RECEIVED BY DTIC OCT 7 1971

QUARTERLY TECHNICAL PROGRESS REPORT

April, May, June 1971

MASTER

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN • URBANA, ILLINOIS

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

QUARTERLY TECHNICAL PROGRESS REPORT

April, May, June 1971

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

peg

TABLE OF CONTENTS

	Page
1. CIRCUIT RESEARCH	1
1.1 Bundle Processing	1
1.1.1 Bundle Signal Repeater and Restorer.	1
1.1.2 SABUMA	1
1.2 APE	1
1.2.1 The 42.5 MHz Clock Receiver	1
1.3 PENTECOST	3
1.3.1 Testing the Land System for Sequential Presentation	3
2. HARDWARE SYSTEMS RESEARCH	4
2.1 LASCOT	4
2.1.1 Laser Specifications	4
2.2 OLFT	4
2.2.1 Intermediate System Tests	4
2.3 ORBIT	5
2.3.1 Receiver Synchronization	5
2.4 Tricolor Cartograph	5
2.4.1 Physical Wiring	5
2.4.2 Power Supply	5
2.5 BLAST	7
2.5.1 Screen Signal	7
2.5.2 Scan Correction	7
2.6 Eidolyzer	7
2.7 Sematrix	8
2.8 LINDA	8
2.8.1 Project Status	8
2.8.2 Future Work	10
2.9 Stereomatrix	13
2.9.1 Transformer	13
2.9.2 Display	13
2.9.3 Observer Position Detector	14
2.10 PAGAN	14
2.10.1 Completion of the Project	14
2.10.2 Suggestions for Improvements	15
2.11 Scantrix	17
2.11.1 Possible Implementations	17
3. SOFTWARE SYSTEMS RESEARCH	22
3.1 Numerical Processes	23
3.1.1 Sparse Matrix Inversion	23
3.1.2 The Steady State Package	29
3.1.3 Steady State Package Testing	32
3.2 Non-Numerical Packages	35
3.2.1 Item Analysis	35
3.2.2 Global Analysis I and II	39

3.3	Graphical Remote Access Support System . . .	43
3.3.1	Disk Monitor System	43
3.3.2	Display Terminals	43
3.3.3	Information Retrieval Package . . .	43
3.3.4	Disk Interface	44
3.3.5	Monitors	44
3.3.6	Remote Data Structure Utilities . .	46
3.4	Computer Maintenance and Construction . . .	53
3.4.1	CompuTek Computer Graphics Terminal	53
3.4.2	Equipment Maintenance Log Summary .	53
3.4.3	Air Conditioning.	54
3.4.4	Stereomatrix Interface	55
4.	IMAGE PROCESSING AND PATTERN RECOGNITION RESEARCH:	
	ILLIAC III.	59
4.1	Interactive Picture Processing	60
4.1.1	Show-and-Tell	60
4.1.2	Intermachine Link	62
4.1.3	Scan-Display Devices	63
4.2	Parallel Processing of Pictures	64
4.2.1	Pattern Articulation Unit	64
4.2.2	Signal Detection Theory	64
	4.2.2.1 Extensions of Signal	
	Detection Theory	66
	4.2.2.2 Pattern Elements	66
	4.2.2.3 Development of the	
	ROC Curve.	66
	4.2.2.4 Detecting Textural	
	Boundaries	67
4.2.3	Interval Coverings	71
	4.2.3.1 Orientation	71
	4.2.3.2 Formal Description . . .	71
	4.2.3.3 Remarks on Applications.	74
4.3	Structural Inference	78
4.3.1	Scene Segmentation	78
4.3.2	Structure Operation Language	
	Support	81
4.4	Applications	85
4.4.1	SEM Micrographs	85
4.4.2	Cervical Smears	86
4.4.3	Brain Mapping	90
4.4.4	Cytospectrometer	92
	4.4.4.1 Objective of a Cyto-	
	spectrometer	92
4.5	Computer Systems Support	96
4.5.1	System Documentation	96
4.5.2	Design and Fabrication	96
4.5.3	Control Point Strategy and Its	
	Automated Diagnosis	97

	Page
4.5.3.1	Introduction 97
4.5.3.2	Developments of the Past Year. 98
4.5.3.3	Current Status 98
4.6	Documentation 100
4.6.1	External Documents Issued 100
4.6.2	Logic Drawings Issued 101
4.6.3	Engineering Drafting Report 101
4.7	Administration 102
4.7.1	Personnel Report 102
4.7.2	Computer Usage Log Summaries 103
5.	ILLIAC IV 104
	Report Summary 104
	HARDWARE 106
5.1	Off-Line Diagnostics 106
5.1.1	PE Test 106
5.1.2	Card Test 106
5.2	Program Conversion - ATII 106
5.3	Development - ATII 106
5.4	ILLIAC IV Maintenance 106
5.5	Financial 107
	SOFTWARE 108
5.6	Operating System 108
5.7	Compilers and Translators 109
5.7.1	Cockroach 109
5.7.2	Clypnir 109
5.8	Assembler (ASK) 110
5.9	Simulator (SSK) 110
5.10	Interactive Communications and Graphics 110
5.10.1	Interactive Communications 110
5.10.2	Graphics 111
5.11	Library 111
	APPLICATIONS 112
5.12	Numerical Analysis 112
5.12.1	Partial Differential Equations 112
5.12.1.1	Block Jacobi Method 112
5.12.1.2	Numerical Solutions of Problems in Hydrodynamics. 116
5.12.2	Solutions of Systems of Linear Equations 117
5.12.3	Eigenvalues 118
5.12.3.1	The Eigenvalue Problem 118
5.12.3.2	Eigenvalues and Eigen- vectors of Symmetric Tridiagonal Matrices 119
5.12.3.2.1	QR-algorithm for Symmetric Tridiagonal Matrices 119

	Page
5.12.3.2.2 Inverse Iteration for the Correspond- ing Eigen- vectors. . . .	119
5.12.3.2.3 Iterative Method for Solution of the Eigen- problem . . .	120
5.12.4 Polynomial Root Finding	120
5.12.5 Time Series Analysis.	120
5.12.5.1 Identification and Estimation of Stochastic Models	120
5.13 Linear Programming	121
5.14 Signal Processing	121
5.15 Education	122
5.15.1 Teaching	122
5.15.1.1 CS-491: Architecture, Applications and Languages for a Parallel Computer. .	122
5.15.1.2 Two-Day Seminar for LRL. .	123
5.15.1.3 One-Day Seminar for Fluid Dynamics Workshop.	124
5.15.2 Consulting.	125
5.15.3 Training	125
5.15.4 Technical Writing	125
5.15.4.1 An Introductory Description of the ILLIAC IV System: ILLIAC IV Document No. 225	126
5.15.4.2 IEEE Article	128
ADMINISTRATION	129
5.16 Administration and Services	129
5.16.1 Contract Status	129
5.16.2 Financial Report	129
6. NUMERICAL METHODS, COMPUTER ARITHMETIC AND ARTIFICIAL LANGUAGES	133
6.1 Computerized Mathematics	133
6.2 Computational Geometry	133
6.3 Educational Timesharing System on the PDP-11 .	134
6.4 Debugging Interpreter/Trace System (DITS). . .	135
6.5 PDP-11 Hardware Progress Report	137
6.6 Factorization Techniques Used in the Solution of Partial Differential Equations.	138
7. THEORY OF DIGITAL COMPUTER ARITHMETIC	141
7.1 Evaluation of Some Elementary Functions in Radix 16	141
7.2 Continued Fraction Arithmetic	142

	Page
8. SWITCHING THEORY AND LOGICAL DESIGN	144
9. MACHINE AND SOFTWARE ORGANIZATION STUDIES	147
9.1 Fortran Parallelism Detection	147
9.2 Weighted Node, Directed, Acyclic Graph Schedule by m-Machines	150
9.3 Simulation Processor	151
9.4 Memory Hierarchies	152
9.5 Microprogramming and Control Unit Design .	152
9.6 S-Language Assembler	153
9.7 Text Searching	153
9.8 Debugging.	154
10. COMPUTER SYSTEMS ANALYSIS	156
10.1 Computer Network Modeling	156
10.2 Center Throughput Analysis	156
11. SOUPAC AND STATISTICS	157
11.1 SOUPAC Consultants	157
11.2 Research	158
12. GENERAL DEPARTMENT INFORMATION	160
12.1 Personnel	160
12.2 Bibliography	161
12.3 Colloquia.	163
12.4 Drafting	164
12.5 Shops' Production	164

1. CIRCUIT RESEARCH

(Supported in part by the Office of Naval Research under Contract N000 14-67-A-0305-0007, W. J. Poppelbaum, Principal Investigator.)

1.1 Bundle Processing (Project No. 21)

1.1.1 Bundle Signal Repeater and Restorer

During this quarter the design of the signal restoring section of the system was completed. Currently work is being done on the input portions of the bundle repeater and that of the bundler restorer. The signal readout section for both the repeater and restorer is also being designed.

Construction of the system will begin during the next quarter.

Bernard Tse

1.1.2 SABUMA

The final additions to the system have been designed and are being built. These are the error detecting circuits and their sequencing circuitry.

The bundles have been built and are being installed. The only job remaining, prior to the system being complete, is to install the display panel.

Trevor Mudge

1.2 APE (Project No. 25)

1.2.1 The 42.5 MHz Clock Receiver

Over the past quarter, much effort was devoted to the development of AM receivers for the APEs. In the design of these receivers, the overriding factor is low power consumption. This precludes the use of some newer devices, such as dual gate MOSFETs, as the active elements. Bipolar junction transistors are used in both the clock receiver and the data receiver.

Figure 1 shows the schematic diagram of the 42.5 MHz receiver.

It is a two-stage tuned RF amplifier followed by an AM detector. The carrier frequency of 42.5 MHz is chosen to be much higher than the band of frequencies for the data channels. Furthermore, the subharmonics of the clock carrier frequency also lie outside this band. The sensitivity of the clock receiver is about 500 μ V. It draws 2.5 mA from a 6 volt power supply, and has a 1 MHz bandwidth to receive pulses as little as 2 μ s wide. Due to the binary nature of the clock signal and the careful frequency assignment to avoid

T₁ : HEATH GD-19-40-916 TRANSFORMER
 T₂ and T₃ : DALE IPB-2042-31 RF TRANSFORMERS

Figure 1. The 42.5MHz Clock Receiver

interference, automatic gain control is not necessary for the clock receiver. RCA transistor 40245 was chosen for the clock receiver because of its exceptionally high gain at a low bias level. Although it is not shown in the schematic, provision is made for the insertion of a neutralization capacitor between the collector tank circuit and the base of the transistor in the tuned RF stages. The divider in the base circuit of the detecting stage is to bias the transistor just below the threshold of conduction. When the 42.5 MHz signal appears at the input to the clock receiver, the detecting transistor will be driven into saturation and will remain there as long as the 42.5 MHz carrier frequency appears at the input to the clock receiver. Since the 42.5 MHz carrier is switched on and off according to the clock signal, the output from the clock receiver produces an inverted clock signal.

