the 370 was altered to support the new terminal configurations at LARS and at JSC, cables, line sets, the line interface base and 2741 replacements were installed; and LARS personnel hosted a visit by Glen Prow of Lockheed Electronics Corporation for the installation and checkout of the modem and multiplexor which supported the additional JSC keyboard terminals. One additional dedicated terminal and four additional dial-up ports were installed and working by the end of June.

Establish JSC/LARS Tape Transfer Capabilities. Prior to this contract year, tape data such as the Landsat data for LACIE segments had to be shipped to LARS via the U.S. Postal System. A JSC user wishing access to a tape located at JSC, but to be analyzed on the LARS system, had to wait from 3 weeks to a month for that tape data to become available at Purdue. In order to reduce this wait time, the Data 100 Remote Job Entry Station installed at JSC included a tape drive. At LARS, documentation on the characteristics and use of the Data 100 tape drive and the steps the Data 100 operator had to perform in order to use the drive was secured. Software was designed to support transfers of tape data between the two sites and an initial implementation of a tape transfer capability was programmed. Tape transfer software for the initial implementation was successfully demonstrated during the third quarter and is currently being documented. During programming of the tape transfer capability, many bugs were uncovered in the Data 100 load program.

The tape transfer software is presently being used in a quasioperational mode. Procedurally, the transfers are being handled through Glen Prow at JSC. A user wishing to transfer a tape either to or from LARS would take an input or output tape to Glen along with a control card deck. The Data 100 operators at JSC are being trained in order to help users in setting up their control cards. Copies of the output are automatically sent to Glen Prow at JSC and Mike Collins at LARS to maintain a record of transfers and to allow a convenient method for requesting an update to the authorized ring-in userids.

The experience gained from that initial implementation has led to the identification of several modifications to the Data 100 software which would result in greatly enhanced tape transfer capabilities.

After having completed several successful tape transfers between the IBM 370/148 (at LARS) and the Data 100 at JSC, it is now appropriate to examine the possibilities of expanding the existing capabilities. Presently, a single tape file may be transmitted in either direction and multiple tape files may be transmitted to the Data 100.

In order for the transfer capability to be used operationally some additional Data 100 capabilities would be desirable. First, the ability to position the output tape on the Data 100 is needed. This could be implemented by adding an FF=nn parameter to the '..MT OU=60' PCL card, where nn would be the number of tape files to skip, the default would be zero. Second, the ability to transmit multiple tape files from the Data 100 without operator intervention and without sending the entire tape is desirable. This could be implemented by adding an NF=mm parameter to the '..MT OU=60' PCL card, where mm would be the number of contiguous tape files to be sent, the default would be one. This parameter would function in a manner analogous to the ET parameter. Third, in both the ET and the proposed NF parameters, CC records of zeros should be sent when intermediate tape marks are encountered. Also, two CC records of zeros should be sent when a double end-of-file is encountered (this could happen prematurely on the NF parameter). A summary of these Data 100 enhancements are as follows:

- a) FF=nn parameter added to the '..MT OU=60' PCL card to enable positioning of an output tape.
- b) NF=mm parameter added to the '..MT EX=60' PCL card to enable selection of contiguous intermediate files to be transferred.
- c) CC records of zeros to be transmitted when intermediate tape marks are encountered.
- d) Two CC records of zeros to be sent when a premature end-of-tape is encountered.

The above proposed changes would greatly enhance the capabilities of Data 100 and provide a tape transfer capability which could be used operationally.

The above enhancements would still rely on the Data 100 operators to initiate the transfer properly. To avoid human errors it would be ideal if the tape transfer software on the 370 could control the tape operation on the Data 100. In this manner the only requirement of Data 100 operators would be to mount the desired tape on the Data 100. To a small degree this is already done on a transfer to JSC; along with the tape, data control information may be sent to write tape marks and to rewind the tape. For this to be

done in a transfer to LARS would require a major modification to software running on the Data 100. This is one of the possibilities that will be investigated in the coming months.

If JSC should decide to pursue these alterations with Data 100, an upgrade of the LARS produced tape transfer software would be made and documented.

Upgrade Batch Machine Support. JSC usage appears to differ somewhat from Purdue usage of the computer system in terms of the mix of system resources required by the individual users. It was, therefore, likely that batch machine requirements for support of JSC users were somewhat different than the batch machine characteristics needed to support LARS users. A preliminary assessment of JSC batch needs based on input from JSC users was made in February. Among the batch needs of JSC users which were not supported by the available LARS batch machines was the need to run CMS370 in batch mode. In order to accommodate this need, a pseudo batch machine was quickly implemented on ID JSC370 specifically for the support of LARSYSP1 operating under CMS370 in batch mode.

During the third quarter a priority rate batch machine (BATEOD) was implemented to support CMS370 and the 2 megabyte machine size often needed by JSC users. The CMS370 batch machine charged the basic CPU rate (BATLONG) was expanded from 512K to 2M of core to support JSC SPSS batch requirements. A specialized batch machine (TAPTRAN) was implemented to specifically support tape transfers between JSC and LARS. A new batch machine (BATJSC) was added to support the SR&T research community. This machine qualifies for the lower cost basic CPU rate (like BATLONG). Experience with JSC use of these batch machines (BATEOD, BATLONG, TAPTRAN, and BATJSC) has led to recommendations for future batch upgrades and additions.

In order to allow JSC users to access the programs stored on their 191 disks, all minidisks associated with JSC ID's were assigned read passwords. Bill Shelley and Sue Schwingendorf both dedicated portions of their visiting consultant trips to presentations on the use of batch machines and helping individual users to run batch jobs.

It is important for JSC users to be able to put any printer and/or punched output in a hold status. This is easy for the user to do when running in interactive mode, but not so easy when running batch. Consultants at LARS are currently able to supply JSC users with the list of commands necessary to place batch output in hold status. This method is awkward, however, and a 'hold' option is currently being implemented for inclusion on the BATCH OUTPUT card in the standard BATCH MACHINE definition deck which precedes each batch job. Not only will users be able to specify hold status when the option is available, but also the number of copies of output.

Since the initial assessment of JSC batch needs, usage of the LARS terminal via JSC has increased in volume and scope. In addition, it was anticipated that additional SR&T sites would eventually be using the LARS system. Therefore, batch requirements for SR&T use were re-examined as was overall system batch requirements for the entire LARS system.

New systems batch capabilities were recommended to the systems group within the LARS Computer Facility. Some important recommendations for BATJSC updates included error reporting for invalid batch job decks, pinpointing the bugs in EDIT when used during BATJSC, remedying the problem of SPOOL modes not being reset before a job is executed, and a procedure to inform users of the reasons for cancellation when it has become necessary to cancel a batch job.

During October and November of 1978, LARS is running an experiment providing users with CPU time which may only be used during the third shift and on weekends. One objective of this experiment is to identify problems with the computer use during these periods including problems with the batch machines. Upon completion of this experiment, additional data on needed batch upgrades should be available.

Provide JSC Accounting by User Group. The Computer Processing Support Task (SR&T Task 2.4) consumed over 1/3 of the total LARS computer usage between June and October of 1978. This usage is not expected to decrease. The community of users under this task is wide and varied including users

from Purdue, Lockheed, NASA, IBM, Texas A&M, and Ft. Lewis College, Colorado. The systems accounting programs available at LARS provide accounting information only on a project basis. In order to better understand, monitor and control usage by the diverse user community supported under SR&T Task 2.4, more detailed accounting information was needed. To secure this information, the various user groups supported by this project were identified. A CPU accounting program has been written to provide user groups specific information and a list of recipients of this report has been formulated. This program is assigned an operator and run weekly. After this program has been upgraded to supply some additional information, it will be integrated into the weekly accounting activity that takes place at LARS.

When the program has stabilized, JSC will be asked to maintain the file defining the user groups and associating user names with specific user ID's.

Initiate LARSYSPl IPL System. The initial implementation of the EOD LARSYS and Procedure 1 software on the LARS machine required 2 megabytes of core to execute. Because of the way LARS batch machines operate, it was very difficult to run a LARSYSPl job in a batch environment. A LARSYSPl IPL system has been implemented by LARS personnel with core requirements reduced from 2M to 768K. To support the IPL system, disk space had to be secured and those load modules required by each separate EODLARSYS processor identified. Changes to the 370 control program (CP) were necessary for the IPL system to be recognized. IPL system characteristics and maintenance requirements are documented so that the LARSYSPl IPL system may be maintained by personnel at JSC. The programmer documentation includes instructional information for generating the overlay modules along with EXEC routines to simplify the updating process.

Setting up a LARSYSPl IPL system has three benefits for system users. The first benefit is that by using the IPL system, the user is insured of using the current version of the prompting EXEC and the EODLARSYS software. The second benefit is that the IPL system helps reduce the total system load and thus indirectly benefits the users with improved response and reduced CPU times. This second benefit was accomplished by generating overlay modules (See Appendix B) for the EODLARSYS software and thus

reducing storage requirements from 2M to 768K. An added advantage of the overlay structure is that storage requirements are determined by the largest processor versus the sum of the sizes of all processors. When additional processors are added to the IPL system, the core requirements will remain constant whereas a system which loads all available processors would continue to grow with each additional processor. The third advantage is that by using modules, rather than text decks, the CPU and response times required to load a processor are reduced and only those modules (processors) specified in the job stream are loaded.

Transfer IPL System Responsibility to JSC. LARS led the effort to put the LARSYSP1 software into a modular IPL system because of its intimate knowledge of the Purdue computer system and because of its years of experience with modular systems such as LARSYS, LARSYSDV, LARSYSXP, EXOSYS, etc. JSC has been controlling several activities to upgrade the P1 system performed by LEC, IBM, Texas A&M, and ERIM. To perform meaningful comparisons of alternative techniques and to minimize the resources required to transfer new technologies and integrate them into an active processing system, all new analysis software should be implemented as a modular portion of the IPL system. Since JSC is managing the upgrades to the processing system, it should also be in a position to provide those contractors working on the upgrades with the specifications to the baseline system, and, where necessary, alter the baseline system to accommodate additional system resource demands required by the system upgrades.

During early November, Pat Aucoin and Kitty Havens of LEC visited LARS to receive detailed training and documentation (See Appendix B) on the use and maintenance of LARSYSP1 as a modular IPL system. Dave Davenport of IBM was also present to learn how to make his Procedure 2 system a modular, compatible adjunct to the baseline LARSYSP1 IPL system. At the conclusion of their visit, Pat Aucoin assumed responsibility for the maintenance of the IPL system.

Create Timelimit per Interactive Session Software. Several JSC users have encountered "bugs" while their programs were executing in a disconnected mode. The result of this problem has been that several jobs have run away, consuming

multiple hours of CPU time. In order to prevent this problem, a way to limit the amount of CPU time which may be used by a single job has been created. The software that was created, TIMELIMIT, allows the user to specify the maximum amount of (virtual) CPU time that a job may use before execution is halted and an error message is printed. TIMELIMIT works only under CMS370. Should the user be in disconnect mode, his machine will log out after the error message is printed. If the user is running in connect mode (logged in at a terminal), CMS370 will be re-IPled after the error message printed. For more information on the TIMELIMIT software, see Appendix C.

Secure Disk Storage Space. The increase in usage and users on the LARS system has resulted in an increased need for disk storage space. As a general rule, installations consider 60 to 80% utilization of disk storage on a permanent basis as acceptable with long term usage of over 80% as over utilized. During the past nine months, the utilization of disk storage space at LARS has exceeded 95%. In addition, the IBM 2314 disk system has been experiencing some reliability problems.

The users have been polled as to their immediate needs for additional space and their expectations for the next year. A need for 100 megabytes as soon as possible with additional usage of up to 200 megabytes over the next year has been expressed.

Purdue has leased two CDC 33302-11 disk drives and a 33332-1 channel adapter unit. These CDC disks are the equivalent of IBM 3330 mod II disk systems. The 33302-11 disk drives have a usable capacity of 170 megabytes each in the format used by CMS. The installation is expected to be complete by January 1, 1979. If more space is needed, additional drives can be added given a four to six month lead time.

Additional disk space is expected to reduce certain system loading problems which were experienced during the heavy usage period of July, August and September. Currently the IBM 3350 disk drive which serves as the system's paging device also supports a number of user minidisks. When the new disk packs are installed, this source of potential disk contention will be removed, hopefully reducing the time required for system paging operations.

Improve Efficiency of System Operation. Several steps have been taken to improve the overall efficiency of system operations for all users:

- * A set of user efficiency guidelines has been published (See Appendix D).
- * Additional disk space is being acquired to help reduce contention on the paging device.
- * The FORTRAN H optimizing compiler is being acquired to reduce paging and make more efficient use of the CPU (See Appendix E).
- * The LARSYSPl software system has been implemented in a form requiring .75 megabytes rather than 2 megabytes of storage.

There are also certain activities which are currently being pursued which will help to further increase the efficiency of the LARS system. An experiment is being conducted to help identify problems associated with third shift and weekend use of the computer. If usage during these periods can be increased, the load during the prime shift would be reduced, leading to increased usable capacity and system efficiency.

A systems analysis package is being acquired from IBM to help identify those uses of the system which contribute to loading problems and the specific hardware components which are acting as system bottlenecks. After identification, steps may be taken to augment or alter systems components serving as bottlenecks and user applications causing loading problems may be examined to identify possible efficiency modifications.

Work is being done to restructure and implement a more efficient version of RSCS. For further details on this activity reference Appendix A and the section entitled "Upgrade Terminal Communications Software", above:

A guide for efficient programming techniques under a paging operating system is being produced and will be distributed after its completion.

Insure Computer Financial Support. One of the contributing factors to the increased usage of the LARS computer by JSC personnel was the need

to maintain the computational facility at LARS. Purdue/LARS will continue to review and communicate its financial requirements (\$550-650K is needed for computer usage during CY79). This will necessitate continuing estimation and communication of funding needs; review of work statement modifications, and responses to request for proposals. To better and more cost effectively support JSC's expanding demand for computer time, LARS has been investigating several special funding options for JSC. One option which may be implemented in the short term would be the sale of the third shift to JSC. If JSC users could effectively utilize the third shift, a very significant savings in their cost per CPU hour would result.

LARS currently rents its computational hardware on a yearly basis because funding for that hardware cannot be guaranteed for a period longer than a year. If funding could be guaranteed for a period of four years or longer, alternatives for more cost effective computational hardware would be possible. Should funding be sufficient it may be possible to purchase a machine with capabilities significantly greater than those of a 370/148 for less than the yearly rental of the 148.

Provide Consulting Support for "SR&T Computer System at LARS"

There are various tasks being carried on that can be classified as assistance oriented: information exchange, consulting, support of user needs, documentation and training.

Establish Consulting Activities: The first step in providing consulting activities to JSC was to assemble a team of experts at LARS to assist JSC users with obtaining resources and training on the LARS computer system. Periodically members of this team of specialists have travelled to JSC to act as consultants to the various users there. This consulting team is also responsible for hosting visitors to LARS from JSC.

A number of LARS specialists are responsible for consulting tasks. The team consists of Bill Shelley, Keith Philipp, Sue Schwingendorf, Luke Kraemer, Jeanne Etheridge, Nancy Fuhs and Ross Garmoe. An up-to-date list of the tasks each of these people is responsible for is included in Appendix F. Consulting comprises a very substantial portion of the work performed under this contract. A good portion of the consulting is done over the telephone. The number of

users of the system and their computer usage reflect the success of our consulting support.

Consulting activities include the bimonthly visits that members of the consulting team have made to JSC. These visits last a week, and the LARS consultant frequently presents or demonstrates various aspects of the computer system as well as consulting with users on a one-to-one basis.

From May 22 - May 26, Susan Schwingendorf served as a visiting consultant at JSC. While there, she presented a seminar on the use of batch machines, demonstrated the EODLARSYS EXEC, discussed the availability of IMSL and assisted new users with CMS370.

During the week of July 10, Bill Shelley and Keith Philipp made a visiting consultant trip to JSC. Items discussed during their visit included LARSYSP1 prompting EXEC, the LARSYSP1 IPL system, how to use batch machines, the tape transfer system, candidate configurations for future terminal hardware.

During the LACIE Symposium at JSC in October, Luke Kraemer consulted with IBM on some problems they were having with their Fortran programs for LARSYSP2.

Consultants also play a role in interfacing the needs of users at JSC with the system resources at Purdue. Between December 1, 1977 and November 1, 1978, 53 new user ID's were added for users at JSC. Nine new ID's were added at LARS for support of the increased JSC usage. Three new batch machines were developed and a Library ID (JSCDISK) was established for storage of the IPL system, IMSL, and other JSC specific library needs. These activities are summarized in Table 1 below.

Table 1

	<u>Dec. '77</u>	<u>Nov. '78</u>
JSC Users	26	71
LARS Support	3	12
Deleted IDs		8
Batch Machines		3
Library ID (JSCDISK)	<u> </u>	<u>1</u>
Total IDs	29	95

Almost all of the JSC ID's were reconfigured at least once during the year (and many several times) to increase disk storage, increase maximum core allocated or to move to the 3350 disks (i.e., convert to CMS370).

Provide System Information. Numerous and varied forms of documentation and information about the LARS computer system are provided to JSC users through the consulting team.

In July, two complete sets of LARS abstracts and an abstract list were sent to Glen Prow. Glen Prow is the contact at JSC for any documentation we have sent or any that we will be sending in the future. He also recently received a 2-page document which briefly describes the software systems available at LARS (See Appendix G) and how to access them. This document refers users to Glen Prow if they wish to look at LARSYS User's Manuals and System Manual, LARS Abstracts, SPSS Pocket Guide, LARS' SCANLINES, the IMSL library document, LARS Computer User's Guide, and a demonstration of graphing capabilities in the EXOSYS software system and the control cards for EXOSYS.

Personnel at Houston have been provided with all current CP/CMS documentation. Ross Garmoe and Keith Philipp tested Release 5 of CP and installed it on the system August 29, 1978. Any new documentation will be sent to JSC as soon as it is received by LARS. The configuration of the LARS computer system is presented in Appendix H.

Several demonstrations have also provided information about the capabilities of the LARS system. A sample Mead output product and cost figures related to that product have been provided to JSC for evaluation of that product's utility to several specific JSC output product needs. A demonstration package for the EOXSYSVDV software system has been prepared and transmitted to JSC personnel. An atmospheric transmission data tape in a Univac object code format has been converted to IBM format for the accuracy assessment group at JSC.

Implement SR&T News Capability. As the use of the SR&T research computer system expands, the need to make information about new developments available to the SR&T community increases. To help meet that need, a special information system (SRTNEWS) is being implemented in CMS370. This system will require support from a single individual at each site working on system related projects. When a new piece of technology or a new capability is ready for access and use by the SR&T community, it should be announced via the SRTNEWS software. This is done through the creation of a special file containing:

- * The article's title
- * The date the article was entered in SRTNEWS
- * The author of the article
- * An abstract describing the new capability
- * References to further documentation and information about the capability
- * A contact person for additional information (preferably contacts at several sites)

An SR&T NEWS software capability was implemented during October 1978, so that any users of the LARS system may obtain information about the most recent system alterations or capability upgrades by typing 'SRTNEWS'. Recent news paragraphs are typed at the user's terminal, ending with a list of titles of older news paragraphs which the user may see by answering appropriate questions asked by the program. If the user prefers to have the news paragraphs printed on the local batch terminal printer, he should call the news facility with:

Make Statistical Packages Available. In order to evaluate the results of demonstrations and experiment related to the development of remote sensing technology, a number of statistical analysis capabilities are necessary. LARS, at the current contract's inception, had a version of the Statistical Package for the Social Sciences (SPSS) and the Bio-medical Statistics Package (BMD). JSC users had requested that the possibility of acquiring the International Mathematical and Statistical Library's (IMSL) and Statistical Analysis System (SAS) be investigated. System requirements for IMSL and SAS were examined resulting in the elimination of SAS as a viable package on the LARS computer because of its incompatibility with the way CMS performs disk I/O.

Early in the contract year, the BMDX85 routines were fixed to allow users to supply Fortran statements as part of the input deck.

IMSL was ordered and received in late April 1978. During May 1978, an existing Fortran program was modified to read the IMSL source tape and separate the IMSL subroutines into individual files on the Purdue/LARS computer. These files were then compiled and the source, listing and text files were stored on tapes 523, 524 and 525 respectively.

A majority of the IMSL routines have single and double precision versions. On the source tape supplied by IMSL, the single precision version is given, with double precision statements "commented out". Since the double precision text files did not appear to be compatible with our system, we edited the source files to provide a text library of double precision IMSL routines. Two text libraries have been created on disk 19E of computer ID JSCDISK for use under CMS370. One contains all single precision subroutines (SIMSLIB) and one contains all double precision routines (DIMSLIB). To use IMSL under CMS370 a person must issue the following CMS commands (from his terminal or EXEC file):

```
GETDISK JSCDISK 19E
GLOBAL TXTLIB CMSLIB FORTRAN SIMSLIB DIMSLIB
```

The text library names SIMSLIB and DIMSLIB should be reversed in the GLOBAL command if the double precision version of a subroutine, which has

both a single and double precision version, is to be used. This was documented in the September 1978 issue of SCANLINES.

There is a maintenance fee that has to be paid yearly to SPSS, Inc. for their statistical package. Because of change in personnel at LARS, this fee did not get paid until the 3rd quarter although it was due in December 1977. This contract paid half of the \$600 fee. SPSS Pocket Guides for Users were ordered at that time and have since been sent to Mike Pore at JSC.

During the second quarter, a problem with SPSS was reported. It was discovered that the discriminant analysis routines do not work on Release 7.1, which runs under CMS370; they do run on Release 6 which runs under CMS360. Release 7.2 was supposed to be converted by the U. of Waterloo for SPSS and put on line at LARS in October 1978. A Purdue representative attended the annual SPSS conference in Chicago during October and found that the conversion had been sick for over 2 months. As a result of the lack of communication SPSS and Waterloo about the conversion effort, both parties have come to a new agreement and feel that Waterloo will be able to effectively perform future conversions. The agreement is that Release 7.2 will not be converted, Release 8A will be converted in 6-8 weeks (by the beginning of January 1979), and any further conversions will only take about 6 days. SPSS talked with Waterloo in detail about the techniques they will use in future conversions and is satisfied that Waterloo can meet this schedule.

During the fourth quarter, JSC users expressed renewed interest in obtaining access to SAS on the SR&T system. After consultation with SAS systems personnel, it was discovered that a CMS compatible version of SPSS was under development and would become available in 4-6 months. The decision whether or not to procure the CMS version will be made when CMS/SAS becomes available.

Exchange Information. In order to provide a general background on the use of the LARS computer, personnel at LARS prepared a short course on the use of the LARS system and CMS370. Since many computer users at

LARS had not yet become familiar with CMS370, a trial run of this short course was conducted at LARS prior to the presentation of the short course at JSC. The course was presented in the first quarter.

During the third quarter, we reorganized the training materials for this course so that it can easily be presented again for new users of our system. In September, part of this course was presented to LARS personnel to encourage use of CMS370. The course is scheduled to again be presented at JSC during January 1979.

In order to keep personnel at LARS informed of activities at JSC and personnel at JSC informed of activities at LARS, the organizations have decided to exchange their respective information bulletins. The Facility Management Bulletin and the Transition Year Weekly Status Review for the Earth Observations Division are currently being distributed to certain members of the LARS staff and posted on bulletin boards at LARS, and LARS' SCANLINES is being sent to a number of JSC personnel.

In order to facilitate communications between Purdue/LARS and JSC/EOD, the Purdue telephone officer has been asked to request an FTS line at LARS. Since he is in the process of dealing with FTS for the entire campus, he is studying Purdue's total FTS need and will make a recommendation at the end of this study. At the present time, he has not given an indication of when the study will be complete or when the FTS line will be requested. It is estimated that this will not occur before January 1, 1979.

During the LACIE Symposium, Luke Kraemer began the exchange of information on a set of Fortran programs which were written to scan information found on header records for LACIE segments. A user can request a list of all segments for a certain date or range of dates, a range of latitudes or longitudes, etc. In January he plans to present a formal demonstration of these programs and talk about plans to expand the data base to include other information. The day after the LACIE Symposium, a short course on EXOSYS was presented. EXOSYS is a set of computer programs used to analyze the spectra and identification records of agricultural field data. The morning session described the Field Measurements Library and EXOSYS. The afternoon session

spent in performing a hands-on demonstration with a small number of users.

During November, Pat Aucoin and Kitty Havens (both of LEC) were at LARS to exchange information. Appendix I is a schedule of their activity during this trip. They received training in the details of the LARSYSPL IPL system so that they could take over its maintenance. A LARSYSPL Systems Manual (See Appendix A) was provided as documentation. The IPL system reduces the amount of virtual core needed to run LARSYSPL from 2 megabytes to 768 kilobytes. The amount of temporary disk storage required by the system was reduced from 25 cylinders to 10 cylinders of 2314 disk space. This was done by implementing Pat Aucoin's new software for ISOCLS on the IPL system. LARSYSPL bugs and efficiency problems were also discussed. CMS programming practices that increase the efficiency of the use of the LARS IBM 370/148 were also discussed. Future data base requirements were reviewed and the advantages of LARSYS support routines MOUNT, CPFUNC, and TAPOP were discussed with the goal of their inclusion in the DATAMERGE function in mind.

Dave Davenport of IBM visited Purdue during the same time frame in order to understand the modular LARSYSPL IPL system so that the Procedure 2 system which IBM is constructing will be fully compatible with and take full advantage of the features incorporated in the Procedure 1 system.

Investigate Computer Conferencing Systems. As more and more use is made of the SR&T research system by the geographically dispersed SR&T community, the need for quick and effective communications will grow. The SR&T NEWS capability is one means of reducing the communications problem. Visiting Consultant trips, Information Exchange Sessions and correspondence and phone calls are others. An additional means of communication which is being examined is the computer conferencing system. If use can be made of the site-to-site computer communications facilities for human communication, some expensive travel may be eliminated.

2.2 Data Base Management

Date Base Design and Implementation.

Design Data Base for Historical Data. The initial data base proposed in the second Quarterly Report has been updated to include additional data fields and files for a better coverage of important information. Major changes in the data base (Appendix J) include new fields in the Segment Index Data File which further define the location of the segments, i.e., Country, State, County, Agro-Physical Unit Description and Crop Reporting District. In the Acquisition List Records, newly added fields will contain such information as Orbit Number, Time Data Collected, Peak Sharpness, Cloud Cover, etc. Two new files, a Crop Names List and a Crop Status List, have also been added to the design. These two files will be indexed by the Dot Label Table; therefore, storage will be saved by elimination of duplicate crop names and their statuses. Probably the most important addition to the design though, is a pointer field which indicates the corresponding CAMS/CAS interface tape data file for each analysis. This pointer is included because the CAMS/CAS interface tape contains abundant information which may be valuable. Rather than consuming disk space to duplicate this data, a pointer was included in the Dot Label File.

Another change in the data base design has been the reorganization of the record layouts. A powerful feature of FORTRAN is the EQUIVALENCE statement. Through this statement, a user can reference a data field of a record directly without performing any complex byte manipulations. In order to use this statement though, certain machine dependent rules had to be followed. These rules demanded that the record formats be slightly modified. Although some of the new record layouts are not as logically ordered as previous designs, the increased computational efficiency and ease of use are expected to overshadow any ordering drawbacks.

Further updates to the data base design, especially in files other than the Acquisition List and the Segment Index, are expected as implementation and testing proceed.

Select Media and Format of Digital Data Base. The computer system at LARS has considerably different characteristics than the computer systems which personnel at JSC accessed for test evaluation and production processing during the LACIE Project. Virtual machines allow much greater flexibility in analysis software packages and their capabilities than was previously available in the Building 12 and the Building 30 complexes. Since virtual machines can work on job X while waiting for data input on job Y (consequently losing little CPU efficiency due to I/O), tapes have always been a suitable medium for large data bases housed at Purdue. The JSC computer systems, in contrast, sit idle while waiting for data input. Hence the large data bases at JSC have been stored on the more expensive medium of disk. The characteristics of the LARS computer system were reviewed with JSC personnel and the decision was reached to store the Landsat segments and wall-to-wall ground truth for the RT&E data base on tape in Universal Format.

Point and ancillary data will be stored on disk according to the format described in Appendix J. Disk was chosen as a medium for variables describing characteristics of the sample segments and segment acquisitions so that the space of segments and/or acquisitions could be rapidly searched by computer in order to locate segments or acquisitions meeting the user defined criteria. This capability is discussed in the "Implement Data Base Search Capability" section below.

Obtain Data Storage Space. In order to house the SR&T data base, additional tape racks had to be secured. Storage space for 720 tapes was ordered and installed in the LARS digital tape library. In addition, additional disk space is required to house the Segment Catalog described in Appendix J. This disk space will become available when the additional disk system goes on line in January 1979.

Receive Landsat Data Bases. To support the research needs of the SR&T community, it is necessary for users to have both processing software, which is available in the form of LARSYSPL, and the data to be exercised by that software. Consequently, the LACIE Phase I, Phase II and Phase III data bases have been unloaded from disk storage in Building 30 and shipped to Purdue. Upon reaching Purdue, tapes are labelled and placed in tape racks. Because the tension on the tape drives at Purdue may differ from tape drive tension

on the drive at JSC which wrote the tapes (degrading the ability of the LARS tape drives to read the tape successfully), the tapes received from JSC are placed on a LARS drive, forward filed to the end of the tape, and then rewound. After labelling and retensioning operations are completed, a letter verifying receipt of the data is sent to JSC for each batch of tapes received.

In March 1979, LARS received the 1977 Phase III data base consisting of 222 tapes. These were assigned slot numbers and placed in the Purdue/LARS Tape Library, with JSC tape 77001 in slot 7001 and JSC tape 770222 in slot 7222.

The 1976 Phase II Data Base was completed at LARS the beginning of June with the addition of tape 76110 which was inadvertently left out of the shipment. Hence, JSC tape 76001 is in LARS tape slot 7223 and JSC tape 76110 is in LARS slot 7332, with intermediate tapes in between.

A serious problem was discovered with some of the data base tapes sent from Houston; they would stick to the tape units. The tapes unloaded from the segment data base at JSC come to Purdue in 800 bpi. The LARS computer system has 9 tape drives which read 9-track tapes. However, only 4 of these tape drives are capable of reading 800 bpi. In order to allow more than 4 users at JSC to concurrently access the data bases available on the LARS system, and to improve the reliability of the tape data, the historical data base has been copied to 1600 bpi tapes.

During this copying process about a dozen tapes were found to be unreadable or blocked incorrectly. Replacements were requested, and received in October. Several of these replacement tapes were still not correct, and new copies have been ordered. The 1975 Phase I data base, consisting of 26 tapes, was also received at LARS in October. Nine of these tapes were desired for immediate use, and have been copied to 1600 bpi tapes. The remainder will be copied when a shipment of blank tapes is received from JSC.

Catalog Landsat Data in the Segment Catalog. The Segment Catalog is a portion of the historical data base which contains reference and ancillary information about the data stored in the data base. As Landsat segment data is received, information about the newly received segment data is to be recorded in the Segment Catalog.

Since the design of the historical data base is virtually complete, implementation of a skeletal version of this data structure for testing purposes was begun. During this testing period, an appraisal of storage requirements was made. Also, the initial implementation of the Segment Catalog allows users of the Landsat data currently at LARS to use the data search and data acquisition software (discussed below) to support their research needs. After the design of the Segment Catalog has been finalized, the test structure will be reformatted to the desired record layout.

Data Management Software.

Implement Tape Referencing Capability. To locate the specific LARS tape and file which contains Landsat data for a specific acquisition of a specific segment, utility subroutines (SEGFO and GETACQ) have been written. These subroutines take segment and acquisition numbers as input and return specific information about those segments and acquisitions (such as the tape and file of the data). These programs are documented and have been demonstrated to JSC personnel.

SEGFO searches a segment index file in order to secure information about acquisition numbers, corresponding tape and file numbers, number of channels, number of columns, latitude, longitude and ERTS scene frame ID for acquisitions of the user's specified segment number.

GETACQ accepts a unit number, a segment number and an acquisition number as input. It then locates the tape and file where the specific acquisition of the desired segment is archived and mounts and positions the tape to that file.

The user may gain access to these routines by typing the following commands in CMS370 mode:

```
'LINK JSC DISK 19E 19E'  
'ACCESS 19E B'  
'EXEC GETACQ'
```

The last command 'EXEC GETACQ', initiates necessary file definitions and links, and accesses the disks which house the segment index information and the necessary text decks.

LARS will maintain the tape referencing software, making alterations as the format of the Segment Catalog is updated.

Implement Landsat Data Search Capability. Algorithms which allow the Landsat segment data to be searched based on a set of user specified input criteria have been completed. This system of routines, known as SUBSET, processes logical expressions which clearly resemble FORTRAN IF statements. As SUBSET evaluates the logical relationships which make up the expression, appropriate attribute sub-files are searched to find any valid acquisitions. Examples of these sub-files are longitude, sun elevation, peak sharpness, and soil line greenness. SUBSET can query the LACIE data base on these and many other attributes and return a list of the desired acquisitions. SUBSET currently can handle expressions with up to 50 logical relationships; however, if there ever is a need to process more complicated enquiries, SUBSET can be easily modified. Not only is SUBSET capable of processing complex expressions but it can evaluate these expressions extremely economically. Most input should be handled in less than a minute of CPU time; whereas, a very simple expression will be processed in a matter of seconds. Documentation of SUBSET is being started now and the entire system will be ready for formal demonstration at JSC during the January CP/CMS short course.

Implement Data Base. In conjunction with the design of the current historical data base, an index was maintained which detailed how and where the necessary data can be accessed. Each field in the data base

must be defined clearly and the source of data for each field must be identified; e.g., Landsat, System, Analyst,... Since the design of the data base is virtually complete and LARS has obtained most of the data necessary to build the Segment Index and the Acquisition List, design and implementation of programs to maintain these two files is complete. In fact, test results for speed and efficiency on these files have been favorable. Acquiring data for implementation of the other files in the Segment Catalog, though, has been more difficult. Since most of the data for these files requires an analyst's input or preprocessed data that doesn't exist, progress has been slow. Until the source for this data is more clearly defined, advancements will continue to be hampered. Designing algorithms to maintain these files without a clear understanding of where the data is coming from and what kind of values the data can have is pointless. A major effort to more fully understand the role of the analyst in maintaining the Historical Data Base appears to be necessary.

3. RESULTS

In section 1.1 of this report four potential strategic benefits of a centralized SR&T computer system were identified:

- * User access, at the users' locations, to remotely sensed data and processing capabilities,
- * Centralization and sharing of expensive portions of processing hardware at a cost advantage,
- * Centralization of software allowing flexibility in software maintenance, addition, and updating at a cost advantage over independent systems, and
- * Ease of training users and sharing experiences through standard data formats, terminology, and shared communication channels.

Based on these potential benefits, the following umbrella objective was established for the Computer Processing Task:

The LARS computer should serve as the centralized prototype for ERDS; as such, as much candidate ERDS software as is reasonable should be integrated into the LARS system.

Tactical reasons for pursuing the Computer Support Task at Purdue/LARS included:

- * Purdue's experience as a host site,
- * Improving the capability for techniques exchange (such as Pl or ECHO) between the two organizations,
- * Reducing RT&E computational constraints by supplanting or augmenting on-site computing as JSC processing capabilities are implemented at LARS,
- * Beginning to actually centralize SR&T computing,
- * Maintaining valuable computational capabilities supporting LARS and JSC research and development,
- * Increasing access to useful resources for both organizations (data, technology, processing systems, hardware, etc.), and
- * Providing JSC/EOD users access to a more interactive and responsive computational system than was available at JSC.

This section will organize results in terms of the evidence they bring to bear on the strategic benefits of a centralized SR&T computer system and on the tactical reasons for choosing Purdue as the initial host site for the system.

3.1 Strategic Benefits of a Centralized SR&T Computer System

User Access.

JSC support contractors making access to the centralized system during this contract year included Purdue/LARS, Lockheed Electronics Company, IBM, Fort Lewis College, and Texas A&M University as well as users at NASA. In addition, documentation and computer sources have been made available to the Environmental Research Institute of Michigan and the University of California at Berkeley. User access statistics are especially impressive since only Purdue and those support contractors located on site at JSC had access to full-fledged remote job entry stations; other contractors accessed the system through dial-up keyboard terminals only.

Centralization of Hardware.

The increased use of the LARS computer has allowed EOD to significantly reduce its computations in Building 12 and Building 30 on site at JSC. There are at least two reasons why the LARS computer has not totally supplanted hardware at these other sites. First, it takes time, money and personnel to convert software. One of the most desirable features of a centralized system is that these expenses can often be avoided once all research sites have adopted the centralized system. However, conversion of all software in Building 12 is not yet complete. Second, certain expensive hardware with specialized capabilities have been married to the Building 30 facility (the Staran and the ERIPS terminals), since Purdue cannot provide equivalent capability to users at JSC without the investment of additional NASA resources. On the other hand, it should be pointed out that the Purdue system does provide a number of unique capabilities necessary for a centralized site which are not available at either Building 12 or Building 30 and which would take a very substantial investment to duplicate. As more of the SR&T user community gains access to this system, the opportunities to centralize hardware will expand.

Centralization of Software.

Several important steps have been taken to centralize research software on the Purdue/LARS computer:

- * LARSYSPI has been implemented at LARS,
- * Development of Procedure II is proceeding on the LARS computer,
- * A LARSYSPI IPL system which will easily accommodate upgrades and additional development techniques,
- * The Texas A&M spatial clustering algorithm AMOEBA has been implemented on the LARS computer in a form compatible with LARSYSPI,
- * Lockheed Electronics is implementing several algorithms in LARSYSPI compatible form.

When these and future analysis system upgrades have been completed, NASA will for the first time be in a position to accurately evaluate and compare the benefits of alternative processing techniques developed at various research sites without a substantial investment to reprogram all alternative techniques or questions about their respective differences being due to different base-line systems. Also, for the first time, the various SR&T research sites are in a position to utilize techniques developed at other sites without prolonged, expensive learning and reprogramming efforts. The availability of field measurements, historical digital imagery and blind site ground truth data bases provide a more powerful array of data sets to all research groups accessing the system than has ever been available to any single site (including JSC/EOD) before the centralization.

Information and Techniques Exchange.

To work together effectively, the various organizations researching remote sensing must be in a position to exchange techniques, ideas, concepts and test results. Such activity requires well developed communication channels. Sharing data processing resources and capability provides a base line and a channel for such communication. Several demonstrations of the power of such communication have resulted from this years activities:

- * After a three-day training course on the use of the computer at LARS, a number of JSC users were able to use the system to obtain the data processing capability they needed.
- * A project at LARS was able to use the analysis system (LARSYSP1) placed on the system by another group at JSC for the analysis of multicrop data.
- * JSC personnel have begun to use a field data analysis system developed and maintained at LARS (EXOSYS) with only minimal training.

All these activities demonstrate the usefulness of centralized software as well as the ability of the shared data processing system to expand the capabilities of those organizations willing to communicate information about capabilities they have developed.

3.2 Tactical Reasons for Selecting Purdue as Host

Purdue's Experience.

Purdue has operated a remote terminal system for the analysis of earth resources data since 1972¹. Johnson Space Center already possessed a terminal to the Purdue system which was used primarily for training of new Analyst/Interpreters and for support of the LACIE Flow Control (FLOCON) analysis tracking system. In addition, since LARS is located in two buildings and operates a terminal in the building which does not house the computer, LARS has experienced the problems of a remote terminal system from the point of view of a remote site as well as the host site.

Improved Techniques Exchange Capability.

The examples cited in the "Information and Techniques Exchange" section above demonstrate the success achieved in this area.

Supplanting or Augmenting JSC On-Site Computations.

While computations in Building 12 and Building 30 at JSC decreased significantly, JSC computer usage at LARS rose from roughly 11 CPU hours in December 1977 to roughly 106 CPU hours in August 1978. During September and October 1978, JSC sought to limit usage of the LARS machine to reduce costs resulting in decreased use during these months. See Figure 2 for a month-by-month breakdown of JSC use of the LARS computer. The cross-hatched area in October represents computer hours used during the third shift experiment by users at JSC (these hours are not charged against the contract). The data indicates that making better use of the LARS computer has allowed JSC to reduce its dependence and expense at other computational installations.