Yiu Kwan Wo

1.3 PENTECOST (Project No. 31)

1.3.1 Testing the Land System for Sequential Presentation

Two slide projectors were used to project the red and green fields of the Land Color system. A plastic wheel, half of it painted black, was rotated by a synchronous motor at 1800 rpm. The wheel was placed between the two projectors. By rotating the wheel the fields were successively projected on the screen at a field frequency of 30 cps. The Land effect was still observed with this sequential presentation. The proper combination of filters to give the best color effect is now being investigated.

The camera for the Pentecost system has been ordered from General Electric. An Amperex red extended Plumbicon was ordered for this camera.

G. Panigrahi

2. HARDWARE SYSTEMS RESEARCH

(Supported in part by the Atomic Energy Commission under Contract
US AEC AT(11-1) 1469, W. J. Poppelbaum, Principal Investigator.)

2.1 LASCOT (Project No. 09)

2.1.1 Laser Specifications

The specifications of the laser needed for LASCOT have been derived. A mixed gas (argon-krypton) is needed, with an output of 200-250 mW in both blue ($4880\text{\AA}$) and red ($6471\text{\AA}$) and 75 mW in the green ($5145\text{\AA}$). A survey of the available lasers has been made.

M. N. Cooper

2.2 OLFT

2.2.1 Intermediate System Tests

An intermediate version of the cooled-crystal light-valve has been constructed and consists of the entire cooling assembly of the new system mounted in the vacuum chamber of the old system, using the old electronics and optics. This facilitates testing of the cooling components - sample crystal and substrate, thermoelectric elements, heat sink, temperature sensors and control circuits - under conditions independent of the new electron beam system. This intermediate system can be used only in a single-frame mode, requiring manual frame gating and erasure.

Initial testing has begun on this system, and the results have led to several conclusions. The first cooling cycle was manually controlled, at a rate of about 20°C per hour. Unfortunately, the substrate cracked at -6°C . Although this test substrate is not mounted exactly as planned for the new system, it is obvious that this is a critical problem. Also, to minimize stress, the temperature cycle should require on the order of several hours for complete cooling or warming. The temperature control circuits are

now designed for steady-state operation, and should be redesigned for automatic control of the temperature cycle.

Fortunately, the substrate damage affected only a corner of the crystal, so that tests will continue, specifically covering crystal time constant, uniformity of crystal characteristics, assembly aging and surface damage, secondary emission, etc. This should result in predictable performance when the new crystal assembly is constructed.

Douglas Sand

2.3 ORBIT (Project No. 15)

2.3.1 Receiver Synchronization

The low bandwidth synchronization scheme, designed by Ed Carr, which has been discussed in its various phases in these more recent reports, has been built and is now working. A full description of the design and its implementation will shortly appear as an M.S. thesis. This brings the ORBIT project, therefore, to its conclusion.

M. Faiman (ed.)

2.4 Tricolor Cartograph (Project No. 16)

2.4.1 Physical Wiring

The switches and potentiometers have been wired into the control panel and connections made in the display junction box to allow for them. Work was done on the lightpen to make it pin for pin compatible with the Artrix pen. A new pen enable circuit was installed. Wiring was also done in the main junction box. A platform was built for the Corning delay line, which is now mounted in its place.

The first phase of the store outline circuitry described in the last report has been wired and works as expected.

2.4.2 Power Supply

Finally, a power supply was designed and tested for the Corning delay line. The schematic, shown in Figure 1, also shows the overvoltage protectors.

Don Hanson

2.5 BLAST (Project No. 19)

2.5.1 Screen Signal

As will be recalled from a previous report, the screen signal was to be picked up by two interwoven combs and amplified by a differential amplifier which was floating at the second anode potential (approximately 18 kilovolts). A reasonable amount of success was achieved using this method, as reported last time.

Unexpectedly, during the last quarter, the internal connections to one of the conducting combs open circuited. This precludes the use of the interwoven combs to obtain a synchronizing signal; although one comb is still connected it may break down at any time. The reason for the breakdown is not known, but it is conjectured that the "sandwich" type of connection used to contact the nickel conducting pad on the glass screen was constructed of incompatible metals causing a nonconducting layer to be formed between layers of metal. Therefore, a second approach to the problem of aligning and synchronizing the combined video information behind the vertically oriented lenses is being tried.

2.5.2 Scan Correction

It is certain that the monitor scanning pattern can be controlled so that the video information is aligned with the lenses. An initial attempt at this proved quite successful. Using seven surplus relay coils placed ahead of the yoke in close relation to the tube envelope, a static correction was applied experimentally to the raster. The correction was sufficient to provide alignment for perhaps, one-third to one-half of the raster. In this area the 3-D effect was quite striking to many observers. It is apparent, however, that satisfactory alignment over the entire raster can only be accomplished using a dynamic correction. To this end a diode shaping waveform generator has been constructed and coils are being driven dynamically to determine the actual correction needed. Moderate success has been achieved in this area. It is felt that this approach is almost certain to succeed and will by-pass the troubles that have plagued BLAST due to the manufacturing of the special tube with its conducting stripes.

Larry Wallman

2.6 Eidolyzer (Project No. 23)

Eidolyzer is finished; Art Simons has his Ph.D. The interested reader may peruse his thesis: "Eidolyzer: A Hardware Realization of Context-Guided Picture Interpretation," D.C.S. Report No. 448, June 1971.

Demonstrations of the machine may be had in Room 231, Digital Computer Lab (by appointment, please).

M. Faiman (ed.)

2.7 Semantrix (Project No. 24)

The circuit boards for the sense amplifiers have been constructed and are under test. The control logic has been designed and fabrication is under way. This will be tested out upon the receipt of a teletype now on order.

Trevor Mudge

2.8 LINDA (Project No. 28)

The goal of this project is to build a LINE Drawing Analyzer capable of recognizing a number of simple line drawings. A flying spot scanner is used to project a given drawing on an oscilloscope. Photocells are used to extract edge information from the drawing. The information from the photocells will be decoded to produce an identifying output.

2.8.1 Project Status

Past work has been focused on building a circular array of photocells, expanding the pattern to the circle's edge, and trying to find a simple decoding scheme. A flying spot scanner was obtained in January and deflection circuits and a video amplifier have been built. Expansion of the figure was originally planned to result from the lowering of the accelerating voltage across the CRT. Work this past quarter indicates this method is not the best since focusing and intensity problems cannot be conveniently overcome. Expansion of the figure will be accomplished by raising the deflection voltage of the flying spot scanner - work on this has not yet begun.

Due to lack of a simple decoding scheme for the circular array, work was temporarily stopped and new work (particularly during the last quarter) was begun on a linear array.

The principal of the linear array is illustrated in Figure 1, using a simple triangular figure. The triangle is swept across the photocell line, turning on every photocell which comes into its boundary. The sum of the photocell signals is taken and is termed the ordinate sum. The ordinate sum signal is differentiated twice and, ideally, the second differentiation produces one pulse for each discontinuity in the figure. This analog decoding scheme provides a simple method of recognition for polygons by counting pulses.

Figure 1. Illustrating Principle of Linear Array

To date a linear array consisting of 10 photocells has been built and tested. Figure 2 shows the circuit used to sweep the pattern across the array. Results have been promising but not entirely satisfactory. The photocells used have a 1/5" diameter and do not provide sufficient resolution. The ordinate sum signal is not smooth, and it is difficult to distinguish a 5 from a 6 sided figure.

2.8.2 Future Work

Two techniques will be tested to improve resolution. First, fiber optics will be employed. Small fiber bundles can be grouped closely in a linear array to carry light to photocells which are less closely spaced. Second, two or more photocell lines can be staggered, as shown in Figure 3. The signal from the first line is delayed, so that all photocells appear to be in the same line. The fiber optics technique will be tried first. The delayed array will be used if needed. It is hoped that patterns having up to 15 sides may be distinguished.

Dick Blandford

Figure 2. Circuit to Sweep Pattern Across Linear Photocell Array

Figure 3. Staggered Photocell Lines

2.9 Stereomatrix (Project No. 30)

2.9.1 Transformer

All the cards for the transformer has been tested. The rack wiring is done and debugged. The transformer has been tested and run for several hours.

The outputs of PAGAN, the three-dimensional pattern generator, have been used as input to the transformer to see two different pictures (one for each eye). The transformer output was connected to a CRT and a change in perspective was observed by a simulated moving of the observer's position.

Some changes in the cards of the perspective generator have to be made so that output of transformer is compatible with the deflectors. It is hoped to accomplish this in the next quarter.

Shiv Verma

2.9.2 Display

Some further refinements on the laser power supply were made. The low-voltage control board was removed from the supply and driven with a pulse generator to determine its transient response. As a result of this study a capacitive divider was removed from the control board. Also, a heat sink and diode isolator were added to the 22-volt zener on the control board. No transistor or zener failures have yet occurred with this modified circuit. Remaining to be changed is the start relay section of the control board. This section was previously changed from a differential amplifier to a monostable. However, large line transients caused by nearby motor start-up and some control settings still incorrectly trigger the start transformer on.

Two ultrasonic deflectors have been received and assembled into a two-dimensional display. Geometric line patterns with no blanking have been displayed using signals from PAGAN. The patterns are reasonably bright in a darkened room and dimly visible with normal room light. In this test set-up the laser light encounters eight uncoated lenses and two coated deflectors. The extinction is excellent for a crossed polaroid. This is using a temporary aluminized lenticular front projection screen.

Simple emitter follower circuits were designed to convert the input +10 volt signals to the -0.5 to -7.0 and -1.5 to -14.2 voltage levels required by the ultrasonic cell drivers. As the cells are rather non-linear, more sophisticated conversion amplifiers are being designed.

A third deflector has subsequently been received and the first unit is being returned for repair. The first unit developed a frequency jump problem. It is not as efficient as the second and third units either. As soon as it is returned the complete three-dimensional system will be assembled.

Steve Whiteside

2.9.3 Observer Position Detector

All digital circuit boards have been designed, built and tested.

The encoder assembly in its entirety has been built and some success has been achieved in reading the disc optically. Variations in the manufacture of phototransistors have altered their alignment with the optic fibers in the reader mount to the extent that some tracks cannot be detected. Also several sources are being considered which should provide uniform illumination to the source optic fibers.