Begin to Centralize SR&T Computing.

This topic is addressed by the User Access portion of Section 3.1 above.

-
1. T. L. Phillips, et. al., "Remote Terminal System Evaluation," LARS Information Note 062775, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana, 1975.

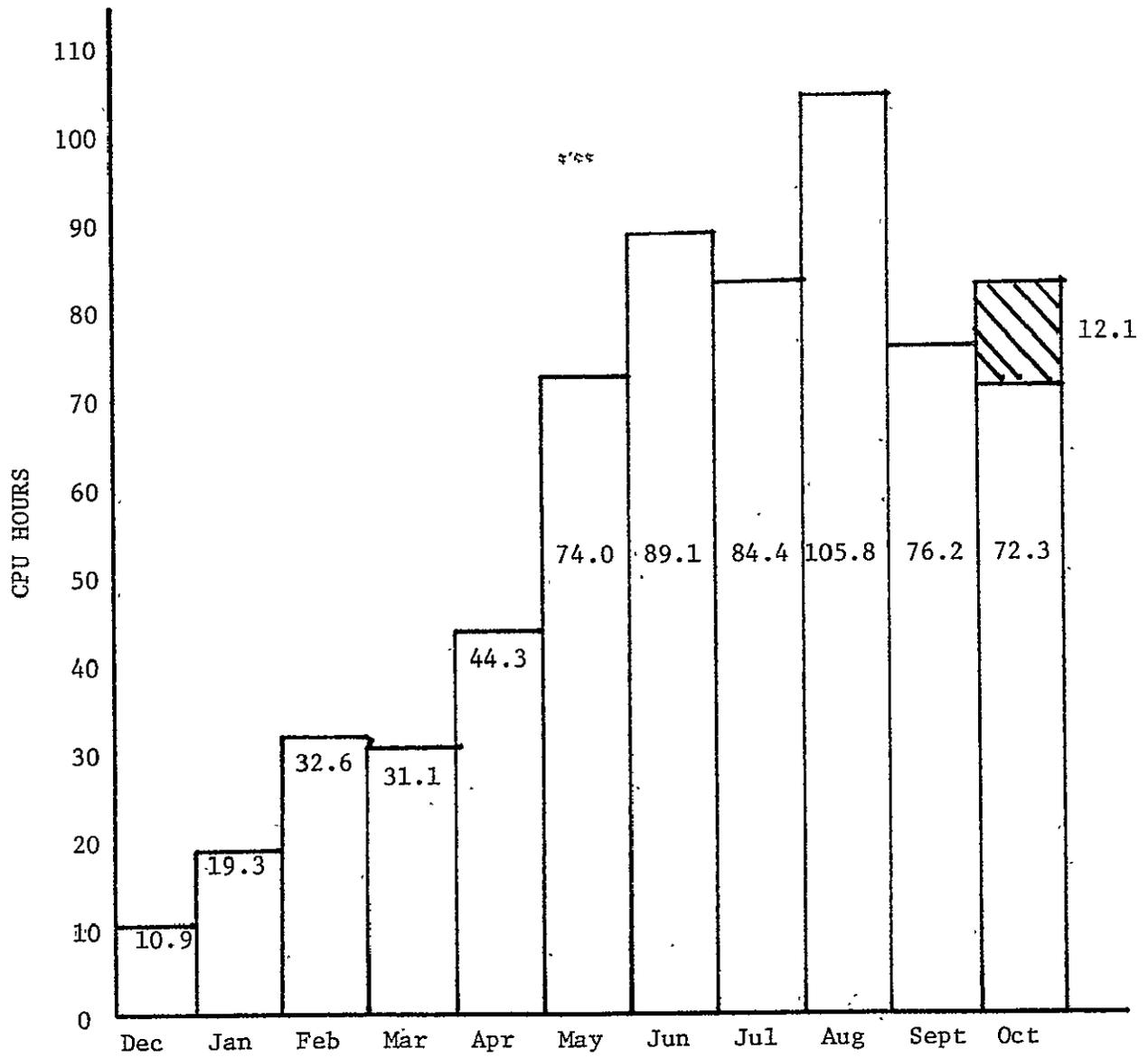


Figure 2
 Monthly CPU Utilization
 December 1977 - October 1978

Maintain Valuable Computational Capabilities.

While personnel and computer cost have been steadily rising over the last two years, funding for research work at Purdue has suffered a slight decline. Had JSC not been able to make significant use of the LARS computer, its continued existence would have been problematical. Without a computational facility much of the research work at Purdue would have been severely impacted. By using the LARS system to supply EOD computational support needs, Purdue's facility has been maintained.

Increased Access to Shared Resources.

Development of the centralized SR&T computer system has made possible access to a wide range of research analysis systems and techniques, data bases, documentation, etc. These benefits are discussed in Section 3.1 above. It should be pointed out that a centralized system supports increased access to shared resources, but that for resources to be effectively shared by all users, very strong communication channels are necessary.

Provide JSC Users an Interactive and Responsive Computational System.

The computer systems in Buildings 12 and 30 at JSC were not specifically designed to support remote sensing research and techniques development. The machine at LARS, on the other hand, was. The Virtual Machine operating system (VM/CMS370) on the LARS computer has several very useful features. Each user is treated as if he were the sole user of a machine (hence virtual machine). This concept allows the definition of machine memory size, devices available to the machine, etc. It allows very interactive processing modes without sacrificing CPU efficiency, processing user Y's job while waiting on input from interactive input by user X. Certain earth resources data processing functions strain the ability of a machine to perform complex computations; others, the ability to handle large, complex data sets; others the ability of human interaction with intermediate results; still others combinations of these capabilities. The LARS computer allows a number of users to simultaneously use the real computer; each user specifying the machine configuration which will best suit the characteristics of his job. Not only can the aggregate of user-defined system resources surpass those actually available on the real machine, but any single user might

define his virtual machine to have resources superior to those of the real machine. This operating system made possible the research and implementation of geometric correction and registration processors on machine only a fraction of the size that would have been required had a more conventional operating system been employed.

During the contract period a number of specific tasks required for the centralization of SR&T computations were initiated and completed:

- * Hardware, phone lines, and communications software were acquired and upgraded to support an additional remote job entry station (Data 100) at JSC;
- * The capability to transfer tapes between LARS and JSC was implemented and demonstrated;
- * Communications hardware and software were altered to support five additional keyboard terminals at JSC;
- * A survey of the remote terminal hardware market was conducted to identify a potential replacement for the aging IBM 2780 hardware resident at JSC;
- * Hardware was acquired and resources were set aside for SR&T research sites (Texas A&M, University of California at Berkeley, and ERIM) to access the Purdue/LARS computer in a dial-up mode;
- * Satellite computer communication data links were investigated;
- * The new remote job entry station (Data 100) at JSC was reconfigured to run as an IBM System 360 Model 20 terminal with resulting improvements in data transfer rates and terminal flexibility;
- * The baseline EODLARSYS Procedure 1 software system was implemented on the Purdue computer;
- * Batch requirements for the SR&T user community were examined and three new batch machines (BATEOD, BATJSC, TAPTRAN) were designed and implemented to satisfy those requirements;
- * The EODLARSYS Procedure 1 software package was configured into a modular IPL system (LARSYSPl) which made effective use of the available computational resources;
- * An accounting program detailing weekly and monthly usage of the LARS computer by Earth Observation Division branches and user groups was designed, written and implemented;

- * Storage media was selected and space was secured for the RT&E data base;
- * Short courses on the use of the LARS computer system, batch capabilities, and on the use of LARSYSPI and EXOSYS were conducted at JSC and LARS;
- * Temp disk storage available to EOD users was increased by 75%, disk space for JSC private minidisks increased from 16 megabytes in January to approximately 90 megabytes in October;
- * Computer usage for SR&T computer support increased from 11 CPU hours in January to 106 CPU hours in August;
- * LARS computer specialists spent three to five days at JSC as visiting consultants on five separate occasions;
- * A prompting EXEC was written to support users of EODLARSYS;
- * The IMSL statistics routines were acquired and made available on the LARS system;
- * The LACIE Phase II and Phase III data bases were received, catalogued, and placed in the LARS tape library; and
- * The initial design of the RT&E data base was completed.

3.4 Evaluation

A preliminary evaluation of the effort to centralize SR&T computing on the Purdue/LARS system indicates that, even after this short period of effort, results support the potential benefits of centralization discussed in Sections 3.1 and 3.2, above. There appear to be three essential ingredients for the success of this effort:

- * Recognition that the shared data processing environment is more than simply a computational resource. Although it is true that centralization of SR&T computing should provide all members of the SR&T community an efficient computational system, it is also true that no single system will be optimal for all user applications. Since most SR&T research sites also have access to local processing systems, there is liable to be some resistance to centralizing research software, if the shared system is viewed only as a computational resource. This is especially true when the characteristics of a local system make that local system a more desirable place for the implementation of a specific processing function. It is very important, therefore, for managers at each SR&T site to understand that the shared system also provides a basis for techniques exchange and for meaningful evaluation of alternative processing techniques. Developing new processing techniques on a local system, rather than on the centralized system, is tantamount to deciding that other institutions should not have access to the newly developed techniques and to deciding that these techniques should not be meaningfully evaluated with respect to processing alternatives.
- * Mutual support by all system users. In order for potential techniques exchange and information exchange opportunities to be realized, each institution making use of the shared system must make a commitment to support the system. A site specialist needs to be developed at each local site to help answer questions and solve problems encountered by

local users of the shared system. The site specialist should also be in a position to refer difficult and exotic problems to those responsible for the maintenance of the system at the host site. In addition, each local site must endeavor to educate users at other sites in the use of promising techniques developed at the local site. This activity will require extensive documentation, demonstrations, presentations, and an omnibusman to respond to questions about newly developed techniques.

- * Establishment of standards. In order to quickly and cost effectively transfer techniques developed at a site to other researchers at other sites, certain standards for program and user documentation must be established. In addition, if software is to be integrated into a single centralized processing system, certain software delivery requirements must be met. This is especially true if the software is to be evaluated against a base-line system. From time to time as new techniques are developed, additional resources will be needed on the centralized processing system. Mechanisms need to be developed for the request, selection, and procurement of additional system resources if the centralized system is to continually provide the raw materials necessary for the construction of improved procedures for analyzing remotely sensed data.

4. RECOMMENDATIONS

4.1 Coordinating SR&T Research Site Access to the LARS Machine

The benefits of centralizing the computational support for SR&T efforts are fairly obvious to those supplying the funds (reduced total costs, improved efficiency); those trying to test and evaluate newly developed technology (immediate access to software and appropriately formatted data, a common operating environment and baseline system for comparisons); and those assembling systems utilizing new algorithms (immediate access to all newly developed software which has been implemented to be compatible with the baseline processing system).

Several major steps have been taken to achieve a centralized SR&T computer system utilizing the Purdue/LARS computer:

1. The philosophical decision has been made to centralize the SR&T computational resources.
2. A baseline analysis system has been implemented at LARS (LARSYS/Procedure 1).
3. An extensive SR&T data base is being shipped to and implemented at Purdue.
4. Computer accessibility for JSC users has been more than doubled, and data transfer capabilities have been greatly enhanced between JSC/EOD and Purdue/LARS.
5. Training courses have been prepared and presented by both LARS and JSC.
6. New software developed by certain JSC contractors (IBM, LEC) is being implemented and tested on the central SR&T computer.
7. Computer ID's and dial-up terminal access is available for other members of the SR&T research community to access the Purdue/LARS system on a limited basis. (Two institutions, Texas A&M and Fort Lewis College are now doing so).
8. Specialized SR&T software support routines are being implemented on the central computer (specialized batch machines, SR&T News files, data search and tape referencing utilities, etc.).

9. Communications systems (including satellite communications) and terminal hardware options are being investigated for the identification of cost effective means to tie other SR&T research sites into the centralized system with much greater throughput capability than is provided through dial-up keyboard terminals.

These steps represent considerable progress toward the establishment of a centralized computer site. Even more impressive is the relatively short time period it has taken to achieve these accomplishments. Considerable work to further support and develop a shared data processing environment for the SR&T community does remain, however.

Remote Job Entry Station.

For research sites other than JSC and Purdue/LARS to make effective use of the shared SR&T data processing network, interactive access is needed to card readers and printers which interface to the shared system. Such capability would be provided through remote job entry stations at each research site. Each SR&T research site should be surveyed to determine its needs for access to data processing capabilities. Cost effective means for providing these capabilities should be identified and a remote data link should be established as soon possible if the potential benefits of the shared system are to be maximized. In order for this recommendation to be effectively implemented, individuals at each of the remote sites must be assigned responsibility for aiding the central site in assessing the remote site's computational needs and in implementing the data link. The various remote SR&T research organizations must come to understand the benefits of a centralized computational facility, plan their access and use of that facility, and aid in the planning and implementation of terminal hardware and the training of their research and development people in the use of the central site.

Interactive Color Device.

One of the key requirements of remote sensing technology is the need for effective man-machine interaction. In addition, this need is greater in a research and techniques development environment than in a high volume, production environment. Color is a means of adding an

additional dimension to what the human researcher or analyst can see, and therefore understand, about an image. One shortcoming of the current research system is the lack of an interactive color display or hardcopy device capability at remote research sites. An investigation needs to be conducted on the color display and hardcopy devices currently available, how such devices could be interfaced with the shared SR&T data processing system, and a means for evaluating the cost and capabilities associated with such devices.

Alternative Processing Technique Comparisons.

One problem which currently faces the people responsible for the transfer of processing techniques from the research community into a production system is that techniques developed at independent sites and tested with differing data sets and differing data processing environments are impossible to compare meaningfully. The shared system offers all SR&T research sites the same processing system and the same data set availability. If the centralized data processing is used for techniques development, little or no reprogramming effort may be required to perform head to head comparative tests of new algorithms and procedures.

Programming Documentation and New Techniques Delivery Standards.

In order to expedite technology transfer within the SR&T research community and between the research and the application communities, certain programming, documentation and software delivery standards should be established. One benefit of the modular LARSYSP1 IPL system which has been developed is that its framework will allow the addition or the replacement of analysis processors without requiring major system rewrites. Therefore, if all SR&T sites would delivery new techniques software in a form compatible with the IPL system, virtually no additional programming would be required to conduct tests comparing new techniques with each other and/or with the base-line system. In addition, making use of programming documentation and data delivery standards will greatly enhance the ability of researchers at the various sites to understand and utilize the software developed at other sites. JSC should allocate people and funds to produce these standards immediately if the first year of the algorithm development under the Multicrop project is to be beneficial.

Local Site Support Responsibilities.

One key component to effective use of any data processing system is the availability of an expert or consultant. These people are vital in the chain of communication between the people who maintain the system and the researchers attempting to make use of the system. In order to maximize the benefits which can be derived from the shared system, it will be convenient to establish a local site expert at each remote site. The site expert should be a technical specialist, familiar with computer programming as well as the technical aspects of typical remote sensing data processing software. At most sites it would probably be convenient for this person to double as a programmer. Site experts will need to spend two or more one-week periods at the central site becoming intimately familiar with the processing system.

Each local site should also identify a person to serve as a computational resources manager. This person will be responsible for interfacing with the central site in order to secure and maintain the computational resources necessary to support users at his local site. Depending upon the amount of activity at the local site, the resources manager and the site expert may be the same person.

Establish Training Course.

A detailed training course for remote SR&T site users should be formulated by Purdue/LARS and JSC/EOD personnel:

1. How to access and use the LARS computer system
2. How to access and use the LARSYSP1 (and/or.P2) baseline software.
3. How to use special SR&T utilities (batch, SR&T News, data search, etc.).
4. Programming standards for research software (e.g., baseline system compatibility, universal format capability, commenting practices, transferability, etc.).
5. Documentation standards.
6. Standard algorithm evaluation and tests, procedures.
7. Procedures for requesting data.
8. Others.

Coordination of ERDS and SR&T Systems Development.

There is some questions as to whether or not the SR&T shared system for research should eventually be integrated into ERDS. There are obvious advantages in terms of transfer of techniques from research to applications use, maintenance of only a single system and data base rather than two, etc. There are also difficulties. Research systems must be highly interactive at all levels, very flexible and easily programmable for the development and exploratory testing of numerous techniques, and highly accessible by the range of researchers utilizing it. Pressures on applications systems are for high efficiency and large throughput capabilities which have traditionally been incompatible with the characteristics of research systems. However, whether or not ERDS and the SR&T data processing system become one, they will be married in the sense that new techniques must be developed and tested in a research mode prior to their incorporation in the applications system. It is recommended that certain JSC personnel working on the design of ERDS also be given the responsibility of contributing aid in designing future hardware, software and procedure additions to the SR&T research system.

Establish a Research and Development Data Processing Network Steering Committee.

To accomplish the above goals, it is suggested that a committee of personnel from SF3, SF12, and Purdue/LARS be established to plan and manage the integration of other SR&T research sites into the user group of the SR&T computer system.

4.2 Usage Projections

It would help prevent bottlenecks and shortages of various system resources if projections of computer usage can be made for two to three month periods. LARS has been able to successfully monitor and plan for systems use by Purdue researchers. The Laboratory does not have sufficient resources or information to project usage by non-Purdue sites. As more sites begin to utilize the computational facilities at LARS, systems problems and resource shortages are likely to occur unless ballpark estimates of usage over a six to eight month time period (the time necessary to react to demands for increased or decreased hardware) from the remote sites can be obtained. We will continue to do our best to keep problems to a minimum, provide the best service possible with the available resources, and be as responsive as possible to the needs of our collective users. Should usage projections indicate the need for additional equipment, we will seek counsel and assistance from our contract monitor at JSC.

APPENDIX A
SYSTEM LOADING PROBLEMS IN A VIRTUAL MACHINE ENVIRONMENT

SYSTEM LOADING PROBLEMS IN A VIRTUAL MACHINE ENVIRONMENT

The IBM 370/148 and the VM/370 control program used at LARS are the hardware and software components respectively of a virtual machine/virtual memory computing system. The purpose of a virtual machine operating system is to allow each user to define a system having all the resources best suited for his use. A characteristic of a virtual memory system using "demand paging" (a user's page of memory is placed in real storage only as needed) is that as memory demands upon the system approach a critical value, the system response time increases in a uniform manner. After the paging demand exceeds that critical value, system response time increases rapidly. As the system attempts to share a fixed amount of real memory among more and more users, the probability of a referenced virtual memory page being in real memory decreases. The user referencing a page not correctly in memory is deactivated until the referenced page can be moved into real memory. If there is not an unused page of real storage available to receive the incoming page, an additional paging operation must be performed to swap out a page of virtual memory. The page that will be swapped out is chosen on a least recently referenced basis. Thus when a user is deactivated, waiting for a page, the probability that he will immediately reference another unavailable page increases. If all of the users of the system are deactivated, waiting for unavailable pages, no processing can be performed and CPU time is irretrievably lost.

High paging rates can arise from several causes and demands by any one user may increase the paging requirements of another user. High paging demands arise from the following general causes:

- A. User code efficiency
 - B. Excessive user demand
 - C. Inefficient system software
 - D. Insufficient hardware resources.
-
- A. User Code Inefficiency

The primary source of inefficiency in user programs is non-optimal use of arrays in FORTRAN Programs. Attempts should be made to keep array size to

the minimum necessary. When using multi-dimensional arrays, care must be taken to insure that the subscript that varies the most is also the subscript that causes the least change in the address of the referenced element.

Several sources of user program paging inefficiency have been identified and the responsible users have been requested to take corrective action. Corrections of the problems can only be performed by the user, LARS can only act in an advisory capacity.

B. Excessive User Demand

This area is closely related to the category of insufficient hardware resources. As the number of concurrent users of the system increases, the demands for CPU time, real memory and channel I/O capacity increase until performance degrades to an unacceptable level. Several stop-gap measures can be taken to deal with excessive user demand. A quick, but in general, undesirable solution is to limit the number of concurrent users. This can be ineffective because a small number of users can saturate the system under some job mixes while a large number of users can peacefully coexist with no problems under other job mix situations. Limiting the number of users has been considered but will only be implemented as a last resort.

A large number of users operating with virtual machines having a small amount of virtual memory will have less impact on the system than a few users with large virtual machines. The policy of System Services is to suggest users limit the size of their virtual memory to 512K bytes. In addition, users with large memory requirements are encouraged to run during non-prime hours.

C. Inefficient System Software

There are two particular examples of system software which effect the system response. The first is the FORTRAN G compiler which is currently used at LARS. This compiler is no longer supported by IBM and does not have an efficient interface to CMS and CP. LARS has recently leased the FORTRAN H (EXTENDED) compiler from IBM. Preliminary tests indicate that user execution time will be significantly decreased and that the program size will be smaller.

This will improve performance by increasing the amount of useful work performed per unit time and decreasing memory requirements. The I/O routines supplied with the new compiler will be more efficient.

The other system software example is the communication software used for remote job entry. The Remote Spooling Communication Subsystem (RSCS) supplied by IBM is subject to periodic re-release and maintenance activities. IBM recently supplied a new release of RSCS. The LARS modifications have been made to this release and it is now the current operational remote job entry system. Prior to the Purdue modification, user satisfaction with RSCS had been quite low due to the slow transmission rates. When the system paging rate is high (above 35 paging operations per second) the reading and printing rate at a remote site will drop to one or two cards or lines per minute. After some study we found that the drop in transmission rate is due to the design of the RSCS system. RSCS consists of a resident code section which is always available and a free storage area. As each remote site is activated, a new copy of the proper line driver is loaded into the free storage area. During program execution, advancing from one line driver to the next causes references to new pages of memory and additional page faults. Under heavy system loading, any processing for a communication line would almost always cause paging activity. To reduce the impact of the RSCS design, LARS has committed a number of pages of real memory to the RSCS system. We currently reserve 20 pages of memory for which RSCS does not have to compete with any other system user. This change has made the RSCS reading and printing rate essentially independent of system load. The memory dedicated to RSCS is unavailable to the other users. This means that there is more competition for the remaining available space.

We have investigated several alternatives to RSCS including a communications software package supplied by the IBM user's group, SHARE. This system is called RASP and is a derivative of RSCS. Although RASP has several nice features, it still retains many of the basic design flaws of RSCS, including separate line drivers for each communication line.

The LARS Systems Group is currently attacking several of the RSCS problems. The line drivers are being separated into a code portion which is re-entrant and usable by many active lines and a variable portion which must be unique for each active line. A better free memory management system is being considered which will allow all of the RSCS data to be stored in fewer memory pages. The prospective net effect of these changes is to reduce the memory requirements of RSCS by about 72,000 bytes. This will allow us to decrease the dedication of real memory to RSCS and still allow the remote sites to operate at acceptable speeds.

D. Insufficient Hardware Resources

After all reasonable system software and user program improvements have been made, the workload may exceed the capacity of the current hardware system. However, to identify the situation, several parameters must be defined or determined by measurement.

1. An "acceptable response" level must be defined.
2. The current workload must be categorized.
3. The system capacity must be defined in terms of the workload and the "acceptable response".

The control program of VM/370 can collect data about the status of hardware and software during system operation. This data can be saved and analyzed by an IBM supplied program called the VM Analysis Package (VMAP). VMAP allows the comparison of various system parameters such as paging rate, disk utilization, and CPU usage, as a function of some selected base variable such as the time of day or number of active users.

VMAP has been leased by the Systems Group and will be used on a continuing basis to monitor the performance of the system and to measure the effectiveness of operational changes.

APPENDIX B
LARSYSP1 SYSTEM MANUAL

Table of Contents

	<u>Page</u>
Section 1. INTRODUCTION	62
Section 2. THE LARSYSP1 SYSTEM ENVIRONMENT.....	63
2.1 LARS Computer Equipment.....	64
2.2 IBM-Supplied Software.....	68
2.3 The LARSYSP1 Virtual Machine.....	73
Section 3. THE LARSYSP1 SYSTEM ORGANIZATION.....	79
3.1 The Overall LARSYSP1 Hierarchy.....	80
3.2 Executive Level.....	82
3.3 Monitor Level.....	83
3.4 Process Level.....	83
Section 4. LARSYSP1 IMPLEMENTATION TECHNIQUES.....	84
4.1 Common Block Usage.....	85
4.2 Generating Functional Load Modules.....	86
4.3 Use of the LARSYSP1 System for Test Runs.....	94
Appendix 1. FUNCTIONAL LOAD MODULES CONTAINING COMMON BLOCKS.....	101
Appendix 2. COMMON BLOCK USAGE.....	103
Appendix 3. LARSYSP1 DATA SET REFERENCE NUMBERS.....	112

LARSYSPl System Manual

SECTION 1

INTRODUCTION

This manual is directed primarily at the programmer or analyst who maintains the LARSYSPl system or writes new programs for it. It is assumed that the reader is familiar with the external characteristics and general operating concepts of the system.

SECTION 2THE LARSYSPl SYSTEM ENVIRONMENT

This section describes the computing environment that forms the basic framework for LARSYSPl. This environment consists of the computer equipment, the IBM-supplied system software, and the use of virtual machines. An understanding of these topics is essential to understanding the internal design and implementation of LARSYSPl. References for further information on all of these topics are included at the end of the section.

The subsections are:

- 2.1 LARS Computer Equipment
- 2.2 IBM-Supplied Software
- 2.3 The LARSYSPl Virtual Machine

LARSYSPI System Manual

2.1 LARS COMPUTER EQUIPMENT

The LARS computer facility uses an IBM System 370 Model 148 computing system. The current configuration includes 1M bytes of main storage, about 700 million bytes of auxiliary direct access storage, 10 tape drives, 1 card reader, 1 printer, one card punch, 20 terminals, six remote reader-printer-punch high-speed terminals, two intelligent remote terminals, 11 dial-up lines and a special digital display and photo copy unit. Figures 2-1 and 2-2 illustrate this configuration. The numbers within parentheses are the current real device addresses.

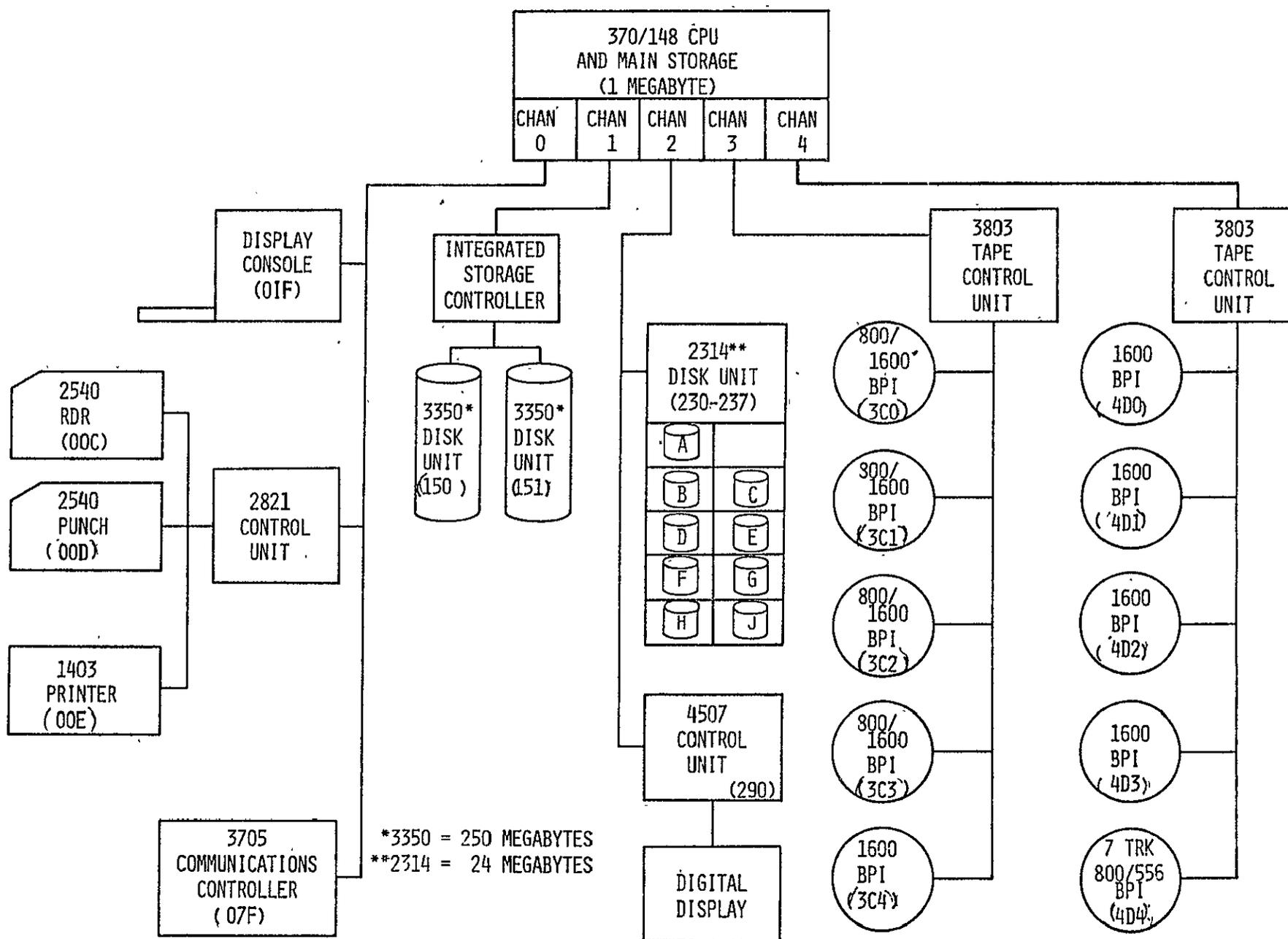


Figure 2-1. LARS Computer Configuration

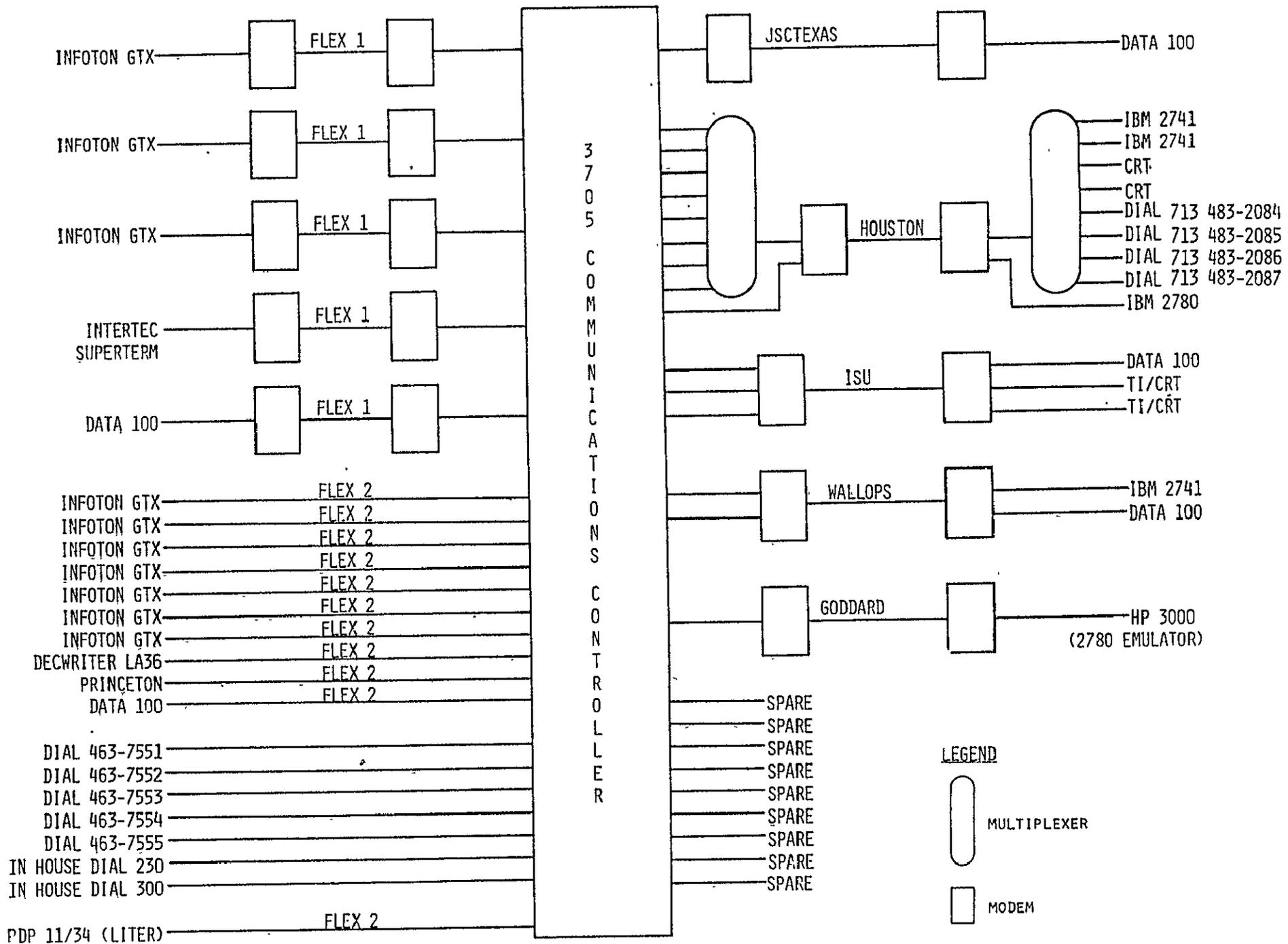


Figure 2-2 IARSS Remote Terminal Configuration

LARSYSPI System Manual

References

The following collection of manuals describes the hardware.

Machine Systems

- GA22-7000 IBM System/370 Principles of Operation
- GA24-3634 IBM 370/148 Functional Characteristics

Teleprocessing Equipment

- GA27-3051 Introduction to the IBM 3705 Communications Controller
- GA27-3005 IBM 2780 Data Transmission Terminal - Component Description

Data 100 User Manual

LARS

- LARS Computer Center User's Guide

LARSYSPl System Manual

2.2 IBM-SUPPLIED SOFTWARE

The LARSYSPl system uses the following IBM-supplied software components as its base: Virtual Machine Facility/370, (VM/370) Conversational Monitoring System (CMS/370), the FORTRAN IV language, and the OS/370 Assembler Language. These components are very briefly described below. The reader should consult the references listed at the end of this subsection for more detailed information.

VM/370

The basic control program under which the LARS computer operates is VM/370. VM/370 is a multiprogramming control program that uses the special hardware features of the Model 148 to create a time-sharing, virtual-machine environment. VM makes it appear to each user that he has individual control of a dedicated System/370 computer complete with I/O devices. These apparent machines are called virtual machines since they are software-created and do not exist in any physical sense. The virtual 370 is indistinguishable to the user and his programs from a real 370. VM/370 allocates the total resources of the real CPU to each virtual machine for a short "slice" of time of predetermined duration.

LARSYSPl System Manual

Since the real machine does not have enough real main storage to support every user's virtual main storage requirement, VM/370 uses a technique called paging. The user's virtual storage is divided into 4096-byte blocks called pages. All pages except those currently in use are kept on secondary storage (3350 disk), and are called into and exchanged out of real storage on a demand basis (demand paging). Pages that are exchanged out of main storage are placed on the disk. VM/370 also handles all virtual machine input/output. All of these operations are transparent to the user and his virtual machine.

VM/370 simulates the card reader, punch, and printer of the virtual machine by a spooling function. If a program running on a virtual machine is to process a card file, the card file must first be read into VM/370, preceded by an ID card to identify the intended virtual machine. VM/370 stores the card file in its spooling area on disk. When the virtual machine requests card input (i.e., reads from its virtual reader), VM/370 supplies it with card images from the stored disk file. The same process works in reverse for printer and punch output: a disk

LARSYSPl System Manual

spooling file is created, which is later transferred by VM/370 from disk to a real printer or punch. This spooling function permits VM/370 to simulate many card readers, punches and printers when, in fact, there are a limited number of these real devices.

CMS/370

Like real machines, virtual machines operate most efficiently under an operating system. LARSYSPl uses CMS/370 as its virtual machine operating system. CMS/370 is a single-user, conversational operating system designed to provide full use of a system/370 through a simple command language entered at a remote terminal. CMS/370 provides a full range of capabilities - creating and managing files, compiling and executing programs, and debugging - from the remote terminal. There are seven basic types of CMS/370 commands:

- File creation, maintenance, and manipulations
- Language processors
- Execution control
- Debugging facilities
- Utilities
- Control commands
- Library facilities

LARSYSPl System Manual

The LARSYSPl programmer will normally use all the facilities of CMS/370 whereas the typical LARSYSPl user need only be familiar with a subset of these.

FORTRAN

CMS/370 provides a FORTRAN IV compiler which is identical to the OS G-level compiler. FORTRAN IV is a mathematically-oriented language useful in writing programs for applications that involve manipulation of numerical data. The majority of the LARSYS program modules are written in the FORTRAN IV language.

Assembler

CMS/370 provides the OS/370 Assembler Language. The Assembler Language provides the full facilities of the hardware to the programmer which FORTRAN does not. A small number of LARSYSPl program modules are written in Assembler Language.

References:

The following collection of documents describes the programming supplied by IBM.

VM/CMS

- GC20-1819: IBM VM/CMS User's Guide
- GC20-1818: IBM VM/CMS Command and Macro Reference

FORTRAN

- GC28-6515: FORTRAN IV Language

Assembler

- GA22-7000: System/370 Principles of Operation
- GC33-4010: OS/VS - OS/VS-VM/370 Assembler Language
- GC33-4021: OS/VS - VM/370 Assembler Programmer's Guide

LARSYSPl System Manual

2.3 THE LARSYSPl VIRTUAL MACHINE

VM/370 implements the virtual machines. Since the virtual machines are simulated, their configurations may differ from each other and from the real machine. Most virtual machines, however, have the same configuration. This configuration is based on the requirements of LARSYSPl and CMS/370 which is the virtual machine operating system required by LARSYSPl. Refer to Figure 2-3 for the configuration diagram. Note that the dashed lines represent optional devices and that the virtual device addresses are in parentheses. When the user completes the LOGIN sequence, his virtual machine is automatically established as that represented by the solid lines. Certain optional devices may be attached to his virtual machine automatically by LARSYSPl.

Configuration

The virtual machine components are:

System/370

VM/370 provides virtual System/370 processors with an installation-specified amount of main storage. Because of the single level of overlay load modules used by LARSYSPl, 768K bytes is large enough to handle the largest functional load module.