During testing, some crosstalk between optic channels in the reader has been experienced. Previously .040" source light fibers were replaced with .020" source fibers to reduce crosstalk. It may be necessary to use .020" fibers also on the pickup side of the disc to insure proper channel separation. In the case where .020" fibers are installed, spacers of the same size are provided to separate and align them.

During the next quarter, the encoder readout will be improved and tested with the detector circuitry. The second reader assembly will be built and the analog circuitry constructed.

Chuck Pirnat

2.10 PAGAN (Project No. 32)

2.10.1 Completion of the Project

The PAGAN project (Pattern Generator Analog) reached completion during this quarter. The project has been documented in the form of a thesis. Further documentation is now being assembled into an operating manual.

At the beginning of the second quarter, three of PAGAN's circuits had been laid out on printed circuit cards. As much of the remaining circuitry was not highly repetitive, in the interests of economies of time and resources, most of those circuits were handwired on universal cards.

One circuit that is used a number of times was laid out on a printed circuit card. This card is the 8 x 10 programming matrix for storage of preset patterns. PAGAN's patterns may be specified by entering the appropriate digital increments and limits on front panel switches. As a convenience to users who do not wish to derive the proper control signals, sixteen preset

patterns are stored internally. The control signals for the preset patterns are stored on the 1469 - 491 card shown in Figure 1. The inputs of the TTL gates are used in place of the diodes in a diode matrix. The vertical and horizontal lines of the figure are on opposite sides of the printed circuit card. This wired - in storage is programmed by inserting a pin at a location indicated by an open circle and soldering it to the printed circuit on each side. d.

2.10.2 Suggestions for Improvements

The PAGAN project has met its desired goals as a pattern generator for refreshed display of three-dimensional geometric figures. The present circuitry is operating satisfactorily and is not in need of modification. However, the project can be extended, for example to include spherical and toroidal surfaces. If additions are considered, some circuit improvements might be contemplated.

If some of the hand wired digital circuits are to be converted to printed circuit layouts, use should be made of some of the newer MSI TTL devices. When changing from hand wired universal cards, a different partitioning of the circuitry could clear some of the congestion of the card cage back panel wiring.

It would be easier to adjust the analog circuitry for optimum performance if some modifications were made. The Sinusoidal Transconductors require $\pm 15V$ accurate to .01%. Currently, the sinusoidal waveforms are tuned up by adjusting the system's $\pm 15V$ supplies, which also affects the other analog modules. Any redesign of the 1469 - 485 card probably should include the addition of Integrated Circuit voltage regulators.

The card containing the analog switches has undergone some slight modifications this quarter. These switches are made up of discrete components and Junction FETs. If the circuit should require future adjustments, consideration should be given to incorporating Integrated Circuit MOSFETs.

Richard Partridge

1469-491
8 X 10 Programming Matrix

Figure 1. Programming Matrix Card

2.11 Scantrix (Project No. 35)

2.11.1 Possible Implementations

Four possible schemes for Scantrix were considered. The first one utilizes a rotating mirror to reflect light from a line of 128 light emitting diodes (LEDs), thus scanning the picture across a screen. The second approach (Figures 1 and 2) has four lines of LEDs instead, mounted on the surface of a cylinder, and with a 90-degree angle between each line. As the cylinder rotates, signals are fed into the lines of LEDs sequentially, to scan out the picture, as in the first method. However, the second approach, with four lines of LEDs, eliminates the jumping of the scanning device (the mirror in the first case) since the diodes are now oriented so that when one line reaches the bottom of the screen the next line will just come in at the top. But, both of these two approaches have to make use of fast rotating motors to rotate the scanning structures which are reasonably big and probably heavy. Besides, high intensity is required for the LEDs in order to give reasonably bright project.

The other two approaches require 128 x 128 LEDs to form the whole frame, the advantage of these electronic scanning approaches over the mechanical scanning approaches aforementioned should be obvious.

Refer to Figure 3 for the analog switching approach. In this scheme, the video signal from the TV is sampled at appropriate intervals and directly fed into the frame of LEDs through control gates. The requirement of gray levels and switching over the 128 x 128 (16,384) LED elements presents a financial problem, since a switch has to be incorporated with each of the LEDs in order to light each up in sequence in accordance with the TV scanning structure.

The last scheme uses digital switching, which can be economically achieved, made possible by the low price of digital ICs. Refer to Figure 4 for a simplified block diagram of the scheme. The video signal from the TV is sampled at appropriate intervals (128/line), as before. The sampled signals, instead of being fed directly into the LEDs, are changed to digital signals via an A/D converter. Every 128 sampled, digital signals, corresponding to one single line on the screen, are stored in a buffer. There are two buffers alternating in storing the signals. One is being fed with the signals while the other is being gated to the next stage, namely, 128 D/A converters. Thus the original signals are recovered and are used to drive the LEDs, one line at a time.

Figure 1. Rotating Drum with Four Lines of LEDs

Figure 2. Side View of Rotating Drum

Figure 3. Block Diagram of Analog Switching Approach

Figure 4. Block Diagram for Digital Switching

A closer look at the system reveals that an integrator will be needed to average out the video signals along each of the 128 intervals of the $62.5\mu\text{s}$ in between every horizontal blanking, in order to produce a reasonable value for the sampling circuit. Direct sampling is avoided because in each of the 128 intervals (very roughly $0.4\mu\text{s}/\text{interval}$) drastic differential change in the video signal can occur for a very short period of time. If integration is not provided, sampling at that particular instant when the spike occurs obviously gives a different value from that sampled at another instant. Since there are only 128 rows of LEDs on the screen, integration has to be made vertically also. However, this can be easily overcome by allowing our humans eyes to do the integration instead. In other words, appropriate numbers of lines are superimposed onto each row of LEDs (4:1, including interlacing).

Note that two buffers have to be used, each consisting of 128×4 bits (128 bits for 128 LEDs, 4 bits of gray levels for each point). One buffer is gated to drive the LEDs while the other is loaded at the same time.

Sik Yuen

3. SOFTWARE SYSTEMS RESEARCH

Publications this quarter:

Guimaraes, C. J. F. WEED: A Wonderful Equation Elimination Device, Department of Computer Science Report No. 444, University of Illinois, Urbana-Champaign, Illinois 61801; also, submitted in partial fulfillment for the degree of Master of Science in Computer Science, June 1971.

Ratliff, K. SPARSE MATRIX INVERSION, Department of Computer Science Report No. 443, University of Illinois, Urbana-Champaign, Illinois 61801, June 1971.

C. W. Gear, Professor
and Principal Investigator

3.1 Numerical Processes

3.1.1 Sparse Matrix Inversion (J. Deogun)

Sparse matrices occur frequently in engineering and industrial problems. Expert handling is needed to develop routines for (SPARSE) matrix operations. Currently we are concerned with the group of routines which generates matrix inversion code for the Jacobian of a system of equations. These routines have been combined with the main program and the subroutine DIFSUB via the subroutines P1 and P2. The function of DIFSUB is to integrate ordinary differential equations. These results are used in the corrector step of the integration process. The combination of all these subroutines is called the numerical package. Figure 1 gives the block flow diagram of the interaction of various processes which constitute the set of packages which perform the numerical and associated processing for transient analysis.

The subroutine SPARSE has been improved considerably. The section of SPARSE which selects pivot elements at each step has been successfully modified, and its present version is considered to have the "best" pivot selection algorithm. The meaning of the word "best" used in this context is explained in the latter part of this report.

Previously, various pivot selection strategies were tested only on small test examples. Recently, they have been tested with a practical (53 x 53) example. Various algorithms were compared and the basis for comparison is the number of instructions generated by subroutines MATMUL and MATINV.

Following are the results of this comparison on the 53 x 53 example:

No.	STRATEGY	NUMBER OF INSTRUCTIONS
1	Min Row--First Column	1547
2	Min Row--Min Column	1128
3	Min PRDT	1132
4	Min Column--Min Row	1076

ALL PACKAGES HAVE BEEN COMPLETED EXCEPT:

* AN INVESTIGATION OF TECHNIQUES
FOR SOLVING SPARSE EIGEN PROBLEMS
IS UNDER PROGRESS.

Figure 1

Pivot selection method 4 (i.e. Min Column--Min Row) has produced the best results with almost all examples. In this method a column with a minimum number of elements is taken as the pivot column. If there is more than one such columns, a chain of minimum columns is formed. Then elements in the minimum column(s) is scanned to choose a pivot. Each "eligible" element in the pivot column is compared by the number of elements in its row. An element is eligible for pivoting if it does not lie in either the row or column which has already been pivoted. It may be recalled here that all pivot selection strategies are dynamic; at each step the choice depends only on that part of the updated matrix which consists of unpivoted rows and columns. Choice of a pivot is also restricted by its size. An element may not be used as a pivot if it is less in absolute value than α times the largest element in its column. α is a constant. Values of 0.1 or 0.01 have been used in recent examples. Better results have been obtained for $\alpha = 0.1$.

The SPARSE program has been improved very much, particularly to minimize round-off error. The present version of pivot selection strategy, as described above, helped much in improving upon the round-off error. Subroutine check, a listing of which is given below, was added to the group of SPARSE routines. This subroutine calculates a relative error vector at various steps. It provided considerable use for minimization of round-off error.

```
0001      SUBROUTINE CHECK(DIN,DOUT,MX1,PW1,IRC,N,S,K1,L)
0002      IMPLICIT REAL*8(A-H,Q-Z)
0003      DIMENSION DIN(1),DOUT(1),MX1(6,1),PW1(1),IRC(1)
0004      DIMENSION PW1(1)
0005      DIMENSION T(60),TMAX(60)
0006      K=K1
0007      S=0.
0008      DO 3 I=1,N
0009      T(I)=0.0
0010      TMAX(I)=0.00
0011      J=IRC(I)
0012      2 IF(J.EQ.0)GO TO 3
0013      Q=MX1(6,J)
0014      IF(MX1(1,J).GT.1)Q=PW1(MX1(6,J))
0015      IF(MX1(1,J).EQ.0)Q=0.
0016      T(I)=T(I)+Q*DOUT(MX1(3,J))
0017      TMAX(I)=DMAX1(TMAX(I),DABS(T(I)))
0018      J=MX1(4,J)
0019      GO TO 2
0020      3 CONTINUE
0021      6 IF(K.EQ.0)GO TO 5
0022      I=MX1(2,K)
0023      J=MX1(3,K)
0024      T(I)=T(I)+DOUT(J)
0025      TMAX(I)=DMAX1(TMAX(I),DABS(T(I)))
0026      K=MX1(5,K)
0027      GO TO 6
0028      5 IF(L.EQ.0)GO TO 8
0029      DO 9 I=1,N
0030      9 DIN(I)=T(I)
0031      RETURN
0032      8 DO 7 I=1,N
0033      TMAX(I)=DMAX1(TMAX(I),DABS(T(I)-DIN(I)))
0034      IF(TMAX(I).EQ.0) GO TO 7
0035      S=S+DABS((T(I)-DIN(I))/TMAX(I))
0036      7 CONTINUE
0037      WRITE (6,4) S
0038      4 FORMAT(1H ,7HCHECKS=,D15.5)
0039      1 CONTINUE
0040      RETURN
0041      END
```

SPARSE in its present form is considered quite efficient. The combination of all subroutines in the numerical package has been run successfully with small test examples. Currently, the package is being tested with the 53 x 53 example. Except for minor changes in P1 and P2, the package runs successfully. Relation between various routines or sections of the numerical package are shown diagrammatically in Figure 2.

THE EIGENVALUE PROBLEM

Reasonably efficient routines are available for the solution of the standard eigenvalue problem $Ax = \lambda x$. Our present concern is the more general problem

$$Ax = \lambda Bx$$

where A and B are of sparse nature.

If A and B are both singular and of rank appreciably lower than their order, the problem can be handled by Wilkinson's method.* The SPARSE eigenvalue problem is closely related to the SPARSE matrix inversion and has been divided into three phases:

1. Changes in the subroutine set up.
2. Changes in the pivot selection strategy.
3. Changes in the subroutine SPARSE.

Changes in subroutine SETUP have been completed. The changed routine has yet to be tested in combination with other routines. Changes in pivot selection strategy and subroutine SPARSE are presently underway.

The subroutine SETUP has been changed to obtain the following configuration of matrix A and B. Where columns are concerned, matrices A and B appear to be lying side by side so as to give a $(N \times 2N)$ matrix.

* $Ax = \lambda Bx$ and the Generalized Eigen Problem, G. Peters, and J. H. Wilkinson, SIAM Journal on Numerical Analysis, 7, #4, December 1970.

Figure 2

Where rows are concerned, the matrix A appears to be riding over B so as to give a $(2N \times N)$ matrix. The situation is shown in Figure 3. This configuration is considered convenient for handling by pivot selection strategies and the SPARSE program.

Figure 3

3.1.2 The Steady State Package (B. van Melle)

A new version of DIFSUB was prepared and tested this quarter employing a new strategy for solving the steady state problem ($MF = 3$). The integration is started as before, but only one pass is made through the corrector loop. After each step, the DY's are checked to see if they are converging

$$(D = \sum_{i=1}^N (DY(i))^2 \text{ is reduced}).$$

If so, certain parameters are changed to bring the process towards a Newton method. These parameters are H, the step size, which is made to approach the value of 1, and ALPHA, used in the corrector loop, which starts at 1 and moves toward 0. However, if D is not reduced, we restore ALPHA to 1 and reduce H in an attempt to reduce D.

Several strategies were also tested to minimize the number of function calls and matrix inversions. DIFFUN is called only once for each successful step, and the Jacobian is re-evaluated only when the previous step is unsuccessful (D was not reduced) and then only if it has not been recently re-evaluated. The problem then becomes one

of moving toward a reasonable rate of convergence without "backing up" too often. The Newton method converges rather quickly when near the solution. But the rate at which we vary H and ALPHA to approach this method is important. On the simpler examples, the sooner we go to the Newton method, the faster the convergence, but in general it appears more efficient to proceed cautiously.

The new DIFSUB using these ideas was tested with several examples, and the results follow, with a comparison of the same examples run with the old DIFSUB.

- Ex. 1 DY: $\sin(y_1 y_2) - y_2/2\pi - y_1$
 $(1 - 1/4\pi)(\exp(2y_1) - e) + ey_2/\pi - 2ey_1$
- Ex. 2 DY: $y_1^2 - y_2 + 1$
 $y_1 - \cos(\pi y_2/2)$
- Ex. 3 DY: $4 + y_1 + y_2 - y_1^2 + 2y_1 y_2 + 3y_2^2$
 $1 + 2y_1 - 3y_2 + y_1^2 + y_1 y_2 - 2y_2^2$
- Ex. 4 DY: $y_1^2 + y_2^2 + y_3^2 - 5$
 $y_1 + y_2 - 1$
 $y_1 + y_3 - 3$
- Ex. 5 DY: $\exp(y_1) + y_1 y_2 - 1$
 $\sin(y_1 y_2) + y_1 + y_2 - 1$
- Ex. 6 DY: $y_i' + G(i)(y_i - 1) - (y_i - 1)^2, i = 1, 2, 3, 4$
 $y_5 (G(1)T(3)/(T(3) + G(5)) - y_1) + y_6$
 $2y_7 - y_7^3 - y_6 - T(2)$
 $\ell_1 - y_7 - y_6 + 1$
 $\ell_2 - \ell_1 - (y_1 + y_3 + y_4 - y_2)/2 + 1$
- S1: $G(5) = 1 + G(1)$
 $T(2) = \exp(-T(1))$
 $T(3) = \exp(-G(1)T(1))$
- Ex. 7 DY: $T(2) + y_1 - 6y_3 - y_3'$
 $y_3 - y_2'$
 $y_2 - y_1'$
 $\ell_3 + 11y_2$
 $\ell_2 + 6y_1$
 $\ell_1 - \ell_2 - \ell_3$
- S2: $T(2) = \sin(T(1)/10)$

All examples used the initial values

$$y_i = 1, i = 1, \dots, N - NL$$

$$i = 1, i = 1, \dots, NL$$

and convergence was determined by the condition

$$\sum_{i=1}^N (DY(i))^2 < 10^{-6}$$

	OLD DIFSUB			NEW DIFSUB		
	NS*	NFNS	NW	NS	NFNS	NW
Ex. 1	130	678	120	116	149	10
Ex. 2	12	37	7	21	29	3
Ex. 3	24	66	8	21	32	4
Ex. 4	21	70	12	22	33	3
Ex. 5	13	41	8	19	27	3
Ex. 6	164	1739	162	19	42	3
Ex. 7	15	55	9	12	19	2

NS - number of integration steps

NFNS - number of function evaluations

NW - number of Jacobian evaluations/matrix inversions

3.1.3 Steady State Package Testing (W. Chung)

Graphics-User Interface Routine TRNØUT*

The subroutine TRNØUT, which is called in subroutine P2 for the transient analysis, is written and tested. It is composed of three parts which perform the following functions:

1. Communication with a user to obtain data
2. Integration by calling DIFSUB
3. Graphics plotting on the screen.

* SUBROUTINE TRNØUT(N,NL,T,G,Y,YL,SAVE,H,HMIN,HMAX,EPS,MF,YMAX,
1 ERROR,KELAG,JSTART,MAXDER,PSAVE,YLSV,F,DY,ERSV,F1,IP,IQ,
1 DEL,TEND,YINI)

Data which is to be specified by a user for the purpose of plotting is as follows:

1. Variable names (Y and/or YL) to be plotted.
2. TEND maximum time limit.
3. IPNT number of points where the integration and plotting are requested.

The program has been modified through three steps to provide more flexibility and sophisticated features such as error checking and multiple plotting. Simple descriptions are given below.

1. First version--simple and crude.
 - a. only a single variable can be plotted.
 - b. it calls the subroutine pair MESSAGE/REPLY to communicate with a user at a graphics terminal. User-provided data is recognized and converted to binary by a FINDNM/NUMCHR pair. However, since subroutine FINDNM handles only fixed point numbers, the above procedure should be repeated for floating point numbers, which is considered very inconvenient.
 - c. Plain axis and curve are drawn without any scaling and identification.
2. Intermediate version.
 - a. multi-variable plotting in a very restricted manner.
 - b. use of unit 1 and 2 for graphics I/O--no problem for floating point numbers.
 - c. to compute the function value at an exact point of time, Y variables are interpolated from the derivatives. Since we cannot interpolate YL variables, points where the actual plotting is done are saved in an array TSV(I).
 - d. user is allowed to change parameters and start again.
3. Present version.
 - a. flexible multi-variable plotting, which allows any Y and/or YL variables (total number is limited to 15

because of visibility on the screen) to be plotted in any order at a time.

Names of Y and YL are stored in arrays IYN and IYL, respectively, and the storage used is optimized by saving the integrated values of Y and YL at plotting points in a two-dimensional array YSV(I, J) such that Y(1, IYN(I)) are stored in YSV(I, J) and YL(IYL(I)) in YSV(I+NY, J) where NY is the number of Y variables to be plotted.

- b. interpolation of Y is eliminated not only because time spent in computation is not worthwhile but also because it is incompatible with YL variables if both Y and YL are plotted using the same time axis.
- c. error checking and recovery features are provided; i.e., each datum which is specified through the terminal is checked for errors, and branching is taken to various points in the program according to the error conditions.
- d. axes are scaled to aid reading. Each variable is identified by putting a function name at the tail of each curve.

Communication with a user (question-answering) is implemented by using two kinds of I/O sequences.

1. WRITE(2,*)/READ(1,*) pair.

This is used mainly to get numeric data.

2. CALL MESSAGE/CALL REPLY pair.

This is used for non-numeric data.

Test of TRNOUT on batch was done only for the numerical portion of the program because conflicts between units 1/5 and units 2/5 could not be resolved at the time.

Modification of P2

Due to the implementation of TRNØUT, a subroutine P2 now calls TRNØUT instead of DIFSUB. Also, initial values of Y and YL are saved in an array YINI(40) immediately before P2 calls TRNØUT. These saved values are used for initializing Y and YL everytime DIFSUB is called in TRNØUT.

3.2 Non-Numerical Packages

3.2.1 Item Analysis (J. Koch)

Item analysis was modified this quarter to allow it to be run from batch or interactively with the PDP-8. New commands and error messages were added and the section to create the node/connection table was re-written. The equation format was changed by the addition of an extra word to identify the equation to the user.

Using the graphics filing system, item analysis is able to retrieve any element, network, or mnemonic the user has previously sent to the 360 (see command list). When a picture is sent from the PDP-8, it is immediately stored in PICLIB; then item analysis creates and places the coded output data structure onto ITEMLIB. For PICLIB, the key is created from the six character picture name followed by two bytes that indicate which block of the picture it is (i.e., '0800' for header, '0801' for local lines, '0802' for comment text, '0803' for subpictures, '0805' for node connections and terminals, '0806' for declarations, '0807' for equations and parameters). Itemlib also uses the six character picture name, but appends two EBCDIC blanks ('4040') to the key. Also, any node type declarations that are discovered are written out on NODLIB with their eight character name as a key. The fourth file used is ERRLIB which contains all the syntax errors detected by item analysis. These errors are sent to the PDP-8 to be displayed on the screen and are also written on the line printer for hard-copy reference. At present, error messages appear with no specific reference to the line that was being examined at the time the error was noted. This may be remedied by outputting the error message followed by the line it references. ERRLIB can also be destroyed by the user after he has noted all the errors. The

key for the error message is created by the filing system through the use of the XAPPEND command.