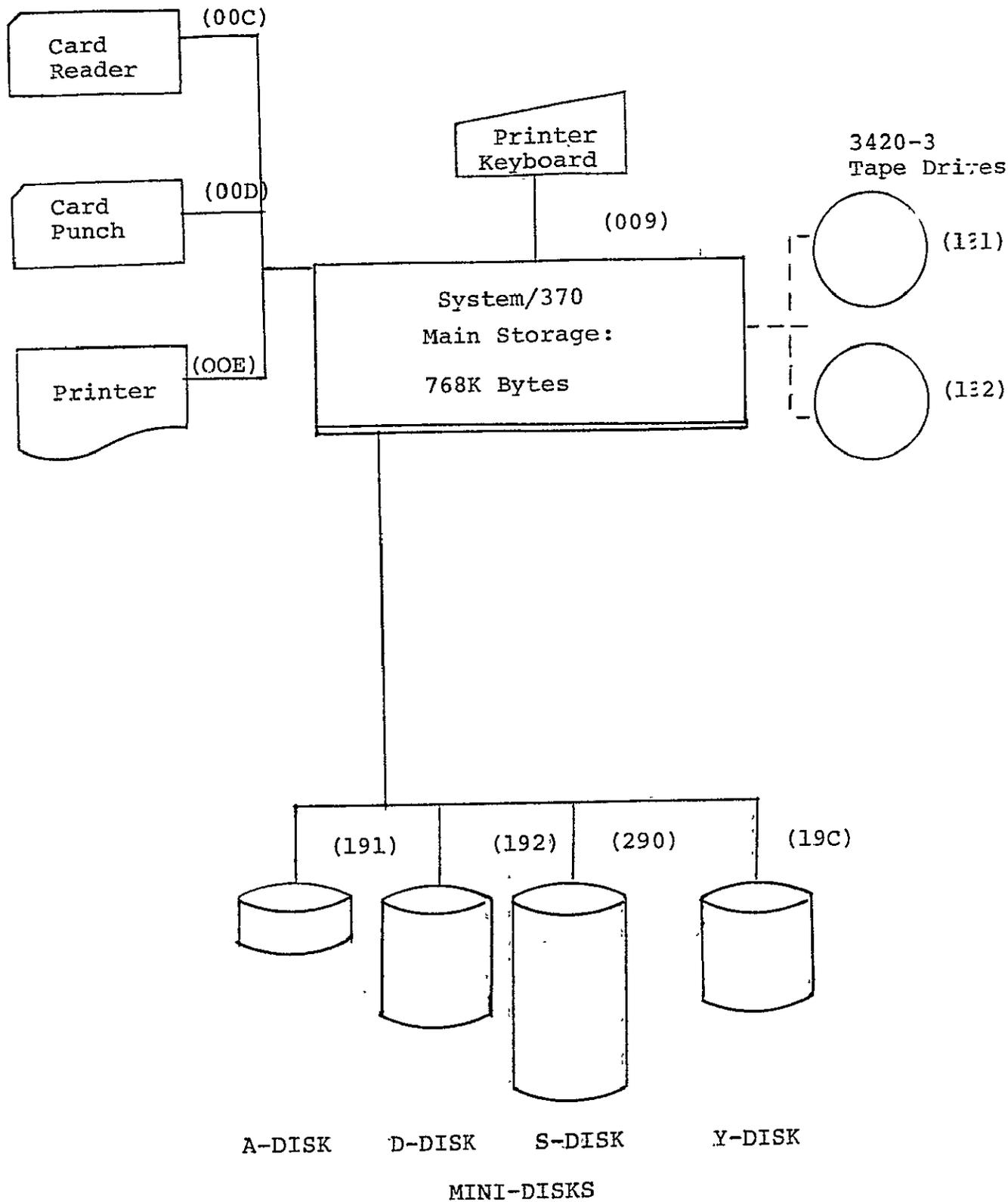


Figure 2-3. LARSYSPl Virtual Machine

LARSYSPl System Manual

Card Reader

This device is automatically specified as a spooled card reader. Therefore, the 2540 card reader in the computer room may be used or, for remote LARSYSPl users, the 2780 or Data-100 may be used. The ID card informs VM/370 as to which userid to spool the input deck.

Card Punch

This device is automatically specified as a spooled punch. Therefore, the 2540 in the computer room will be used unless a REMOTE command has been issued to a specified 2780 or Data-100 punch output. The LARSYSPl module EODLARSYS EXEC will automatically issue the proper REMOTE command for a remote user.

Console Printer-Keyboard

The Terminal where the user LOGIN occurred will be assigned as the console typewriter for the virtual machine (simulating the operator's console of the real machine).

Disk Storage

VM/370 implements the mini-disk concept. This permits many users to have subsets of a real disk assigned to their virtual machine. These mini-disks may be private or shared with other users. The LARSYSPl user automatically has access to four mini-disks. The four disks are:

LARSYSPl System Manual

- A-DISK - This is the user's private or permanent disk and may typically be defined for LARSYSPl as a one-cylinder 2314 (120K bytes) mini-disk. The actual size of the A-disk is defined in the VM-370 user directory for each userid.
- D-DISK - This disk provides space to store intermediate results during a terminal session; thus it is a temporary disk. The D-disk is automatically assigned out of a pool of temporary disk space set aside for that purpose. LARSYSPl currently acquires 25 cylinders (3.0 million bytes) of 2314 disk space.
- S-DISK - This is the single standard system disk shared by all CMS/370 users in the CP directory. The S-disk is currently 39 cylinders of 3350 (17.78 million bytes) and is a single, shared, read-only disk containing all of CMS/370. This includes the CMS/370 nucleus, all transient routines, compilers, FORTRAN library, etc.

Y-DISK - All LARSYSP1 users have access to a single shared, read-only Y disk that contains the LARSYSP1 programs and necessary system support EXEC routines. The Y-disk is defined in the VM/370 directory for each userid and is currently 13 cylinders of 3350 (5.9 million bytes).

Tape Drives

Virtual tape drives must correspond one-for-one with actual tape drives in the computer room. Since there are a limited number of actual tape drives available, their usage must be carefully managed.

The virtual machine may use up to four tapes concurrently; however, none of the present functions require more than two drives, and most of them require only one.

SECTION 3THE LARSYSPl SYSTEM ORGANIZATION

This section describes the overall hierarchy of the LARSYSPl ip1 system. The topics that are covered are:

- 3.1 The Overall LARSYSPl Hierarchy
- 3.2 Executive Level
- 3.3 Monitor Level
- 3.4 Process Level

LARSYSPl System Manual

3.1 THE OVERALL LARSYSPl HIERARCHY

The organization chart in Figure 3-1 shows the hierarchy of control, and where applicable, the individual module names. The system runs under Virtual Machine Facility/370 and the Conversational Monitoring System operating system (VM/370 and CMS/370.) VM/370 is the highest level of software, with CMS/370 running in a virtual machine under VM/370. When the user first logs onto the system, his machine is controlled by VM/370. He then issues I LARSYSPl; at this point his machine has entered CMS/370, and the LARSYSPl Y-Disk has been attached and is ready to be accessed. When RUN EODLARSY is issued by the user, the Executive level is entered. EODLARSY EXEC eventually passes control to the Monitor level by loading the root module and passing control to MONTOR (the FORTRAN MAIN PROGRAM). MONTOR calls MSCAN which reads Function Selector cards and passes control to the appropriate functional supervisor after loading the functional load module. All of the supervisor's names shown in the process level in Figure 3-1 are the names of the supervising subroutines for the individual processors. The names shown in parenthesis are the names of their corresponding load module. The name of a load module must correspond to the first three characters following the '\$' on a function selector control card. The functional modules all load at the end of the root MONTOR and thus overlay each other.

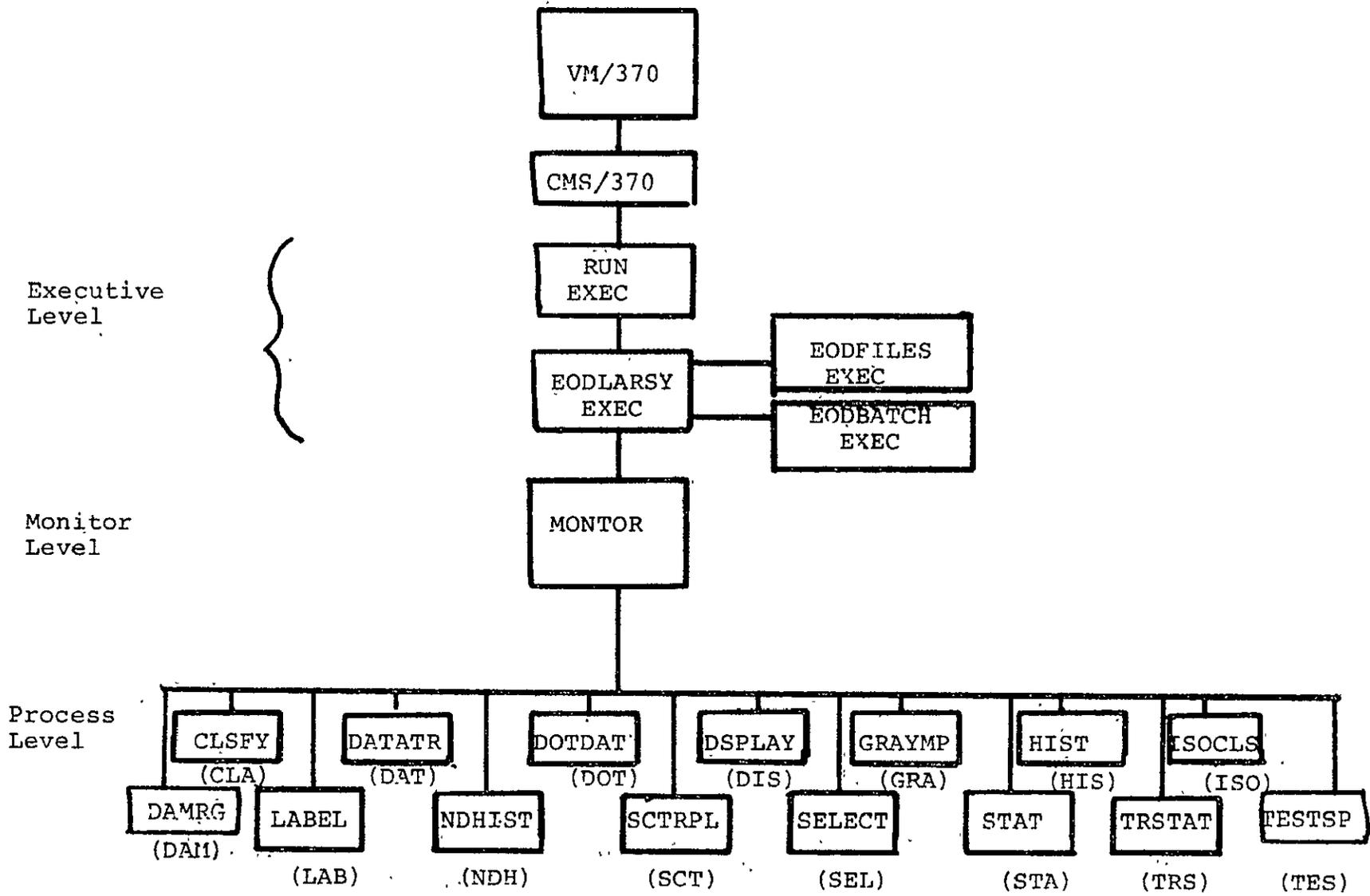


Figure 3-1 The overall organization of LARSYSPI

LARSYSPl System Manual

3.2 Executive Level

The executive level is entered when a user types 'RUN EODLARSY'. RUN EXEC simply passes control to EODLARSY EXEC. It was written so that the user could type 'RUN EODLARSYS' instead of just 'EODLARSYS'. EODLARSY EXEC is a prompting executive routine which asks the user questions and obtains the necessary resources to execute the user's job. That is, it temporarily passes control to EODFILES EXEC to have all FILEDEF's initialized. If the job is to be run interactively, it obtains a scratch D-disk and eventually loads MONTOR and passes control to the monitor level. If the job is to be run by batch, control eventually passes to EODBATCH EXEC which sends all the necessary cards over to the batch machine system.

LARSYSPl System Manual

3.3 Monitor Level

The monitor level is entered from EODLARSY EXEC after loading the MONTOR load module (CMS/370 file MONTOR MODULE). Control enters at MONTOR, the FORTRAN main program of LARSYSPl. The monitor level includes MONTOR, and the subroutines it calls. The Root Load Module which contains the monitor routines also contains a number of system support subroutines not actually used by MONTOR but used by many of the functions in other load modules, as well as common blocks which are used to pass information from one processor to another.

MONTOR functions in a loop calling MSCAN to read Function Selector Cards, requesting major processing functions, and executing these functions. After an '\$EXIT' card has been read by MSCAN, control passes back to EODLARSY.

3.4 Process Level

To enter the process level, the individual processor modules are loaded at the end of MONTOR and control is passed to them.

Section 4LARSYSPl IMPLEMENTATION TECHNIQUES

This section describes a number of implementation techniques that were used in the development of the LARSYSPl ipl system as well as discussions of some of the internal characteristics of the system that are of particular interest to the programmer or system analyst. The topics that are covered are:

- 4.1 COMMON Block Usage
- 4.2 Generating Functional Load Modules
- 4.3 Use of the LARSYSPl System for Test Runs

4.1 COMMON BLOCK USAGE

A BLOCK DATA subroutine must exist for each COMMON block that is used in LARSYSPl. Without the BLOCK DATA subroutine, there would be no CMS TEXT file for the COMMON block and thus no way to explicitly load the block. Even if no variables are initialized in the BLOCK DATA subroutine, it is necessary to explicitly load it in order to force the COMMON to load at the correct location. (see Subsection 4.2 on Generating Functional Load Modules).

4.2 GENERATING FUNCTIONAL LOAD MODULES

This section first defines the functional load modules. It then describes the flow of control and physical arrangement of programs in main storage. Finally, the programming of EXEC routines to generate functional load modules is described in detail. Figure 4-1 showing the main storage arrangement under various conditions is shown on the following page and is referred to throughout the discussion.

The functional load modules are CMS/370 module files. They are created by loading the appropriate TEXT files and using the CMS/370 GENMOD command to create the MODULE file. The name that is assigned to a CMS/370 MODULE file is the first three characters following the '\$' on a function selector control card. The name of the root load module is MONTOR since the FORTRAN program MONTOR is the first program (physically) in the module. The name of a functional MODULE file will always consist of three characters since it is constructed from the first three characters following the '\$' on a function selector control card (e.g. DOT- \$DOTDATA)

In the following text, the word 'load' in lower case refers to the general process of loading from disk to main storage. 'LOAD' in upper case means loading by the CMS/370 LOAD or INCLUDE command, 'LOADMOD' means loading by the CMS/370 LOADMOD command.

1. After LOADING all TEXT files for generating MONTOR
2. After LOADING all TEXT files for generating a functional module
3. After LOAD-MODing the MONTOR module
4. After MONTOR has LOADMODed a functional load module

CMS/370 Nucleus	CMS/370 Nucleus	0	CMS/370 Nucleus	CMS/370 Nucleus
MONTOR other root programs	MONTOR other root programs	A	MONTOR other root programs	MONTOR other root programs
Global common blocks	Global common blocks	B	Global common blocks	Global common blocks
PROCES	e.g. DOT other function program modules local common blocks	C	unavailable (See text)	e.g. DOT other function program modules local common blocks
unused	unused	D	E	unused
CMS/370 EXEC control and loader tables	CMS/370 EXEC control and loader tables	F	FORTRAN DCB's and buffer	FORTRAN DCB's and buffers
		G	CMS/370 EXEC control and loader tables	CMS/370 EXEC control and loader tables

Figure 4-1 Main Storage Allocation

Flow of Control and Physical Arrangement

When the LARSYSPI control command RUN EODLARSY is issued, the EODLARSY EXEC is invoked. EODLARSY uses the CMS/370 LOADMOD command to load the root load module (MONTOR). The contents of main storage at this point are shown in the figure in column 3. When MONTOR (through MSCAN) reads a Function Selector card, it calls BLOAD which uses the LOADMOD command to load the requested functional module. The contents of main storage at this point are shown in the figure in column 4. After the functional module is loaded, MONTOR passes control to the first byte of the module (address C in the figure.)

The figure also shows the contents of main storage at various times. Each of the significant addresses shown on the figure are represented by the following letters:

- O - address zero, start of main storage of the virtual machine.
- A - End of CMS/370 nucleus and start of the storage that is available for user programs. The FORTRAN program MONTOR always starts at this point. The remainder of the programs in the root follow MONTOR. These include LARSYSPI support subroutines such as MSCAN and NXTCHR and FORTRAN library subroutines. These programs end at address B.

- B - Global common blocks are loaded at the end of the other programs in the root. B is the beginning of these common blocks.

- C - The end of the global common blocks. All functional load modules are loaded at this point, with the address 'C' being the beginning address of the load module supervisor. MONTOR always loads a functional load module at this address and branches to it through a CALL to a dummy subroutine names PROCES. This dummy subroutine is used to create a symbolic address for address 'C' in the following way:
 - When the MONTOR load module is loaded during the generation procedure, the dummy subroutine PROCES is loaded at address 'C'. This resolves a CALL to PROCES contained in MONTOR which becomes a branch to address 'C'.
 - The GENMOD command is issued to actually generate the MONTOR load module.

 - When the generated MONTOR load module subsequently loads a functional load module at address 'C' and CALLS PROCES, the call results in a branch to address 'C', which is now the beginning address of a functional load module supervisor.

- D - Highest address used by the functional module.
All functional programs including the supervisor, common, programs unique to this module and support programs lie between C & D.
- E - Start of free storage. Free storage is used by FORTRAN for storing its data control blocks and for I/O buffers for FORTRAN data sets. The address of 'E' must be greater than the ending address of the largest LARSYSPl functional load module. If any LARSYSPl functional load module has an ending address greater than 'E', its loading by MONTOR will overlay data control blocks and I/O buffers established by MONTOR. This will result in strange I/O errors. The address is fixed during the generation of the MONTOR load module, based on the size of the PROCES subroutine. PROCES contains a large array to make it larger than the largest LARSYSPl load module. When MONTOR is GENMODed, CMS/370 stores the address E in the MODULE file. When MONTOR is subsequently loaded, CMS/370 uses the address 'E' from the MODULE file to set the start of free storage.

- F - Start of an area of high memory used by CMS/370. The very top of memory is used by CMS/370 for loader tables. The storage just below the loader tables is used by CMS/370 for EXEC routine control.
- G - End of main storage of the virtual machine.

Programming Module Generation EXEC routines

The first part of this section describes the GMONTOR EXEC routine used to generate the root load module; the second part describes the creation of EXEC routine to generate functional load modules. The GMONTOR and GDOTDAT routines are shown (as examples) in Figures 4-2 and 4-3 at the end of the section.

GMONTOR begins by executing an EXEC routine called LMONTOR which LOADS the programs in the root. It then uses the CMS/370 INCLUDE command to load PROCES. The MONTOR module is then generated by a GENMOD command which specifies that the new MODULE file contains the contents of main storage only from the beginning of the program MONTOR to the end of the program PROCES. The GENMOD command creates the file with a filemode of A1. This must also be done in EXEC routines that create functional load modules. At the time GENMOD is issued, main storage appears as shown in column 1, of Figure 4-1.

LARSYSPl System Manual

After generating the module, GMONTOR types out an address on the typewriter terminal. This is done by using the CP function DISPLAY to display the contents of storage location 574 (hex). 574 is a word called LOCCNT in the CMS NUCON area, which contains the address of the next available free storage. This address may change in subsequent releases of CMS. GMONTOR finishes with a procedure common to all load module generation EXEC routines. The procedure RENAMES the load map to have a filename the same as that of the module.

The EXEC routines, e.g. GDOTDAT, that generate functional load modules begin the same as GMONTOR, i. e., by executing LMONTOR to load the programs in the root. This is necessary to enable the loader to resolve the addresses of calls from the functional programs to subroutines in the root. CMS/370 INCLUDE commands are then issued to load the functional programs.

LARSYSPl System Manual

The following names are UNDEFINED:(etc.)

The above message will appear after each INCLUDE COMMAND: after the final INCLUDE only PROCES should be undefined (except for the GMONTOR module). The message will always state that PROCES is undefined. This is not an error. The GENMOD command is then issued to create a MODULE file containing the contents of main storage from the beginning of the supervisor (address C) to the end of all loaded programs (address D). The address D is then displayed on the typewriter by using the CP DISPLAY function to display address 574 (hex). The programmer should always be certain that this value is less than the address E that is typed out by GMONTOR. The EXEC ends just like GMONTOR by RENAMEing the load map.

4.3 USE OF THE LARSYS P1 SYSTEM FOR TEST RUNS

In program development (modification and debugging of existing functions and writing of new functions), it is necessary to make test runs. The disk hierarchy of CMS/370 makes it very convenient to use most of the standard program modules when testing new modules. The discussion below first lists the steps necessary to make a test run. These are followed by supplemental information which is useful in certain situations.

Steps to make a test run:

1. The standard system programs and load modules will be available on disk. The system programmer is responsible for configuring a programmer's virtual machine to give him read-only access to all standard program modules and load modules or informing him of how to LINK to such disks.
2. All subroutines which are to be modified should be copied to the A-disk. It is strongly recommended that EXEC routines be written to perform this task since the CMS/370 COPYFILE command requires considerable typing. If the programmer is writing a new function rather than modifying an existing one, then this step is not needed, since all new programs will be created on the A-disk.

LARSYS P1 System Manual

3. Use the EDIT command to make the required modifications for the test run.
4. Compile (or assemble) the modified programs.
5. Use the load module generation EXEC routine to create a test version of the load module. The MODULE created will be on the A-disk. It will include the modified programs from the A-disk instead of the standard programs since the A-disk is searched before any other disks to locate the programs. (When writing a new function, time will be saved if a load module generation EXEC routine is written at the beginning rather than later.)
6. Begin LARSYS P1 execution by issuing the RUN LARSYS P1 command to start the program running.