The names of these four files were modified so that they are each preceded by the eight character log-on name of each user. Otherwise, every user would be working from the same files and could modify or delete the work of other users. (At present, all test elements are in POOHBEARPICLIB, etc., to allow them to be referenced when item analysis is run from batch). Upon entry to item analysis, the address R1 + 12 is the eight character user log-on. This name is then moved in front of the four file names before any commands are processed.

An extra word was put in each equation following the first word.

1st WORD

0

Displacement to next equation or zero

2nd WORD

4

Equation number in indicated block

2 bits indicating block:

00--element equation	resolved in
01--network equation	Global Analysis
10--default equation	resolved in
11--parameter equation	Item Analysis

PARSE is the subroutine that parses all the equations in an element. Some new parameters and bit indications were added. The complete parameter list is:

PFLG	DS	X
VLHBIT	EQU	X'80'
HALTBIT	EQU	X'40'
ERRBIT	EQU	X'20'
PADDR	DS	AL3
PEND	DS	A
VLOC	DS	A
VGLB	DS	A
VPAR	DS	A
VEI	DS	A
VLH	DS	A
VCON	DS	A
FRES	DS	A
LASTEQN	DS	A
COUNT	DS	A
S8WORK	DS	A
ERRLIST	DS	A
TRANS	DS	A
FILE	DS	A

POSSIBLE FLAG BITS:

- ONLY 1 VARIABLE ON LEFT SIDE OF =.
- ON IF PARSE HALTED.
- ON IF ERRORS DETECTED BY PARSE.
- ADDR OF WHAT I AM TO SCAN.
- END OF WHAT TO SCAN.
- LOC OF LOCAL VARIABLES.
- GLOBAL.
- PARAMETERS.
- E/I NAMES.
- LH SIDE VARIABLES.
- CONSTANTS.
- WHERE TO GET FREE SPACE.
- ZERO WHEN STARTING NEW EQN TREES.
- EQN NUMBER IN BLOCK.
- ADDR OF WORK AREA FOR MESS. TO PDP8.
- ADDR OF ERRLIB DCB FOR PARSE ERRORS.
- ADDR OF EBCDIC TO ASCII TRANS. TABLE.
- ADDR USED BY XAPPEND IN PARSE.

PARSE can now handle any number of expressible in double precision floating point notation. Also, exponents of 10 can be indicated through the use of either a D or an E (i.e. 0.173768D-10, .987E30).

Command List

(ALPHA is a valid six character element or network name)

1. 'ITEMIZE ALPHA'--causes ALPHA to be read from PICLIB and put through item analysis.
2. 'ANALYZE ALPHA'--causes item analysis to call global analysis with the parameter ALPHA.
Item analysis does no processing of ALPHA.
3. 'DESTROY ERRLIB'--causes item analysis to delete the user's error library. This command is only valid for destroying the error library and will not be recognized if any other file name but ERRLIB is used.

Messages to User

1. 'SYNTAX ERRORS DETECTED.'
If errors detected by item analysis.
2. 'UNKNOWN COMMAND.'
An illegal command was received.
3. 'INVALID BLOCK TYPE RECEIVED.'
An error in transmission occurred.
4. 'ITEM READY FOR FIRST INPUT'
Indicates item analysis is loaded and ready.
5. 'ITEM ANALYSIS COMPLETE'
The analysis of an element is done and item analysis is ready for another input.
6. 'RECORD XXXXXX-XXXX NOT FOUND. ENTER AGAIN.'
When an 'ITEMIZE' command is given and the indicated block does not exit on PICLIB.
7. 'MNEMONIC STORED.'
After user has sent a mnemonic.
8. 'ERRLIB GONE...'
After the 'DESTROY ERRLIB' command is completed.

Error Messages

1. 'NO : IN NODE TYPE DECLARATION. NAME: E(X,Y) I(P,T9)'
i.e., 'ELECTRIC E(A,B), I(C,D)'
2. 'NO) IN NODE TYPE DECLARATION. NAME: E(X), I(A,R)'
i.e., 'ELECTRIC : E(A,B), I(X)'
3. 'VARIABLES IN NODE TYPE DECLARATION NOT E OR I.'
i.e. 'ELECTRIC : W(A,B,C), Z(Q,R,T)'
4. 'VARIABLE REDEFINED.'
Variable defined twice as either a global, local, or parameter.

5. 'INVALID ENDING CHAR IN PARAMETER DEFINITION.'
Must end with a single variable or an equation.
6. 'NO E OR I VARIABLES SPECIFIED FOR TYPE.'
Just a type with no variables was entered.
(i.e., 'ELECTRIC :')
7. 'NON-EXISTANT TERMINAL TYPE REFERENCES.'
When a terminal has been assigned a type but the type was later deleted from the list of types.
8. '***** ILLEGAL XXXXXX DETECTED'
Message originates in PARSE and XXXXXX refers to the level of compilation at which the error was detected in an equation.

3.2.2 Global Analysis I and II (S. Wilkins)

When errors occur during the numerical analysis of a network, symbolic information known to the user must still be available in order to provide valuable diagnostics. All tables created by Global Analysis II have been expanded to facilitate the diagnostic process. During Global II a tree is constructed reflecting all the networks and elements in the structure of the outer network and their interrelationship.

If a network had the following structure

Then the tree constructed would reflect this structure as:

The complete description of the linkage of this tree structure was given in an earlier report.

Each entry in the tree structure is 30 bytes containing the following information.

<u>BYTES</u>	<u>TYPE</u>	<u>DESCRIPTION</u>
0-1	Numeric	Disp* to 1st element in subpicture if this is a network; zero otherwise.
2-3	Numeric	Number of local variables associated with this element
4-5	Numeric	Disp from top of local variable list to local variables associated with element.
6-7	Numeric	Disp to next element on same subpicture level or if bit 0 = 1 then bits 1-15 are disp to network of which this element is a subpicture.
8-9	Numeric	Displacement to Listone associated with this element.
10-11	Numeric	Number of entries in Listone.
12-13	Numeric	Subpicture level of this element counting from level 1 which is associated with main (outer) network.
14-15	Numeric	[#] Number of global variables associated with this element.
16-17	Numeric	[#] Disp from top of global variable list to the variables associated with element.
18-19	Numeric	Number of parameters associated with element.
20-21	Numeric	Disp from top of parameter list to parameters associated with this element.
22-29	Alpha	Symbolic name of element.

*All displacements unless otherwise explained are calculated from the top of the internal tree.

[#]Not always used since some Global Variables are placed in the internal variable list rather than the global variable list.

U. S. ATOMIC ENERGY COMMISSION
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR
DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO.

C00 1469-0193

2. TITLE

QUARTERLY REPORT - April, May, June 1971

3. TYPE OF DOCUMENT (Check one):

- ☒ a. Scientific and technical report
☐ b. Conference paper not to be published in a journal:
 Title of conference _____
 Date of conference _____
 Exact location of conference _____
 Sponsoring organization _____
☐ c. Other (Specify) _____

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

- ☒ a. AEC's normal announcement and distribution procedures may be followed.
☐ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
☐ c. Make no announcement or distribution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

W. J. Poppelbaum, Professor of Computer Science and Electrical Engineering
Editor: M. Faiman

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature

W. J. Poppelbaum

Date

July, 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

- ☐ a. AEC patent clearance has been granted by responsible AEC patent group.
☐ b. Report has been sent to responsible AEC patent group for clearance.
☐ c. Patent clearance not required.

Listone is a table for every element occurring in the structure of the network giving the E and I variable names assigned to each terminal of the element.

Each entry (one per terminal) has been expanded to 10 bytes.

bytes	0	2	4	6	8
					PDP number assigned to this terminal (known to the user).
					Disp from top of terminal name table to terminal type description for this terminal.
					-1 if no I-variables on terminal 1 if I-variables have minus sign 0 if I-variables have plus sign
					First I-variable name assigned to this terminal. Other I-variables are assigned the successive numbers following the first.
					First E-variable assigned to this terminal. Other E-variables are assigned the successive numbers following the first.

The three variable tables have been expanded to 10 bytes per entry.

bytes	0	8
		Displacement from top of internal tree to entry associated with element where this variable is defined.
		8 character symbolic name of the variable.

The beginning of each equation in the output equation block has been expanded to include an extra four bytes of information.

DISPLACEMENT
INFORMATION
Equation

Information bytes

3.3 Graphical Remote Access Support System (GRASS)

3.3.1 Disk Monitor System (M. Michel)

Production use of the DMS continued this quarter without the occurrence of any fatal errors. (c.f. section 3.3.4)

3.3.2 Display Terminals (R. Haskin)

The display terminal control program, ACID, has been used extensively this quarter, and has survived with only minor modification.

A slight improvement was made to the dynamic buffer deallocator which enabled it to release buffers without blowing up.

Heavy multiterminal operation revealed that hardware noise problems (very) intermittently cause data errors and consequent loss or garbling of display files. Several software modifications were implemented to lessen the effect of this problem.

Another minor hardware problem noticed was the again very intermittent failure of the terminal to return coordinates after a read cursor operation (joystick hit). Hitting any key on the terminal keyboard twice will recover from the problem.

All in all, the performance of both ACID and the Computek terminals has been satisfactory, and this part of the system can now be considered as fully operational.

3.3.3 Information Retrieval Package (M. Michel)

Use of the package became more regular and heavy this quarter. Response remained good as the number of files in the directory increased. Resorting of the block list during an operation was extensively tested and shown reliable. Two errors were found and corrected. Saving a file with the overlay option, if the file named already existed, resulted in improper deletion of the blocks of the existing file. This caused some free blocks to be lost from the unsorted free-list. Also, the pointers controlling the unsorted free-list were not being saved and restored by GLØAD/GEXIT.

The version of the package sent to the Griffiss Air Force Base, Rome, New York, has been modified by them, quite easily, to run with a DEC RF08 disk. Debugging of their version is just entering its initial stages, but the outlook for bringing the system up rapidly is quite good.

3.3.4 Disk Interface (C. Hyde)

The disk and associated hardware miraculously survived two floods and numerous power-downs (all due to malfunctioning air conditioning equipment). Only a few minor track failures occurred. An occasional, though devastating, error that has appeared but eluded discovery, is the loss of the first two words of a record on a write operation. A new test program was written, but failed to pinpoint the source of the error.

3.3.5 Monitors (M. Michel)

GRASP (360 Remote Monitor)

A perfect performance record was scored this quarter; no abnormal termination occurred and all functions operated properly and efficiently. Some additions were made to increase error detection of bad 2701 records; namely, extra and more complete log messages. The only contemplated alterations remaining are procedures for performing accounting operations, such as counting subtask I/Ø, attaches, logon/logouts, etc.

2701 Data Link

Adequate performance has continued on the data link. No serious problems have occurred.

GLASP (PDP-8 Local Monitor)

Extensive file-menuing ability was added this quarter. Saving or deleting a picture or mnemonic causes the appropriate function to be performed in the list of file names maintained in a special menu file for each user. The alphabetical list can be displayed in segments of 32 lines each during the operation of the generalized drawing package (GNDRW). Indicating an origin in the draw area and then "picking" a displayed mnemonic with the joystick causes the desired mnemonic to be integrated into the current picture data structure and to be appropriately displayed at the origin. At logout, the new menu file is saved in I.R.; the next time the same user logs on, his new menu will be displayed.

An option was added to CALLER due to the presence of the menu feature: purgeout. This causes all files named in the menu file to be deleted locally, then a regular logout is performed. Hence, a user can easily free the library space he used to make it available to others currently on the system.

The time of day has also been added to the upper-righthand corner of the display area. Whenever the display is regenerated, the updated time is also added in the form HH:MM:SS (hours, minutes, seconds).

Having the time available has enabled the addition of still another feature to CALLER: a system log. Each time a user logs on, logs out, or unsuccessfully attempts to log on ('threat'), the time, terminal number, event, and user ID (or logon string in the event of a 'threat') is output to the TTY printer.

Production sessions for two and three terminals have been held and were quite successful. Response time was adequate at each terminal, even for the worst case of three simultaneous picture regenerations. In addition, extensive testing and debugging of the lockout feature of the monitor (and associated DSMN routine) was performed. Two and three terminals were driven from a total of six core pages (as opposed to the usually available thirty-two) for small, medium, and high complexity pictures. All lockout situations were properly caught.

GLASP Status

The monitor with all basic features currently available is quite stable and well debugged. Additional debugging and "tuning" is being performed as increased activity turns up minor problems. Also, user feedback is beginning to be felt. For example, it was found that the original position of the 'purgeout' box caused some not-so-small annoyances. 'Purgeout' is in the same position as 'Return' in most other modes. Hence, force of habit led to an undesired menu (and file!) deletion. This is being changed, naturally.

Additional features are being planned and implemented. The basic capabilities of the system, while adequate for some activities, are hardly a complete or sufficient set. Increased documentation is also now in the pipeline.

Applications Support

Interactive development of the Simulation and Modeling System under GRASS has continued this quarter.

Extensive use has been made of the batch debugging program.

Use of the system for investigating methods for stiff equations has begun.

3.3.6 Remote Data Structure Utilities (R. Haskin, J. Nickolls)

Data Structure Communications Package

The communications and data structure handling program COMMUNE has been thoroughly tested this quarter, and is now being used by other programs in the system. The routines have been thoroughly documented, and a write-up is now available. Several important improvements and extensions have been made to the package, with care taken to preserve compatibility with earlier versions. These changes are:

CALLER: The FORTRAN link to the package has been completely rewritten. It now supports the FORTRAN I/O package (FLOCS# modification) and includes a much more efficient linkage to routines in COMMUNE.

LINE, LINED, TEXT: X and Y coordinates may now be specified either in screen increments (as before) or in floating point inches. CALL LINED (0., 1., 1) will now cause a one inch vertical line to be added to the line block. Also, an optional error return is provided for these routines in the event that the data structure block being built fills up. For example, the FORTRAN statement CALL LINE (10, 30, 1, & 400) will transfer control to statement 400 if the line block becomes full as a result of the call.

REPLY: An optional fourth parameter has been added in which, if present, will be returned the actual number of characters sent by the PDP-8 before being padded with blanks. The calling sequence for this option is:

CALL REPLY (ITYPE, LEN, ITEXT, NCHARS)

The old calling sequence can still be used if NCHARS is not desired.

MESSAGE: This routine sends a text line to the display. The calling sequence is:

CALL MESSAGE (ITEXT, LEN)

ITEXT is either an INTEGER*4 array containing the characters to be sent or is a character constant. LEN is the number of characters to be sent (≤ 72)

SEND: An option is available which will cause the line block to be sent to the display and re-initialized if it fills up before another SEND call. To use this option, the block must not be marked to replace the current PDP-8 data structure, and must be for display. To do this:

ICODE(2) = 0

ICODE(3) = 0

SEND(0, ICODE)

...

calls to LINE and LINED, possibly

filling up several line blocks

...

SEND(1, ICODE)

This option behaves as follows:

- (a) it remains in effect until the line block is sent explicitly (as by the second SEND call above),
- (b) if more than one block is sent, only the first will cause an erase or frame,
- (c) when the line block is finally sent explicitly, the ICODE parameters from the SEND(0, ICODE) are used (erase and frame behave according to (b)),
- (d) sending the line block explicitly clears the option.

Hardcopy Output

This quarter the routines to produce hardcopy CalComp plots from pictures drawn on a console were incorporated into a package called GPLØT.

The input to GPLØT consists of the name of a picture or mnemonic stored in the graphics filing system, and a string of option words specifying what parts of the picture are to be drawn. This input may be specified via a PLØT command from a console using the library service package, or from data cards. A PLØT command such as:

"PLOT_PICTURENAME/USERNAME.LIBRARYNAME_OPT1,OPT2,OPT3"

is stored as an entry in the file SYSTEM.PLOTLIB for later execution by PLOT, which plots all such entries and those from cards as a batch job.

Picture names entered on data cards must be fully qualified names containing username and library name, but these may be omitted when using the PLOT command. LSD automatically adds the logon username 'PICLIB' in this latter case if not already present in the command.

The options currently available are listed below. Each option word is five characters long and begins with a one letter prefix describing what it affects. Any option may be preceded by a negation symbol '-' to indicate the negative sense of the option. Starred options are defaulted to ON.

P-options concern the entire picture:

- *PILLIN - Draw local lines from line block
- *PCTXT - Print comment text from text block
- *PSUBP - Draw subpictures from mnemonic instance block
- *PTERM - Draw terminals and connections from terminal block
- PDECL - Print declarations
- PEQPM - Print equations and parameters
- *PFRAM - Draw frame as seen on console
- *PFTXT - Print system data in the frame

M-options concern mnemonic storage blocks:

- *MILLIN - Draw local lines of mnemonics
- MCTXT - Print comment text of mnemonics
- MTERM - Draw terminals (when only drawing the mnemonic)
- *MPARM - Print model parameter list beside mnemonic
- MTNM# - Print type name number on mnemonic terminals
- MTNML - Print mnemonic type name list in menu area

T-options concern terminal blocks:

TSEQ# - Print sequence number on terminals
TTNM# - Print type name number on terminals
TTNML - Print type name list in menu area
TDINV - Draw "invisible" connections between terminals

S-options concern subpicture instance blocks:

SPARM - Print parameter assignments of subpicture instances

The user may specify as many options on as many cards or console lines as needed by ending the line with a comma and continuing on the next one.

When PLOT is run from batch, it first plots all jobs specified in data cards, then starts reading jobs from SYSTEM.PLOTLIB. All routines accessing SYSTEM.PLOTLIB use the system macro ENQ to ensure that no plot jobs are lost. PLOT deletes all entries from SYSTEM.PLOTLIB when it is finished.

If desired, a mnemonic may be plotted by itself by using the six character mnemonic name followed by a period and the mnemonic number:

"PLOT_TRNSIS.1_MCTXT,MTERM,MTNML".

Graphics Library Maintenance Program

LSD has been expanded to allow common access of files among users. This has been done by expanding the command syntax to provide for the qualification of all file names (i.e. picture and mnemonic names) with the library from which they are to be taken. Also, library directed commands, such as CATALOG, allowed specification of the library to be used. The new syntax is:

Generalized file name--

$$\left\{ \begin{array}{l} \langle \text{picture name} \rangle \\ \langle \text{mnemonic name} \rangle . \langle \text{mnemonicid} \rangle \end{array} \right\} \left[/ [\langle \text{username} \rangle .] \langle \text{libraryname} \rangle \right]$$

Examples--

- A. PLOT RESIST
- B. PLOT RESIST.Ø
- C. PLOT RESIST.Ø/PICLIB
- D. PLOT RESIST.Ø/HASKIN.PICLIB
- E. COPY RESIST.1/HASKIN.PICLIB NEW.Ø/NICKØLLS.ALTLIB

Generalized library name--

[[<username>.] <libraryname>]

Examples--

- E. CATALOG
- F. CATALOG PICLIB
- G. CATALOG HASKIN.ERRLIB

The defaults for specifying a library name are as follows:

1. If the username is omitted, the name used is the one by which the user is logged onto the 360.
(i.e. examples C, F above).
2. If the entire library name is omitted, the username is chosen as above, and the default library name is PICLIB (i.e. examples A, B, and E above).

EQUATIONS:

$F = \min(V5, TIME \cdot V5)$

PARAMETERS:

T1

T2

V5

DEFINITIONS:

G

EQUATIONS:

$D = \sin(TIME/10)$

SAMPLE HARDCOPY PRINTS OF TWO MODELING SYSTEM NETWORKS

3.4 Computer Maintenance and Construction (C. Carter, I. Cunningham, C. Hyde, T. Kerkering, H. Lopeman, B. Miller, J. Nickolls)

3.4.1 Comutek Computer Graphics Terminal

STATUS:

The three University Purchase Orders (#215711, #228151, and #233131) are complete and the items are presently out of warranty.

The remaining items promised to us but not delivered are:

1. Modifications necessary on 611 storage scope.
2. Graphic alignment section.
3. Software levels 0, 1, 2.
4. Fix for cursor intensity during write-through.

SPARES:

Replacement spares have been checked out on the two Comutek terminals and a third terminal has been assembled incorporating the spare keyboard and storage tube. This third terminal is operational and a part of the PDP-8/I multiplexor graphic terminals.