LARSYSPl System Manual

Supplemental Information

1. Intermediate printouts can be directed to the typewriter (unit 16) or printer (unit 6). Also, the programmer is free to use disk files not used by the function he is running. When adding new data set reference numbers, first consult the system programmer to insure their validity. Be sure when writing to symbolic unit numbers (e.g., TYPEWR or PRNTR) that the subroutine has defined these variables.
2. If a COMMON block needs to be changed, it is necessary to copy all program modules containing the COMMON block onto the A-disk (see the section on COMMON block usage.)
3. If a module in the Root Load Module must be modified to make the test run, then it will be necessary to generate a copy of MONTOR on the A-disk. This is done as described in STEPS TO MAKE A TEST RUN above. If a new root load module is created, it is also necessary to create a new version on the A-disk of all functional load modules. This is necessary because the length of the test version of MONTOR is different from the old one, and consequently, the address of the beginning of the functional load module will be different (as well as the addresses in the root of subroutines called by the functional load module).
4. The CMS DEBUG facility provides a powerful debugging tool. (Note that this facility is different from the FORTRAN G facility and requires a knowledge of assembler language.) One feature of LARSYSPl places a small restriction on the use of the facility: LARSYSPl uses an overlay structure. When MONTOR is loaded in EODLARSY EXEC, it is not possible to set breakpoints in a functional load module, since the load module has not been loaded into main storage. Any breakpoint that is set will be overlaid when the subroutine BLOAD actually loads the load module.

The following explanation of DEBUG contains the action required to use DEBUG in LARSYSPl. Following the explanation is a sample terminal session showing the steps. First, two breakpoints must be set in the root. To do this, a copy of EODLARSY EXEC must be created

LARSYSPl System Manual

on the A-disk and modified to contain a call to DEBUG after MONTOR is LOADMODed and before it is STARTed (step 1). A breakpoint (step 2) is set at location LOADED (obtained from the load map).

LARSYSPl System Manual

When location LOADED is reached (step 3), this means that the overlay module has been LOADMODed by subroutine BLOAD and BLOAD is about to return control to MONITOR to start execution of the function. At this point, a breakpoint is set in the functional load module (step 4). Step 5 shows that the breakpoint in the load module was reached. In this example, this breakpoint was the entry point to a subroutine in the overlay module. The programmer then sets the origin to this address and sets a breakpoint in this subroutine. It turned out that this breakpoint was never reached. A caution should be noted here. When breakpoints have been set and not executed CMS/370 should be re-ipl'ed before using DEBUG again.

If this procedure is to be used several times, steps 2-4 can all be set up as stacked lines before the DEBUG command in the EODLARSY EXEC.

5. Some very important cautions should be noted. Whenever the programmer creates a new load module, any TEXT files that are on the A-disk will be used in preference to the standard program TEXT files. This is a useful feature in making test runs. However, it means that once a test version is no longer needed, it should be erased. Also, any MODULE files left on the A-disk will continue to be used rather than the standard system modules. If the system programmer has changed the root load module since the programmer last created a functional module on the A-disk, that functional module will no longer work. For this reason, it is recommended that load modules for test versions not be kept but rather that they be regenerated whenever needed. In light of the above environment, the first step to be taken when unforeseen events take place is to check for unwanted TEXT or MODULE files on the A-disk. It is also important that when a program source is modified, it must be compiled or assembled before it will appear in the newly generated load module.

LARSYSPl System Manual

```
copyfile eodlarsy exec yl = = al
```

```
R; T = 0.09/0.17 13.46.20
```

```
edit eodlarsy exec
```

```
EDIT:
```

```
l /loadmod montor/
```

```
LOADMOD MONTOR
```

```
i debug
```

```
file
```

```
R; T= 0.13/0.26 13.46.54
```

```
run eodlarsy
```

```
DEBUG ENTERED . . .
```

```
break 1 2bf14          location of LOADED          (2)
```

```
return                (from load map)
```

```
EXECUTION BEGINS . . .
```

```
I0013 DOTDATA FUNCTION SELECTED (MONTOR)
```

```
DEBUG ENTERED . . .
```

```
BREAKPOINT 01 AT 02BF14          (3)
```

```
break 2 3f2d8 }          breakpoint set in overlay module (4)
```

```
go
```

```
DEBUG ENTERED . . .
```

```
BREAKPOINT 02 AT 03F2D8          (5)
```

```
origin 3f2d8
```

```
break 3 160          set breakpoint in subroutine
```

```
go
```

```
DO YOU WANT TO RUN ANOTHER JOB?
```

```
(YES OR NO)
```

programmer makes his
own copy of EODLARSY (1
EXEC AND adds line
"debug"

Figure 4-2. GMONTOR EXEC

```
&CONTROL OFF
EXEC LMONTOR
INCLUDE PROCES (NOAUTO
GENMOD MONTOR
&TYPE FREE STORAGE STARTS AT:
CP DISPLAY 574
ERASE MONTOR MAP A1
RENAME LOAD MAP A1 MONTOR MAP A1
&EXIT
```

Figure 4-3. GDOTDAT EXEC

```
&CONTROL OFF
EXEC LMONTOR
INCLUDE DOTDAT DOTVEC (NOAUTO
INCLUDE SET13 DOTS ORDER TAPHDR FLDTYP (NOAUTO
INCLUDE LINERD FLINT WRDOT BUFILL (NOAUTO
INCLUDE SEARCH FLDLAC NUMBR (NOAUTO
GENMOD DOT (FROM DOTDAT)
&TYPE FREE STORAGE STARTS AT:
CP DISPLAY 574
ERASE DOTDAT MAP A1
RENAME LOAD MAP A1 DOTDAT MAP A1
&EXIT
```

LARSYSP1 System Manual

APPENDIX 1.

FUNCTIONAL LOAD MODULES
CONTAINING COMMON BLOCKS

LARSYSPl System Manual

<u>Common Block</u>	<u>Functional Load Module Contained In</u>
BLKCOM	MONTOR
BESTKN	MONTOR
BMTRX	CLA
CLASS	CLA
DISPL	DIS
DOTVEC	DOT
DVNBLK	SEL
FNTDUM	SEL
FSL	SEL
GRCBLK	MONTOR
HISTOR	MONTOR
IDWORD	NDH
INFORM	MONTOR
ISOLNK	MONTOR
LABS	LAB
NDIM	NDH
PASS	ISO
PASSA	ISO
PASSB	MONTOR
SCRACH	CLA
SCTTER	SCT
STBASE	STA
STCBLK	STA
TAPERD	MONTOR
TRBLCK	MONTOR
WRTAP	MONTOR
BLANK	MONTOR
MRGDAT	DAM

LARSYSPl System Manual

APPENDIX 2.
COMMON BLOCK USAGE

LARSYSPl System Manual

Lists of routines which use each of the common blocks, listed by common block, appear on the following pages:

LARSYSPl System Manual

BLKCOM (GLOBAL)

ALLKIN	LINPLT	SETUP8
AMFIL	LNTRAN	SETUP9
AMFILE	MIXMAP	SET10
BLKCOM	MONTOR	SET11
BMFIL	MSCAN	SET13
CHAIN	NDHST1	SET14
CLSFY	NDHST2	STODAT
CLSFY1	OFFSET	STOFIL
CLSFY2	PICT	STOMAP
CLSHIS	PLOT	SUNFAC
CLSMAP	PRINT	TRSFER
CLSSPC	PRTCOV	TRAMPX
COVAR1	PRTFLD	TRHIST
CRDSTA	PRTPCT	TRSTAT
DATATR	PRTSUM	WRTAMT
DOTDST	RANK	WRTBMT
DOTS	RDDATA	WRTFIL
DSPLY1	RDDOTS	WRTFLD
DSPLY2	RDMEAN	TESTSP
DSPTAP	RDMODK	COVPAT
DSTAPE	REDDAT	ISOPAT
EMTHRS	REDIF2	RDDDAT
FILERD	REDIF3	DAMRG
FINT1	REDSAV	SET18
FLDCOV	RESTO	
GENRPT	SAVFIL	
GRYMP	SCATTR	
HEADING	SELECT	
HISTGM	SETADR	
HISTIC	SETUP1	
ISOCLS	SETUP2	
ISODAT	SETUP3	
KBTRAN	SETUP4	
KNEAR	SETUP5	
LABLR	SETUP6	
LEARN	SETUP7	

LARSYSPL System Manual.

BESTKN

* BESTKN
 PRELIM
 REDDAT
 SELECT
 SETUP4

BMTRX

* BMTRX
 CLSFY

CLASS

	CATGRY	REDIF2
* CLASS		SETUP2
	CLSFY	
	CLSFY1	
	CLSFY2	
	CONTEX	

DISPL

* DISPL	FDIST
DISTCV	PRTPCT
DSPLAY	PRTS4M
DSPLY1	REDIF3
DSPLY2	SETUP3
EMTHRS	

LARSYSPl System Manual

GRCBLK

GRAYMP	HISTGM	SETUP5
* GRCBLK	HISTIC	SETUP6
HEADNG	PICT	

HISTOR

GRAYMP	*HISTOR
HISTGM	SETUP6

IDWORD

*IDWORD	RESTO
---------	-------

INFORM

ALLKIN	CRDSTA	FLDFLD	LNTRAN	SAVFIL	SET14
AVEDIV	DATATR	FLDLAC	MAPHDG	SCATTR	TNSFEP
BHTCHR	DAVIDN	FLDMEN	MAPHND	SCTRPL	TRAMTX
BSTCHK	DOTDST	FLDSUB	NDHST1	SELECT	TRHIST
CATGRY	DOTS	FLDTYP	PRELIM	SETADR	TRNDIV
CLRCOD	DSPTAP	GENRPT	PRTCov	SETUP2	TRNSFP
CLSCHK	EVALSP	GRPSCN	PRTFLD	SETUP4	TRSTAC
CLSFY	EVLFET	GTSTAT	RDMODK	SETUP8	USERIN
CLSFY1	EXSRCH	*INFORM	REDDAT	SETUP9	WHRPLC
CLSFY2	FILERD	KBTRAN	REDIF2	SET10	WRTFIL
CLSMAP	FINT1	KNEAR	REDSAV	SET11	
CNDMAP	FLDCLS	LABLR	REORDER	SET13	

LARSYSPl System Manual

ISOLNK

DOTS	*ISOLNK	TAPHDR	PSPPAT
ISOCLS	PSPLIT	TESTSP	DAMRG
ISODAT	SETUP7	ISOPAT	SET18

LABS

ALLKIN	FILERD	MANORD
CLSMAP	KNEAR	MIXMAP
CNDMAP	LABDOT	SET14
DOTDST	LABLR	
DSPTAP	*LABS	

NDIM

ADDRES	FLDFLD	FLDSUB	NDHST2	SET10
FLDCLS	FLDMEN	*NDIM	PICOLR	WRTFIL
		NDHST1		

PASS

CHAIN	RDDATA
CLDIST	RDMEAN
COVAR1	SETUP7
DSTAPE	TESTSP
ISOCLS	COVPAT
ISODAT	ISOPAT
*PASS	PSPPAT
PRINT	RDDPAT
PSPLIT	

LARSYSPI System Manual

PASSA

*PASSA RDMEAN

PASSB

CRDSTA *PASSB RDMODK

SCRACH

CLSFY RELERR
 CLSFY1 *SCRACH
 CLSFY2

SCTTER

CLRCOD	RESCLS	.SETADR	UNPCKV
CNTER	SCATTR	.SET11	
LINPLT	SCTRPL	STOFIL	
OFFSET	*SCTTER	TNSFER	

STBASE

LEARN	.SETUP1	.STAT	*STBASE
-------	---------	-------	---------

STCBLK

LEARN	SETUP1	.STAT	*STCBLK
-------	--------	-------	---------

LARSYSPl System Manual

TAPERD

FLDINT	LINERD	MSCAN	SEARCH	*TAPERD
DAMRG	SET18			TAPHDR

TRBLCK

DATATR	MAXMAT	TRANSF
KBTRAN	PRTCOV	*TRBLCK
LNTRAN	SETUP8	TRHIST

WRTAP

*WRTAP	WRTHED	WRTLN	DAMRG
SET18			

BLANK

*BLANK	COVPAT	MONTOR	PSPPAT	RDDPAT
--------	--------	--------	--------	--------

MRGDAT

DAMRG	*MRGDAT	SET18
-------	---------	-------

LARSYSPl System Manual

APPENDIX 3⁻¹

LARSYSPl Data Set Reference Numbers

<u>DSRN</u>	<u>Description</u>	<u>FILEDEF</u>				
2	Classification Map	FILEDEF	FT02F001	DISK	FILE	FT02F001 D1 (LRECL 320 BLOCK 320 PERM
4	N- dimensional Histogram	FILEDEF	FT04F001	DISK	FILE	FT04F001 D4 (LRECL 3060 BLOCK 3060 PERM
6	Virtual Printer	FILEDEF	6	PRINTER	(RECFM	FA PERM
7	Virtual punch	FILEDEF	7	PUNCH	(LRECL 80	BLOCK 80 PERM
9	Training Statistics	FILEDEF	FT09F001	DISK	FILE	FT09F001 D1 (LRECL 320 BLOCK 320 PERM
11	MSS data	FILEDEF	FT11F001	TAP1	(BLOCK 30600	RECFM U DEN1600 PERM
12	Scatter Plot	FILEDEF	FT12F001	DISK	FILE	FT12F001 D1 (BLOCK 30600 RECFM U PERM
13	Histogram	FILEDEF	FT13F001	DISK	FILE	FT13F001 D4 (LRECL 320 BLOCK 320 PERM
14	MSS Transformed DATA	FILEDEF	FT14F001	DISK	FILE	FT14F001 D1 (BLOCK 30600 RECFM U PERM
15	Terminal	FILEDEF	15	TERMINAL	(PERM	
16	Cluster Map	FILEDEF	FT16F001	DISK	FILE	FT16F001 D1 (BLOCK 30600 RECFM U PERM
19	Dot file	FILEDEF	FT19F001	DISK	FILE	FT19F001 D1 (LRECL 1860 BLOCK 1860 PERM
20	Statistics	FILEDEF	FT20F001	DISK	FILE	FT20F001 D4 (LRECL 320 BLOCK 320 PERM
21	Control Cards	FILEDEF	FT21F001	DISK	FILE	FT21F001 D1 (LRECL 80 BLOCK 800 PERM
22	Random Access File	FILEDEF	FT22F001	DISK	FILE	FT22F001 D1 (LRECL 800 BLOCK 800 XTENT 500 PERM

APPENDIX C
USER TIME LIMIT OPTION

USER TIME LIMIT OPTION

The `TIMELIMIT` command permits the user to set a time limit on virtual CPU time used during a terminal session. It is available only for CMS370.

The `TIMELIMIT` is independent of the type or quantity of jobs run. It remains until (1) it expires, (2) it is cancelled, or (3) the user re-IPL's. It will remain in effect even if a program terminates abnormally (unless re-IPL is required); it will also remain if the user disconnects while a program is still running.

A time limit may be set or altered by a program in the same manner as any other CMS command. A program segment to accomplish this must be written in Assembler.

The time limit is set or changed by typing the command `TIMELIMIT` followed by the desired function, and, if necessary, the amount of time desired. The functions are:

- `INITIAL` Sets a time limit when none existed previously.
 Example: `TIMELIMIT INITIAL 30 MINUTES`
- `RESET` Sets a time limit when there is another time limit
 already established. Example: `TIMELIMIT RESET 15
 MINUTES`. The time limit becomes 15 minutes
 regardless of the time previously set.
- `EXTEND` Adds time to a time limit already established.
 Example: `TIMELIMIT EXTEND 1 HOUR`. If the time
 limit was 15 minutes before the `EXTEND` command,
 it becomes 1 hour and 15 minutes.
- `QUERY` Requests information on time remaining.
 Example: `TIMELIMIT QUERY` results in a display
 of time remaining.
- `STOP` Cancels the existing time limit. Example:
 `TIMELIMIT STOP` cancels the time limit and
 displays time used while the time limit was in
 effect.

If a function named in the command is not recognized, the user will be informed of his error, and also of the correct responses. The system will then wait for a correction. Only the correct function name need be typed. No abbreviations are recognized.

Time requests may be given in hours, minutes, or seconds, and may not exceed 15.5 hours at any time. The time unit may be abbreviated as H, M, or S, or it may be spelled out. If an error is detected in the time request, the user will be asked for a correction. Again, only the corrected information is entered. For example, if a time request of "3 DAYS" were made, the correction might be "3 HOURS". Both the quantity and the unit must be given.

When the time limit expires, the job currently executing is abnormally terminated. The fact that time limit has expired is displayed at the terminal and printed on the line printer, along with the total time used. The user is not logged off, nor is re-IPL required. However, file definitions are lost.

The user is warned not to use the BLIP when this time limit is in effect. This causes the time limit to expire in two virtual CPU seconds.

APPENDIX D
GUIDE FOR SENSIBLE COMPUTER USAGE

GUIDE FOR SENSIBLE COMPUTER USAGE

The IBM 370/148 at LARS is designed to simultaneously service a number of system users. Because users share the resources of the Purdue system, and because the resources are finite, demands placed on the system by each user affect all the other users of the system. It is the goal of the LARS computer facility to maximize the usefulness of our computer to all system users. It is recognized that different remote sensing applications place a wide range of resource demands on the system and that these demanding applications are frequently valuable, therefore a flexible and cooperative philosophy has been adopted towards users of the system. The following guidelines are a matter of "system etiquette" intended to allow the maximum use of the system resources by all users desiring to use the system. Since the total CPU hours for which a user is billed include some system overhead time, following these guidelines will reduce the cost of using the computer as well as increase the response time for, and satisfaction of, other system users.

1. MINIMIZE THE AMOUNT OF CORE DEFINED FOR YOUR MACHINE

The VM370 system on the 370/148 will allow a large number of users to be concurrently running virtual machines with core sizes ranging from 4K up to 16 megabytes. The 370/148 has only a single megabyte of real memory, however. The excess of virtual memory to real memory is stored on disk in blocks of 4096 bytes called pages. The portion of the disk system set aside to maintain the excess virtual memory is called the paging area. As a user's program executes in the real machine, pages of virtual memory are swapped into real memory as they are referenced by the currently executing code. As the amount of concurrently defined virtual memory increases:

- * the paging area must be larger. This makes disk arm movement longer, which in turn leads to increased response times and leads to contention on the disk.
- * the probable distances between those pages which are being concurrently accessed by each user's machine become greater. This again leads to increased disk arm movement, response times and disk contention.

* the number of pages from which data is required for execution of any particular section of code increases. This leads to increased paging in two ways. First more pages must be brought into real memory to execute User 1's programs. Then, when User 2 gets a time slice, it is less likely his referenced pages will have survived in real memory from User 2's previous time slice. Paging hinders system response and performance because the CPU may be inactive if it has to wait to receive referenced pages and because it takes CPU time to initiate, receive, monitor and manage paging operations.

2. RELEASE SHARABLE DEVICES WHEN THEY ARE NOT IN USE

One of the big advantages to a virtual machine system is that, with a reasonable mix of users, nearly all the resources available on the system can concurrently be used at near capacity. For example, the CPU can be working on User A's computations at the same time the system is positioning tapes for User B and User C; waiting on terminal input from User D, User E, and User F; waiting on display commands for User G; positioning disk heads for User H; etc. The system automatically allocates CPU time, paging, spooling areas, etc. It does not allocate and deallocate temp disks, tape drives and the digital display unit. Efficient and courteous use of these devices is dependent on individual users. Whenever a device is not currently in use and not currently holding data which will be used in the near future, it should be detached, either manually by the user, by the executing program (for Fortran this requires a call to system subroutine CPFUNC) or by the controlling EXEC file. This is true even if a similar device will be needed in five or ten minutes. To illustrate this point with a tape drive serving as the sharable device, suppose User A and User B both wish to run similar EOD LARSYS jobs on a hypothetical system with one tape drive. The job flow for both users is to proceed as follows:

Action	Minutes CPU Required	Minutes Clock Time Required
Request & have Operators mount tape	0.	3
Position Tape to Desired File	.001	1
Read Tape to Disk	.999	2
Execute Clustering Algorithm (Reading/Writing Data/Clusters From/To Disk	6.	12
Reposition Tape	.001	1
Read Tape to Disk	.999	2
Execute Classification Algorithm (Reading/Writing Data/Results From/To Disk/Printer	<u>20.</u>	<u>40</u>
	28.	61

If User A beats User B to the drive, it will take just over 2 hours to execute the two jobs and CPU utilization will be approximately 46%. User A finishes in 61 minutes with no waiting on the sharable device. User B waits 61 minutes and then finishes in another 61 under these conditions.