3.4.2 Equipment Maintenance Log Summary

Comutek

1. Terminal #1 power supply +150 regulator blown. (Repaired)
2. Terminal multiplexor losing flags and going to initial state. (Logic error in multiplexor)

PDP-8/I

1. Erratic problem of occasional channel hang-up. (Replacing filter capacitors with larger ones on channel boards)
2. Indicator lamps out. (Replaced)

PDP-8

1. Core memory bank 2 failing constantly. (Bad memory address selector board replaced)
2. Switches failing on PDP-8. (Replaced with new switches of different manufacture)
3. Top decatape drive slow to respond occasionally. (No remedy as yet)

Disk

1. Track 26₈ failures. (Rewired track)
2. Track 46₈ failures. (Rewired track to spare)
3. Track 128₈ failure. (Rewired track to spare)

Inktronic Printer

1. Nozzle clogged in column 2. (Cleaned)
2. Paper drive motor clutch frozen, burned out motor drive board resistor. (Repaired clutch and board)

3.4.3 Air Conditioning (Room 31A)

There has been a rash of problems with the computer room air conditioning.

1. Control valve air compressors both failed causing heat and humidity build-up. (Thermal shutdown tripped)
2. Control stuck causing excessive heat in computer room. (Thermal shutdown tripped)
3. Condensate pan in air handling unit drain plugged up and spilled water through the ceiling into the computer room (twice). (Minor damage to some tools involved)

During this quarter, a thermal overload protector was installed to dump all power in the computer room if the temperature reached 85°F. It appears to work very well.

3.4.4 Stereomatrix Interface (I. Cunningham)

Construction suggested in the previous quarter has been brought close to completion. Installation and checkout is beginning. Specific work done:

Cards

E6/D6	A/D Data Buffer
E7/D7	DMA Control--Interrupt Flags
#8/D8	Direction Decode
E9/F9	Sequencer
E10/F10	Up Down Counter
E11/F11	Up Down Counter
E12/F12	Up Down Counter
F16	Line Receiver
F15	Line Drivers
F14	Line Drivers
F13	Line Drivers
C19/D19	IOT Decoder

Cables

In addition, six cables were made and checked out. Main frame wiring completed from the IOT decoder to the pallet cards connector block.

Five twisted pair cables and three coaxial cables have been strung from Room 31E for about 150 feet to Room 28 in the basement where information will be transmitted and received.

Mainframe

A rack containing the A/D converters and power supply was wired. A DEC rack was wired which contains all of the above mentioned cards except for the IOT decoder.

Construction of the interface between the PDP-8/I and the stereomatrix display has been completed and approximately half of it has been checked out. An additional power supply was added to the PDP-8/I to handle the extra logic and one bad card was found in the DMØ1 direct memory access multiplexor. The cables to the display have been installed but are not connected.

The total interface can be checked out without the stereomatrix display and this should be completed shortly.

Stereomatrix Interface Instructions

6511: load X beam coordinate position from AC(0-9)
(10 bit sign and magnitude)

6516: read X cursor coordinate position into AC(0-9)
(10 bit 2's complement)

652X: for Y positioning

653X: for Z positioning

6541: skip on line done flag

6542: stop display and clear line done flag

6544: load the 12 low order bits of the line
address and start the display

6551: skip on OR of CURSOR, INCIDENT, and FUNCTION KEY-
BOARD FLAGS

6552: or CURSOR, INCIDENT, and FUNCTION KEYBOARD FLAGS
into AC bits 0, 1, 2, respectively

6554: load extended address from AC bits 9, 10, 11 and
clear CURSOR, INCIDENT and FUNCTION KEYBOARD FLAGS
from AC bits 0, 1, 2, respectively, if that bit is
a one

6561: or FUNCTION BOX bits into AC and clear FUNCTION BOX

6562: or current line address into AC

6564: resume drawing line at current point after
incident interrupt.

Line Elements

The format for line elements is as follows

0	1	5	6	11
I	DIRECTION	LENGTH		

WORD IN MEMORY

Each element can draw one straight line. If I is a 1 the line is intensified. Bits 1 through 5 permit the selection of 1 of 27 directions. The direction is calculated as follows:

$$\text{DIRECTION} = (\text{ZD} * 3 + \text{YD}) * 3 + \text{XD}$$

where XD, YD, and ZD are the values of X, Y, and Z unit vectors (0 is no component, 1 is a unit vector in the positive direction, 2 is a unit vector in the negative direction).

LENGTH is an unsigned integer with a maximum value of 63. If the DIRECTION is 36 octal the LENGTH is interpreted as status. Bits 10 and 11 specify intensity; bits 7 and 6, scale; and bits 8 and 9 are free. A value of 37 octal for the DIRECTION cause the display to halt and the line done flag to be set.

A picture is generated by first loading the X, Y, and Z registers to position the beam. The address of a sequence of line elements is loaded. The interface generates the points for the display until a HALT is reached.

The following program draws a cube.

```
10F
LOOP      CLA
          6554      /LOAD EXT ADDRESS
          TAD      XCOORD
          6511      /LOAD X-CORD
          CLA
          TAD      YCOORD
          6521      /LOAD Y-COORD
          CLA
          TAD      ZCOORD
          6531      /LOAD Z-COORD
          :
          :
```

```

:
CIA
TAD          LINADR
6544         /LOAD LINE ADRS AND START
6541         /SKIP ON FLAG
JMP          .-1
6542         /CLEAR FLAG
JMP          LOOP /DRAW AGAIN
XCOORD       4200         /-40 OCTAL
YCOORD       0200         /+40
ZCOORD       4200         /-40
LINADR       36          /SET SCALE AND INTENSITY
4677
4177
4377
5177
4677
4277
4377
5277
4177
0677
5177
0377
4277
0677
5277
3700         /HALT

```


U.S. ATOMIC ENERGY COMMISSION
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR
DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO.

COO-1469-0193

2. TITLE

2nd QUARTERLY PROGRESS REPORT 1971 (April, May, June)

3. TYPE OF DOCUMENT (Check one):

- ☒ a. Scientific and technical report
☐ b. Conference paper not to be published in a journal:
 Title of conference _____
 Date of conference _____
 Exact location of conference _____
 Sponsoring organization _____
☐ c. Other (Specify) _____

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

- ☒ a. AEC's normal announcement and distribution procedures may be followed.
☐ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
☐ c. Make no announcement or distribution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

C. W. Gear, Professor
and Principal Investigator

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature

Charles W. Gear

Date June 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

- ☐ a. AEC patent clearance has been granted by responsible AEC patent group.
☐ b. Report has been sent to responsible AEC patent group for clearance.
☐ c. Patent clearance not required.

4. IMAGE PROCESSING AND PATTERN RECOGNITION RESEARCH: ILLIAC III
(Supported in part by Contract AT(11-1)-2118 with the U.S. Atomic
Energy Commission)

A balanced program of experiments with an image processing computer has been carried out. Highlights of the program include:

- 1) Extensions of the Show-and-Tell package for interactive picture processing.
- 2) Development of the theoretical basis (signal detection theory, covering techniques, etc.) for plane parallel image processing - procedures critical for rapid preprocessing and local feature extraction.
- 3) Strategies for structural inference using multi-graph scene representation; undoubtedly techniques here will have utility far beyond their immediate application to scene analysis.
- 4) Applications have been initiated in diverse areas: to materials science, bio-medicine, robotics and remote sensing of the environment.
- 5) Finally necessary completions and development of the Illiac III Image Processing Computer integral to the above research are recommended.

Our goal here is to design a sight sensory system predicated upon parallel strategies for image processing. This development must of course evolve from our experience with the Illiac III system. The key property we seek is the ability of the system to learn rapidly from instructional interaction with professional personnel (not necessarily trained in computer technology). The validity of this orientation has been, we believe, vindicated by the rapid growth in significant image processing applications we can now attack at the end of the first year of this contract.

4.1 INTERACTIVE PICTURE PROCESSING

One objective of our research has been to develop a strategy which would permit personnel (not trained in computer technology) to input scanning instructions directly to the image processing system. An interim stage in this development has been the design and implementation of an interactive system for picture processing: The Show-and-Tell system described below.

4.1.1 Show-and-Tell

The Show-and-Tell programming system is a console-oriented software package providing a facile method of using the operational components of the Illiac III computer. The system presently operates on the hardware depicted in figure 1. Overall documentation of the Show-and-Tell system is contained in reference [1].

Show-and-Tell is designed to support local image processing and image acquisition directly using the Illiac III PAU and S-M-V Systems, and also to support experimentation in image processing theory using the high-level language and mass storage of the IBM 360/75. To this end, a bidirectional link with the IBM 360 is provided allowing (1) calling IBM 360 programs and subroutines from the PDP8/e console teletype and (2) transmission of data and pictures back and forth between Illiac III and the IBM 360 via a dedicated PDP8/e interface.

FIGURE 1. HARDWARE CONFIGURATION

4.1.2 Intermachine Link

The intermachine link provides a means for programmers to develop image processing codes using simulators for various sections of Illiac III (e.g. PAX II) until these sections are checked out. In this regard a Digital Equipment Corporation PDP8/e was ordered in January 1971. The machine will initially be used to communicate with the Department of Computer Science's 360/75. It will eventually assume its major functions as maintenance processor and data concentrator for Illiac III as the system evolves.

The PDP8/e was installed and accepted on May 26, 1971. The current programmed I/O interface to the core and scanner was transferred from the PDP8 to the PDP8/e on June 5.

The interface to the 360/75 via the 2701 PDA was designed, wired and checkout started on June 6.

The maintenance processor/Exchange Net interface is a flexible, high-speed interface to be used in debugging the various units of Illiac III. The interface is about 80% designed. The maintenance processor/TP interface has been designed and is being implemented.

The real-time clock, a crystal-controlled clock with program selectable time base, was designed, wired and checkout started. This clock is described in Department of Computer Science File No. 863.

The bus drivers and device selectors, to facilitate interfacing programmed I/O devices, were installed and checked out. These devices are described in Department of Computer Science File No. 862.

4.1.3 Scan-Display Devices

The large format television system was assembled and initially adjusted with good results. The unit consists of a modified graphic arts camera bed on which is mounted a television camera, 2 KW of quarts front lighting, and variable cold cathode back lighting. Both the camera and copy board are movable so that magnification can be changed. Maximum back lighted copy size is about 14 x 17 inches (x-ray film). Front lighted copy can be slightly larger. In terms of size as reproduced at the monitor screen, the magnification range available is of the order 2X to 0.8X (continuous). Lens filters will be used for color investigations.

The remaining television camera was mounted on the automatic microscope unit to be used for remote program controlled operation. All stepping motors for the stage were installed and test run under manual control. A repeatability check showed that the X and Y axis motors relocate the image $\pm .0002$ inches reliably. A 200 step per turn motor will replace the current 24 step per turn unit for Z axis (focus) control.