If, on the other hand, the sharable device (tape) is always released when not in use, both jobs could make concurrent use of system resources:

Real Clock Time	Action	Minutes					
		CPU User A	Clock User A	Wait on B	CPU User B	Clock User B	Wait on User A
0	A Request & Mount Tape	0	3				
0+	Request Tape						6
3	A Positions & Reads Tape to Disk	1	3				
6	A's Program Releases Tape	0	0				
6+	A Executes Clustering Algorithm	6	13				
6+	B's Tape Mounted				0	3	
9	B Positions and Reads Tape to Disk				1	3	
12	B releases Tape				0	0	
12	B executes Clustering Algorithm				6	12	
19	A Finishes Clustering	0	0				
19	A Requests & has Tape Mounted	0	3				
22	A Positions & Reads Tape to Disk	1	3				
24	B Finishes Clustering				0	0	
24	B Request Tape						1
25	A Releases Tape	0	0				
25	A Executes Classifications	20	42				
25	B's Tape Mounted				0	3	
28	B's Tape Positioned & Read				1	3	
31	B Releases Tape				0	0	
31	B Executes Classifications				20	42	
67	A Finishes	0	0				
73	B Finishes				0	0	
		28	67		28	66	7

Under this illustration both jobs are completed in 73 rather than 122 minutes. User A's job takes slightly longer (6 minutes) but User B's takes 49 minutes less!! The use of the CPU improved from 46% to 77%! Although the LARS computer has more than 1 tape drive and several temp disks of varying sizes, the effects of hoarding a sharable resource are the same as those illustrated above. The selfish user will decrease his clock time a small percentage while greatly reducing the effectiveness of the system for other users. Programmers should make special efforts to insure that their programs release devices as soon as they are no longer required.

3. USE THE SMALLEST TEMP DISK WHICH WILL SUFFICE

The LARS system has a variety of temp disks available for system users. These disks are intended to provide quick-access temporary storage of moderately large volumes of data. The user can expect data stored on the temp disk to be lost as soon as he releases the disk. The temp disks available on the system are minidisks from the IBM 2314 Disk Pack. 2314 packs may be accessed by CMS360 or CMS370 whereas the other disk systems on the LARS machine may only be accessed by CMS370.

System temp disks are standardly used by LARSYS, EODLARSYS, LARSYSDV, LARSYSXP, EXOSYS, EXOSYSDV, SPSS, etc. In addition, individual users will also use Temp Disks for all their various individual applications. Currently the LARS system has the following Temp Disks available:

Number	Size
5	.24 megabyte disks (2-2314 cylinders)
5	.60 megabyte disks (5-2314 cylinders)
9	1.20 megabyte disks (10-2314 cylinders)
7	3.00 megabyte disks (25-2314 cylinders)

These disks are usually acquired through use of the GETDISK command which initially attempts to secure a temp disk of the requested size. If no temp disks of the user-specified size are available, the temp disks of succeeding larger sizes are searched until an available disk is found or until the list of available disks is exhausted.

Thus, if User A needs five 2314 cylinders of temp disk space, and enters the command:

GETDISK TEMP 5CYL,

he will receive at least 5 cylinders of temporary storage space if any one of 21 temp disks are available. If, on the other hand, User A enters the command:

GETDISK TEMP LARGE or

GETDISK TEMP 25CYL

when he needs only five cylinders of space, two potentially bad things can happen. First, he is less likely to get his disk since there are only 7 large temp disks. Second, if he does get the 25 cylinder disk, he is likely to prevent User B, who really does need a 25 cylinder disk, from being able to use the system because the large temp disks are all assigned to users who could just as well use smaller disks.

4. USE BATCH WHENEVER POSSIBLE

Use of either priority- or basic-rate batch machines will help to level loading on the machine, thereby increasing the proportion of CPU cycles committed to users' programs rather than system overhead. This is because batch jobs are initiated at the operator's, rather than the user's, discretion, and because, past a given point, more efficient use is made of the system when additional demands are placed on system resources sequentially rather than concurrently. Since a lower proportion of total CPU time goes to overhead when priority batch machines are used during periods of heavy system loading, the total CPU time and the cost for the batch job is lower than it would be if the same job were to be run interactively under equivalent system loading conditions. In addition, use of the basic-rate batch machines results in less cost per CPU hour consumed (currently the basic-rate CPU charge is roughly 3/4 of the full priority-rate charge).

There are three negative aspects to the use of batch machines. First, they are less interactive. However, many standard processing jobs require no truly interactive input. (If none of the eventual input parameters depend on the intermediate results, the job is not truly interactive and could be run in batch mode). Second, in order to effectively use the batch machines, a user must be able to create CMS EXEC routines to control his job.

Third, turnaround for batch jobs is somewhat slower than for interactive jobs. For example, it will take at least one day for basic-rate batch machines to return output. Using batch will result in more computation per dollar, however.

5. RESPOND TO REQUESTS FROM THE OPERATIONS AND SYSTEMS STAFF .

Although periods of system saturation will be minimized if users follow the simple guidelines outlined above, periods of exceptionally high resource demands are likely to remain an intermittent problem, especially during the prime shift. Should an analysis of a severe system loading problem indicate that a single user is responsible for a large portion of the problem, the operations supervisor or systems analyst may request that user to curtail his activities until the system load is reduced or to limit his use of the system for this particular job to other than the prime shift. During times of heavy tape or disk usage, users will be requested to release in excess of two tape drives or in excess of one 25 cylinder temp disk. A job which accesses a large portion of core in a manner which causes the machine to spend a large percentage of its CPU cycles waiting for pages severely degrades the system for all users. Such jobs should only be executed when system usage is sparse, e.g., during weekends or the third shift. Programs which cause paging problems should be examined in an attempt to identify areas where efficiency upgrades (with respect to paging) could be made.

APPENDIX E
FORTRAN H COMPILER

FORTRAN H COMPILER

The FORTRAN compiler presently installed on the LARS system is an old version (OS Release 21.6) of the G compiler for which an interface routine has been written to allow operation under CMS370. This old version of the G compiler is used because it is free whereas a newer compiler would involve paying a monthly lease charge.

There are plans to install the current version of the FORTRAN H compiler in the mid-December 1978 through mid-January 1979 time zone. The new H compiler offers several advantages over the old G compiler. The new compiler will be supported by IBM so that compiler errors will be fixed by IBM. New I/O routines and library routines will be supplied with the compiler and these will also be supported by IBM.

The H compiler is designed as a true production compiler. As such, it has the option of optimizing the produced code. When full optimization is used, the CPU time required to run a given job will be decreased. Preliminary timing tests indicate that, on average, CPU time will be reduced by approximately 15 percent. The H compiler supports two new data types, REAL*16 and COMPLEX*32, which allow extended precision when needed. The compiler has an option which will automatically increase variables from single to double precision in a program, thus eliminating the necessity for a programmer to extensively modify program code when higher computational precision is needed.

APPENDIX F

LARS CONSULTING TEAM

LARS CONSULTING TEAM

1. Bill Shelley
 - a. Implement, document, and train JSC personnel in the Data 100 tape transfer procedure.
 - b. Consult with users at LARS about LARSYSPl problems and pass information on to Pat Aucoin.
 - c. Consult with JSC users about programming problems, including converting programs from other computers to the LARS computer.
 - d. Provide available LARS system documentation.

2. Keith Philipp
 - a. Resolve problems with RSCS.
 - b. Resolve modem and line problems with Glen Prow.
 - c. Answer needs for additional teletype terminals.

3. Susan Schwingendorf
 - a. Update and add computer userids.
 - b. Write information for and about JSC in the LARS monthly SCANLINES.
 - c. Consult on BMD and IMSL problems.
 - d. Answer needs for new SR&T remote terminal sites.
 - e. (Same as Shelley's c).

4. Luke Kraemer
 - a. Design historical data base and implement sections of it.
 - b. Consult with JSC users about problems encountered when running in batch mode.
 - c. Answer needs for additional batch machines and enhancements to current ones.
 - d. Be familiar with and coordinate data base storage.
 - e. Responsible for JSC Accounting by User Group.

5. Carol deBranges
 - a. Consult with users on SPSS problems.

6. Ross Garmoe
 - a. Be responsible for disk storage needs.
 - b. Advise JSC on a replacement for their IBM 2780 remote terminal station.

APPENDIX G
SOFTWARE SYSTEMS AVAILABLE ON THE LARS COMPUTER

SUPPORT SERVICES/PRODUCTS AVAILABLE THROUGH LARS

SOFTWARE CONSULTING

PREPROCESSING

REFORMATTING (LANDSAT, FIELD DATA OTHER)
A/D CONVERSION
GEOMETRIC CORRECTION
REGISTRATION
DIGITIZATION

POST PROCESSING

PHOTO PROCESSING (COLOR, COLOR LASER, OTHER)
GRAPHICS/ART
SLIDES
TRANSPARENCIES

COMPUTER SERVICE

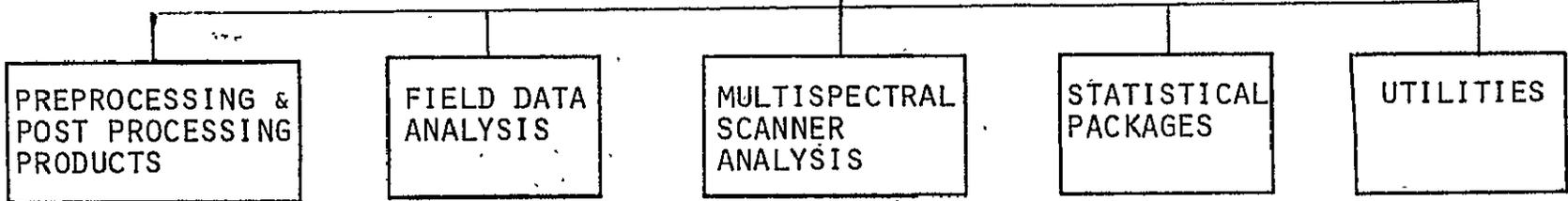
PRIORITY SERVICE
COMPUTER SERVICE
LOCAL TERMINAL SERVICE
DISK STORAGE
TAPE STORAGE

DOCUMENTATION

SYSTEMS MANUALS
USERS GUIDES
PROGRAM ABSTRACTS
EDUCATIONAL PACKAGE

SOFTWARE AVAILABLE ON THE LARS COMPUTER

CP/VM370
CMS360
CMS370



REFORMATTING
GEOMETRIC
CORRECTIONS
REGISTRATIONS
A/D CONVERSIONS
PHOTO PROCESS-
ING
DATA DIGITIZA-
TION

EXOSYS
EXOSYS DV

EOD LARSYSP1
LARSYS
LARSYSDV
ECHO
LAYER
CLASSY
AMOEB
UNIFORM
CHROMATICITY
TRANSFORM
LIST

PLANNED:
EOD LARSYSP2

SPSS
BMD
IMSL

PLANNED:
SAS

CMS370 BATCH
CMS360 BATCH
EXECUTIVE CONTROL
EDITOR
FORTRAN G
ASSEMBLER
DEBUG PACKAGE
TAPE, DISK, CORE
DUMPS
TAPE TRANSFERS
NEWS FILES
GCS
DATA BASE ACCESS
ACCOUNTING

PLANNED:
FORTRAN H
CMSP

APPENDIX H
LARS COMPUTER SYSTEM CONFIGURATION

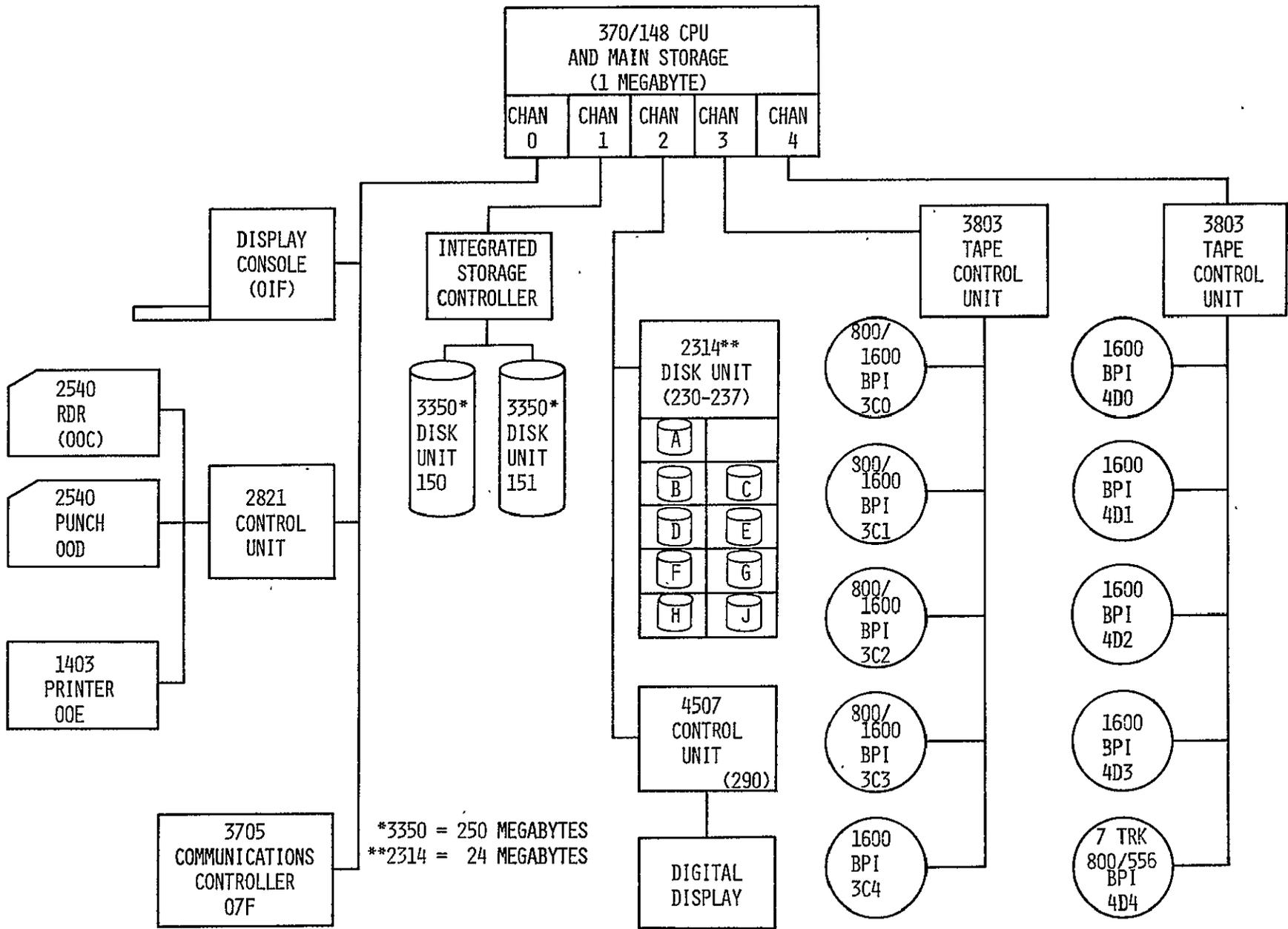


Figure H-1. Purdue/LARS Computer Configuration

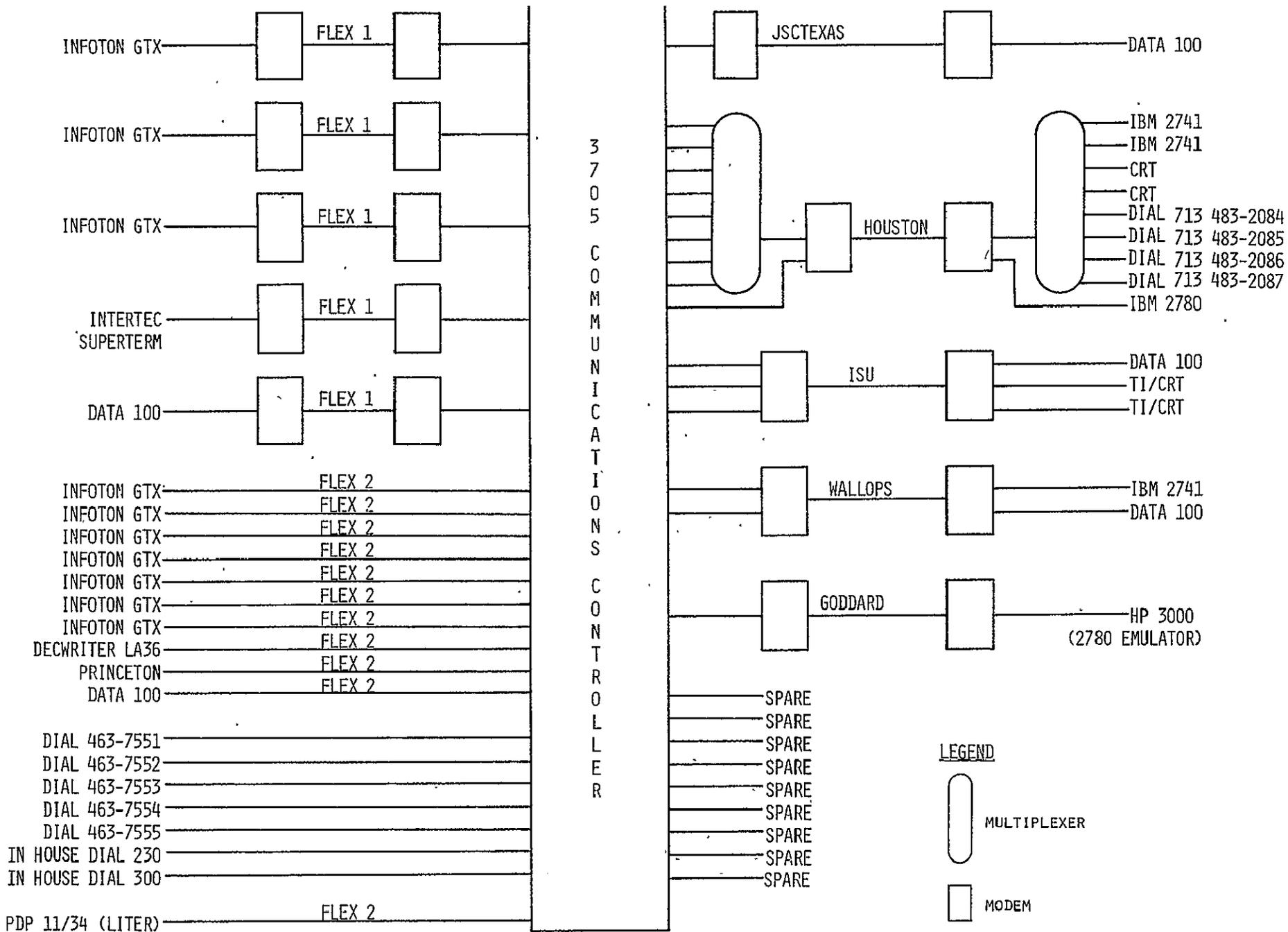


Figure H-2 Purdue/LARS Terminal Configuration

APPENDIX I

AGENDA FOR DEVELOPING A MODULAR LARSYSP1 IPL SYSTEM

AGENDA FOR DEVELOPING A MODULAR LARSYSPI IPL SYSTEM

ITEM	TIME	LARS INTERFACE	AUCOIN	HAVENS	DAVENPOR
Receive & discuss documentation on LARSYSPI IPL System	Mon.AM &	Shelley/Etheridge	X	X	
IPL System maintenance	Mon.PM	Shelley/Etheridge	X	X	
Alter IPL system to modular form	Tue.AM	Shelley	X		X
Alter documentation	Tue.AM	Shelley	X		X
Discuss Data base requirements	Tue.AM	Kraemer		X	
Discussion & documentation to developmental processors/systems	Tue.PM	Shelley/Etheridge	X	X	X
Developmental systems module EXEC	Tue.PM	Shelley	X		X
Implement HEADERPRINT TESTSP ASDM	Tue.PM	Shelley	X		X
See demo of SRTNEWS facility	Tue.PM	Schwingendorf		X	
Present & discuss Prompting EXEC & documentation	Wed.AM	Shelley/Etheridge	X	X	X
Discuss certain systems routines (TSPACE, MOUNT, CPFUNC)	Wed.AM	Shelley	X	X	X
Make 10-cyl update to EXEC & references	Wed.AM	Shelley	X		
Discuss LARS P1 problems & processors needed	Wed.PM	Fuhs	X	X	X
Procedure for reporting suspected system problems	Thur.AM	Shelley/Philipp	X	X	
Efficient VM programming practices	Thur.AM	Philipp	X	X	
LARSYSPI efficiency changes needed	Thur.AM	Shelley/Etheridge/ Fuhs	X	X	
Put SEGFO, MOUNT, CPFUNC in Data Merge	Thur.PM	Kraemer/Shelley	X		
Review & receive LARS Priorities on P1 work	Thur.PM	Kast/Shelley/Fuhs		X	
Review Trip	Thur.PM	Kast/Phillips	X	X	

APPENDIX J
SEGMENT CATALOG

SEGMENT CATALOG

There are at least 6 separate data bases associated with the Multicrop Experiment:

- * Data base of acquisitions of Landsat data,
- * Data base of wall-to-wall ground observations for each segment,
- * Data base of labels and label information for each segment,
- * Data base of agronomic observations data,
- * Data base of crop calendar data, and
- * Data base of meteorological data.

All of these data bases should be organized on a segment-by-segment basis and at least 3 of these data bases (the Landsat data, the ground observation data, and the dot label data) are primarily digital in nature and should be stored in a computer-compatible, computer-indexed form.

Computer indices pointing to the location of various elements of the various data bases associated with each segment, together with the value of certain specific variables for each segment, will compose a Segment Catalog. This document is the current design of that Segment Catalog.

The Segment Catalog (See Fig. J-1) will itself be composed of six separate data files:

- 1) The Segment Index,
- 2) The Acquisition List,
- 3) The Dot Label Table,
- 4) The Ground Observations Index,
- 5) The Ground Observation Fields, and
- 6) Repeated Measurement Records

The last three files compose the Ground Observations Table. All Segment Catalog files will be stored on disk and will be readily accessible through user oriented routines. Routines will be designed to perform two separate functions, 1) producing data base status reports for users and 2) providing the data analysis system (e.g., LARSYSPl) routines direct access to the various data components which the processing systems require.

The Segment Index data file is a master index for the Segment Catalog. It is composed of two types of information:

- 1) Information which is unique to each segment (e.g., the segment latitude, the segment longitude, the county in which the segment is located, the state in which the segment is located, etc.), and
- 2) Pointers to the other data files within the Segment Catalog.

The Segment Index data file will include two pointers to other Segment Catalog data files. For example, the first pointer for the Acquisition List data file will point to the first acquisition over the segment of interest in the Acquisition List and the second pointer for the Acquisition List file will point to the last acquisition which has been entered into the data base. The Acquisition List, the Dot Label Table and the Ground Observation Index will each be double linked lists. Fig. J-2 presents an example of how the records in the Segment Index File might appear. If there are no entries in a particular data file for a segment, both pointers to that data file in the Segment Index File will be set to 0. For example, if no satellite data has been collected for a given segment, no entries for that segment will be present in the Acquisition List and the pointers in the Segment Index for the first acquisition and the last acquisition in the Acquisition List will both be set to 0.

Fig. J-3 presents the candidate set of elements for the Acquisition List File. Of particular interest are the first two elements of each record. The first element, called the previous acquisition element, points backwards through the Acquisition List. This element represents the acquisition list record number which contains information about the acquisition prior to the current acquisition for the segment of interest. When there is no previous acquisition in the data base, this pointer will be set to $-N$, where N is the pointer to the record in the Segment Index for the segment of interest. The second element in the Acquisition List points to where information about the next chronological acquisition taken is located. If the current Acquisition List record contains information about the most recent acquisition in the data base, then the next acquisition value will be set to $-N$, where N is again the record in the Segment Index data file which contains information about the segment.

Figures J-4 and J-7-J-9 represent candidate configurations of the Dot Label Table data file and the Ground Observations Table data file of the Segment Catalog. Like the Acquisition List, these data files have as their first two entries, pointers to the previous and the next entries.

Fig. J-5 and J-6 are simple label files which can be accessed by either the Dot Label Table or the Observations Field Records.

SEGMENT CATALOG

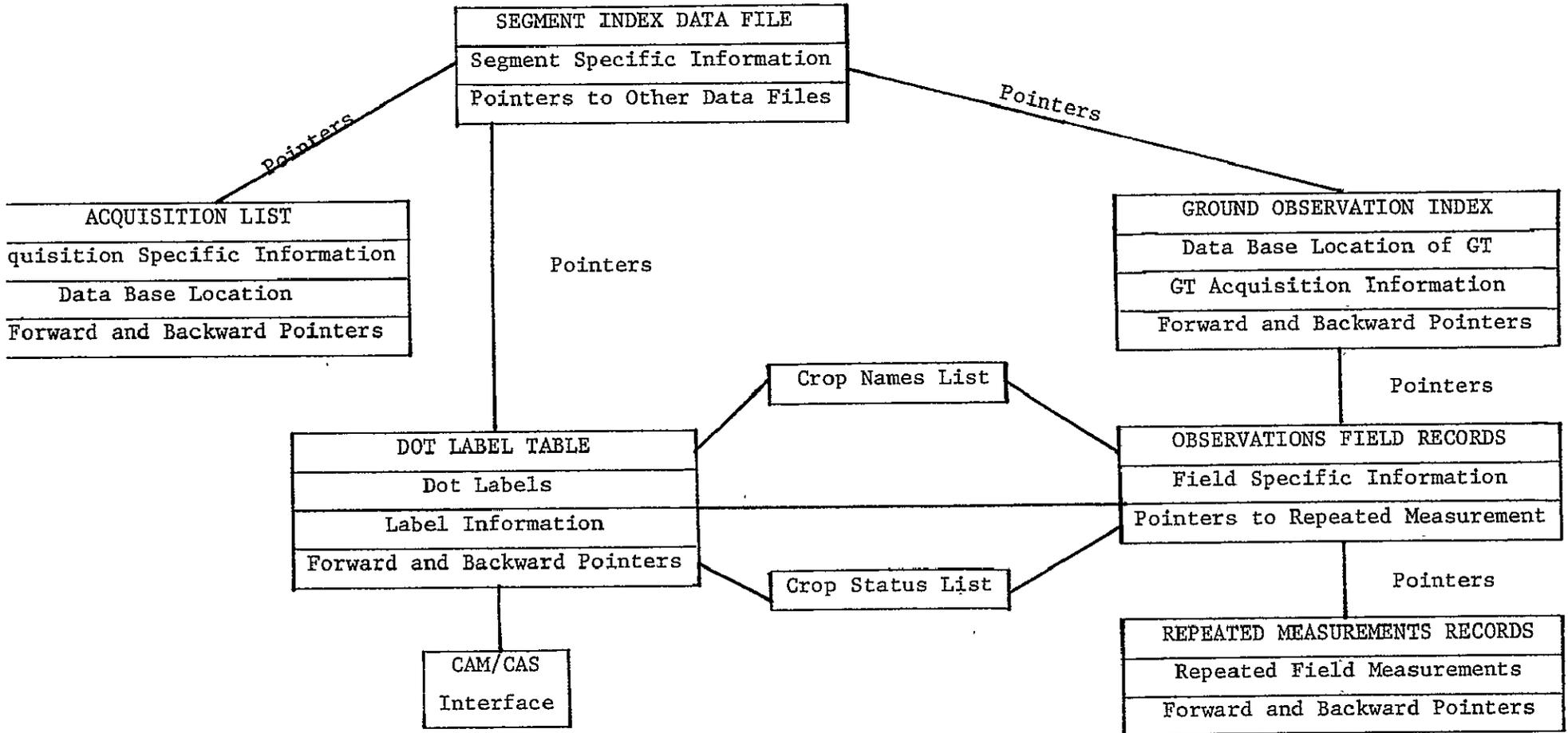


Figure J-1

Format of Acquisition List Records

<u>Entry</u>	<u>Format</u>	<u>Bytes</u>
Sensor System	A8	1-8
Previous Acquisition	I*4	9-12
Next Acquisition	I*4	13-16
Orbit Number	I*4	17-20
Scene Frame ID (SFI)	I*4	21-24
Reference SFI	I*4	25-28
Refernce SFI for Ground Observations	I*4	29-32
Date Data Collected (YYDDD)	I*4	33-36
Time Data Collected (GMT)	I*4	37-40
Date Entered in Acquisition List (YYDDD)	I*4	41-44
Goddard Processing Date	I*4	45-48
Date of Unload Tape	I*4	49-52
Peak Sharpness	R*4	53-56
Normalized Peak to Background Ratio	R*4	57-60
Segment Number	I*2	61-62
Sun Elevation (min.)	I*2	63-64
Sun Azimuth (min.)	I*2	65-66
Tape Number	I*2	67-68
Lines of Data	I*2	69-70
Columns of Data	I*2	71-72
Bias Factor for Channel 1	I*2	73-74
Bias Factor for Channel 2	I*2	75-76
Bias Factor for Channel 3	I*2	77-78
Bias Factor for Channel 4	I*2	79-80
Gain Factor for Channel 1	I*2	81-82
Gain Factor for Channel 2	I*2	83-84
Gain Factor for Channel 3	I*2	85-86
Gain Factor for Channel 4	I*2	87-88
Cloud Cover	L*1	89
Processing Flag	L*1	90
Greenness of Soils Line	L*1	91

Figure J-3

<u>Entry</u>	<u>Format</u>	<u>Bytes</u>
Haze Number	L*1	92
File Number	L*1	93
First Channel	L*1	94
Last Channel (NC)	L*1	95

Format of Dot Label Table Records

<u>Entry</u>	<u>Format</u>	<u>Bytes</u>
Previous Label Entry	I*4	1-4
Next Label Entry	I*4	5-8
Segment Number	I*2	9-10
Analyst Identifier	I*1	11
Number of Categories	L*1	12
Labelling Convention	I*2	13-14
Experiment	I*2	15-16
Date of Labelling (YYDDD)	I*4	17-20
Acquisitions Used in Labelling		
Date #1 (YYDDD)	I*4	21-24
Date #2 (YYDDD)	I*4	25-28
⋮		
Date #8 (YYDDD)	I*4	39-52
Category Names		
Crop Annotated as Category 1	I*2	53-54
Crop Annotated as Category 2	I*2	55-56
Blank fill or Crop Annotated Category 3	I*2	57-58
⋮		
Blank fill or Crop Annotated Category 30	I*2	111-112
Pointer to First Test Field	I*4	113-116
Pointer to Last Test Field	I*4	117-120
Pointer to First DO/DU Field	I*4	121-124
Pointer to Last DO/DU Field	I*4	125-128
CAM/CAS Tape Number	I*2	129-130
CAM/CAS File Number	I*2	131-132
Number of Labels (NC)	I*2	133-134
Labels and Annotation		
Labelled Line for Dot 1	I*2	135-136
Labelled Column for Dot 1	I*2	137-138
*Dot Label for Dot 1	I*2	139-140
**Dot Annotation for Dot 1	I*1	141

Figure J-4

Format of Dot Label Table Records

<u>Entry</u>	<u>Format</u>	<u>Bytes</u>
Dot Status for Dot 1	L*1	142
⋮		
Labelled Line for Dot NC	I*2	$127+8*NC-128+8*NC$
Labelled Column for Dot NC	I*2	$129+8*NC-130+8*NC$
*Dot Label for Dot NC	I*2	$131+8*NC-132+8*NC$
**Dot Annotation for Dot NC	L*1	$133+8*NC$
Dot Status for Dot NC	L*1	$134+8*NC$

* 1 - N - 30 == Type one dot corresponding to category name N
 ** 129 - N - 158 == Type two dot corresponding to category name N-128

** 0 == A Field Pixel
 1 == Dot in DO Area
 2 == Dot in DU Area
 3 == Dot is an edge pixel
 4 == Dot is a boundary pixel

Figure J-4 (continued)

Crop Name List

<u>Entry</u>	<u>Format</u>	<u>Byte</u>
Name for Label 1	A8	1-8
Variety for Label 1	A8	9-16
Name for Label 2	A8	17-24
Variety for Label 2	A8	25-32
		⋮

Figure J-5

Crop Status List

<u>Entry</u>	<u>Format</u>	<u>Byte</u>
Name of Status 1	A8	1-8
Name of Status 2	A8	9-16
⋮		⋮

Figure J-6

Ground Observations Table
Ground Observations Index

<u>Entry</u>	<u>Format</u>	<u>Bytes</u>
Previous Ground Truth Entry	I*4	1-4
Next Ground Truth Entry	I*4	5-8
Segment Number	I*2	9-10
Number of Fields Monitored for Agronomic Data	I*2	11-12
Date of Initial GT Record (YYDDD)	I*4	13-16
Date of GT Reference (YYDDD)	I*4	17-20
Pointer to Acquisition List for first W to W GT	I*4	21-24
Pointer to Acquisition List for last W to W GT	I*4	25-28
Pointer to first Monitored Field	I*4	29-32
Pointer to last Monitored Field	I*4	33-36

Figure J-7

Observation Field Records

<u>Entry</u>	<u>Format</u>	<u>Bytes</u>
Previous Field Monitored	I*4	1=4
Next Field Monitored	I*4	5=8
Segment Number	I*2	9=10
Field Number	I*2	11=12
Field Identifier	A8	13=20
Crop Names Entry	I*2	21=22
Crop Status Entry	I*2	23=24
Date Planted (YYDDD)	I*2	25=26
Nitrogen Fertilization	I*2	27=28
Row Width (meters)	R*4	29=32
Pointer to first of Repeated Measures Data	I*4	33=36
Pointer to last of Repeated Measures Data	I*4	37=40
Number of ARCS (NARC)	I*2	41=42
Line Coordinate 1	I*2	43=44
Column Coordinate 1	I*2	45=46
Line Coordinate 2	I*2	47=48
Column Coordinate 2	I*2	49=50
Line Coordinate NARC	I*2	39+NARC*4-40+NARC*4
Column Coordinate NARC	I*2	41+NARC*4-42+NARC*4

Figure J-8

Repeated Measurement Record

<u>Entry</u>	<u>Format</u>	<u>Bytes</u>
Previous Measurement	I*4	1-4
Next Measurement	I*4	5-8
Segment Number	I*2	9-10
Field Number	I*2	11-12
Date Measured (YYDDD)	I*4	13-16
Maturity	L*1	17
% of Ground Cover	I*1	18
% of Green Leaves	L*1	19
Condition	L*1	20

Figure J-9

157

Final Report Distribution List

NAS9-14970

<u>NAME</u>	<u>NUMBER OF COPIES</u>
NASA/Johnson Space Center. Houston, Texas 77058	
ATTN: J. D. Erickson/SF3	(1)
ATTN: M. C. Trichel/SF3	(1)
ATTN: L. F. Childs/SF	(1)
ATTN: K. J. Demel/SF5	(1)
ATTN: F. Weber/SF5	(1)
ATTN: G. O. Boatwright/SF3	(1)
ATTN: K. Baker/SF4	(1)
ATTN: H. G. DeVezin, Jr./FM8	(1)
ATTN: R. P. Heydorn/SF3	(1)
ATTN: M. C. McEwen/SF3	(1)
ATTN: D. H. Hay/SF12	(1)
ATTN: D. L. Amsbury/SF5	(1)
ATTN: J. G. Garcia/SF3	(1)
ATTN: F. G. Hall/SF2	(1)
ATTN: B. L. Carroll/CO9	(1)
ATTN: E. Laity/SF121	(2)
ATTN: R. Shirkey/JM6	(4)
ATTN: J. T. Wheeler/AT3	(1)
ATTN: G. E. Graybeal/SF4	(2)
ATTN: I. D. Browne/SF3	(5)
IBM Corporation FSD Mail Code 56 1322 Space Park Drive Houston, Texas 77058	
ATTN: Mr. Stanley Wheeler	(1)
Department of Mathematics Texas A&M University College Station, Texas 77843	
ATTN: L. F. Guseman, Jr.	(1)
ERIM P. O. Box 8618 Ann Arbor, Michigan 48107	
ATTN: R. F. Nalepka	(1)
ATTN: W. A. Malila	(1)
ATTN: R. C. Cicone	(1)
Kansas State University Department of Statistics, Calvin 19 Statistical Lab Manhattan, Kansas 66506	
ATTN: A. M. Feyerherm	(1)

NAME

NUMBER OF COPIES

U. S. Department of Agriculture
Soil & Water Conservation Research Division
P. O. Box 267
Weslaco, Texas 78596

ATTN: Dr. Craig Wiegand

(1)

U. S. Department of Interior
USGA National Center
Mail Stop 115
Geography Program
Reston, Virginia 22092

ATTN: Dr. James R. Anderson

(1)

Director, Remote Sensing Institute
South Dakota State University
Agriculture Engineering Building
Brookings, South Dakota 57006

ATTN: Mr. Victor I. Myers

(1)

U. S. Department of Agriculture
Forest Service
240 W. Prospect Street
Fort Collins, Colorado 80521

ATTN: Dr. Richard Driscoll

(1)

University of California
School of Forestry
Berkeley, California 94720

ATTN: Dr. Robert Colwell

(1)

Environmental Remote Sensing
Applications Laboratory
Oregon State University
Corvallis, Oregon 97331

ATTN: Dr. Barry J. Schrupf

(1)

U. S. Department of Interior
Director, EROS Program
Washington, D. C. 20242

ATTN: Mr. J. M. Denoyer

(1)

<u>NAME</u>	<u>NUMBER OF COPIES</u>
John F. Kennedy Space Center National Aeronautics & Space Administration Kennedy Space Center, Florida 32899	
ATTN: Mr. J. P. Claybourne/AA-STA	(1)
Texas A&M University Institute of Statistics College Station, Texas 77843	
ATTN: Dr. H. O. Hartley	(1)
Code 163-427 Jet Propulsion Laboratory 4800 Oak Grove Drive Pasadena, California 91103	
ATTN: Mr. Fred Billingsley	(1)
NASA Headquarters Washington, D. C. 20546	
ATTN: Mr. Pitt Thome/ER-2	(1)
ATTN: Mr. Leonard Jaffee/D	(1)
ATTN: Ms. Ruth Whitman/ERR	(1)
Texas A&M University Remote Sensing Center College Station, Texas 77843	
ATTN: Mr. J. C. Harlan	(1)
USGS National Center Mail Stop 115 Geography Program Reston, Virginia 22092	
ATTN: James Wray	(1)
Canada Centre For Remote Sensing 2464 Sheffield Road Ottawa, Canada K1A 0Y7	
ATTN: Dr. David Goodenough	(1)
Dr. Paul Mausel ISU Terre Haute, IN	(1)

NAMENUMBER OF COPIES

Remote Sensing Laboratory
129 Mulford Hall
University of California
Berkeley, California 94720

ATTN: C. M. Hay (1)

NASA Lyndon B. Johnson Space Center
Public Affairs Office, Code AP
Houston, Texas 77058 (1)

National Aeronautics and Space Administration
Scientific and Technical Information Facility
Code KS
Washington, D. C. 20546 (1)

Department of Watershed Sciences
Colorado State University
Fort Collins, Colorado 80521

ATTN: Dr. James A. Smith (1)

NASA/Johnson Space Center
Earth Resources Program Office
Office of the Program Manager
Houston, Texas 77058 (1)

NASA/Johnson Space Center
Earth Resources Program Office
Program Analysis & Planning Office
Houston, Texas 77058

ATTN: Dr. O. Glen Smith/HD (1)

NASA/Johnson Space Center
Earth Resources Program Office
Systems Analysis and Integration Office
Houston, Texas 77058

ATTN: Mr. Richard A. Moke/HC (1)

ATTN: Mr. M. Hay Harnage, Jr./HC (1)

Earth Resources Laboratory, GS
Mississippi Test Facility
Bay St. Louis, Mississippi 39520 (1)

ATTN: Mr. D. W. Mooneyhan (1)

Lewis Research Center
National Aeronautics & Space Administration
21000 Brookpark Road
Cleveland, Ohio 44135

ATTN: Dr. Herman Mark (1)