PDP/8 machine language and FORTRAN programs are currently being written to calculate the Scanner/Monitor performance parameters.

4.2 PARALLEL PROCESSING OF PICTURES

4.2.1 Pattern Articulation Unit

The Pattern Articulation Unit (PAU) is a unique parallel processor used for the analysis of digitized images in planar form (figure 2). It is our intention here to devise, implement and evaluate parallel processing strategies appropriate to the Pattern Articulation Unit of the Illiac III Computer.

The PAU Display System which is capable of displaying the states of the 1024 stalactites in the Iterative Array, was completely debugged and is now operational.

One-half of the Iterative Array (512 stalactites) is now operational. Wiring of the stalactites of the second half have also been completed but not checked out (figure 3).

Two-thirds of the PAU control is now completed. In particular, the GATEIA instruction has been implemented which allows us to simulate by microprogramming the operation of any other instruction.

PAX II Language Support: The first issue of the "PAX Users Group Newsletter" has been sent to all installations known to have a version of the PAX II Subroutine Package. Requests for copies of this newsletter should be sent to PAX Newsletter Editor, Illiac III Project, Department of Computer Science, University of Illinois, Urbana, Illinois, 61801.

4.2.2 Signal Detection Theory

The central recognition task of image processing is the segmentation of pictures by the detection of boundaries and textural differences. Plane parallel picture processing rests or falls on the success of such methods. Accordingly, we place great stress upon the fact that the design of optimal image filters should rest upon a sound methodological basis (e.g. signal detection theory) and move away from entirely ad hoc procedures. Parallel processing algorithms for pictures should admit a technical basis comparable to the classical design principles for linear filter networks.

P3 DISPLAY

Fig. 2 PATTERN ARTICULATION UNIT

4.2.2.1 Extensions of Signal Detection Theory

Regions of different textures can be differentiated from each other if it is possible to extract the patterns which characterize these textures. Ideally each texture consists of only one single pattern, known as the 'texture element' (unit cell). But extracting the 'texture element' (whose shape and size is generally unknown) in a natural texture is a formidable task. Instead we can define a complete set of patterns and search for the occurrence of these patterns in the given textures.

4.2.2.2 Pattern Elements

We define a template centered around a point which extracts the gray levels of its neighboring points. All possible patterns define the universe of texture elements. A table showing the template and the number of elements in the universe is shown below.

Template		Number of gray levels in the picture			
Regular Mode	Hexagonal Mode	NGL = 2	NGL = 3	NGL = 4	NGL = 16
	2	3	4	16
	4	9	16	256
8	27	64	4,096
16	81	256	65,536
32	243	1,024	1,048,576
64	729	4,096	17M
	128	2,187	16,384	268M
512	19,683	262,144	68,600M

4.2.2.3 Development of the ROC Curve

Considering the n-tuple of gray levels from each sampling template as an 'event', a ROC (Receiver Operating Characteristic) can be built up to discriminate between a pair of textures, say T^1 and T^0 . Let E^1 and E^0 be the set of events obtained from scanning T^1 and T^0 with a given template respectively. Using this information, the universe of events (E_U) is divided into 4 disjoint classes, F^1 , $F^\emptyset$, F^0 and F^* . F^1 and F^0 are the events which occurred exclusively

in T^1 and T^0 and F^ϕ consists of events which occurred in both. F^* contains events which occurred in none of these scenes. Defining the 'likelihood ratio' of an event as

$$L.R. = \gamma \frac{\# \text{ of its occurrences in } T^1}{\# \text{ of its occurrences in } T^0}, \text{ where } \gamma \text{ is a}$$

normalization factor to compensate for intrinsic probabilities, the L.R. of each event in F^ϕ is calculated and these are ordered according to L.R. with the event with the largest L.R. as the first element.

Three sets of events F^1 , F^ϕ (ordered) and F^0 can be represented graphically as shown. The graph is called the R.O.C. Curve [2]. Selecting a 'suitable' break point λ , F^ϕ is divided into two classes $F^{1\lambda}$ and $F^{0\lambda}$.

Now consider the set union $F^1 \cup F^{1\lambda}$. This is a set of events whose occurrence is more probable in T^1 than in T^0 . Similarly with $F^{0\lambda} \cup F^0$.

After obtaining these two sets, we return to T^1 and T^0 and mark differentially the events which are members of the sets $F^1 \cup F^{1\lambda}$ and $F^{0\lambda} \cup F^0$ respectively. We obtain in this manner two 'colored' pictures which are more easily discriminated than the original pictures. If we repeat this process, we commonly obtain yet better results and often find that T^1 will be colored completely with one color and T^0 , another.

4.2.2.4 Detecting Textural Boundaries

Given the proto-samples of a texture present in a combined scene, borders between these textures can be detected using the ROC technique. Here we use the combined picture as T^1 and the collection of isolated proto-samples

as T^0 . Here the equivalence class F^1 which contains the events which occurred in T^1 alone, will predominantly contain events which occur on the border. If we go back and detect all the events in F^1 , we normally are able to extract the border. This experiment has been conducted using a 2 x 2 window on artificially created textures and natural textures, i.e. 'grass' and 'stone' from Broadatz's book of textures. Results are illustrated below.

Example 1: Artificial Texture

A deformation in a regular design is detected.

Template defining the event =

Gray levels = 4

Total number of events in the universe = 256

Typical Distribution of Events:	Set	# of elements
	F^1	3
	$F^\emptyset$	5
	F^0	0
	F^*	248

Two iterations of the filtering process are applied to give the boundary identification shown in figure 4.

Example 2: Grass and Stone

Stone boundaries are stripped from a composite picture containing both textures (see figure 5).

T^1 : Combined picture

T^0 : A window from 'grass' and a window from 'stone'.

Template defining the event =

Gray levels = 4

Total number of events in the universe = 1024

Typical Distribution of Events:	Set	# of elements
	F^1	93
	$F^\emptyset$	138
	F^0	36
	F^*	757

INPUT PICTURE

FILTERED OUTPUT
(X = BOUNDARY)

AS ABOVE WITH NOISE

FILTERED OUTPUT

Fig. 4 SIGNAL DETECTION: Example 1

4.2.3 Interval Coverings

4.2.3.1 Orientation

In the course of a series of seminars this year (Sight Sensory Systems Seminars, see bibliography) Michalski and McCormick have developed the basis of a new pattern classification procedure. The technique allows one to find an optimum set of k-dimensional "intervals" to package an observed set of k-tuple patterns. The procedures are entirely consistent with the foundations of signal detection theory, and in the limit of adequate statistics gives a class of optimal recognizers corresponding to points on the ROC (receiver-operating characteristic) curve. (See figure 6)

This work, in one sense, generalizes classical switching theory concepts of a map covering. This correspondence is mutually beneficial, for the combinatorial algorithms of finding an optimum covering can then be adequately resolved using algorithms of known effectiveness from switching theory. Attention in our work to date has centered upon the very effective "method of disjoint stars" developed by Professor Michalski.

4.2.3.2 Formal Description

A report describing the theory has been prepared [3].

The paper considers:

- (1) a Boolean algebra $\langle 2^E, \cup, \cap, -, E, \emptyset \rangle$ of event set E_i from a discrete finite vector space E , and
- (2) mappings f from the set E into $\{[0,1], *\}$, where $*$ represents some unspecified value. A special case of the above is the Boolean algebra and Boolean functions considered in switching theory, where E is a space of binary vectors and f maps E into $\{0,1,*\}$, i.e. into the endpoints of the interval $[0,1]$ and $*$.

A meet semi-lattice of multidimensional intervals (interval complexes) in E is introduced and then the concepts of exact, free, unordered and ordered interval covers of f are defined.

The simplest case of a cover - an unordered exact cover of a set $F^{1\lambda}$ against $F^{0\lambda}$ - is defined as a set of interval complexes whose set-theoretic union covers a given subset $F^{1\lambda} \subseteq E$ (defined as $\{e \mid f(e) > \lambda\}$) and does not cover any element of another given subset $F^{0\lambda} \subseteq E \setminus F^{1\lambda}$.

(See figure 7.)

FIG. 6 RECEIVER - OPERATING - CHARACTERISTIC (ROC)

$M^q(f/\lambda)$:

$\Delta = 0$

$t = 5.5 \text{ sec.}$

*	/	*
*	*	*
*	*	*

n - number of variables
(in our case Number of elements
of the square)

h - number of grey levels

* - denotes that the corresponding
to square-element variable is not
relevant.

FIG. 7 DISCRIMINATION BETWEEN TWO SETS OF PATTERNS

(U-LIKE PATTERNS $F^{1\lambda}$ ARE DISTINGUISHED FROM O-LIKE PATTERNS $F^{0\lambda}$ BY INTERVAL COVERING WHICH EMPHASIZES THAT ONLY ONE CELL IS SIGNIFICANT - SEE $M^q(f/\lambda)$. THIS COVERING WAS FOUND BY OPERATIONAL INTERVAL COVERING PROGRAM

The concept of ordered covers was developed to accomodate a preferential order in covering the set of 'mixed' events, defined as $\{ e \mid 0 < f(e) < 1 \}$ (a case not considered in classical switching theory). (See figures 8, 9, 10).

The synthesis algorithm of covers is based on the 'method of disjoint stars', which has proved to be very useful for synthesis of complex switching systems. Quasi-minimal covers, produced by this method, are either minimal or approximately minimal. However, when we cannot state that the obtained solution is minimal, an estimate of its maximal possible distance to the minimum is provided.

Applications of the interval coverings concepts to pattern recognition and picture filtering are delineated.

4.2.3.3 Remarks on Applications

A cover of a mapping f consisting of multidimensional intervals can be interpreted as a set of 'filters' for recognizing events from a signal class (represented by $F^{1\lambda}$) for pattern recognition and picture processing purposes, e.g. for discriminating regions of different textures, striping of background, local feature extraction, border detection, etc.

In case of an ordered cover the individual filters correspond to the consecutive points on the optimum receiver-operating-characteristic (ROC), defined as in statistical decision theory. The above ROC curve can be obtained by optimal ordering events from the two classes to be distinguished with regard to the likelihood ratio of their frequency occurrence.

Fig. 9 ORDERED INTERVAL COVERING OF IMAGE $T(f)$
 (Events numbered 1 are covered first, then 2, etc. Compare with Fig. 5)

Fig. 10 COVERING INTERVALS REFERRED TO THE EVENT SETS OF THE ROC CURVE (SEE FIG. 4)

4.3 STRUCTURAL INFERENCE

Our long range goal can be viewed as this: to design a sight sensory system which can infer the syntactic and semantic structure of its visual environment from selected instances of objects and scenes from the environment. Implicit here are parallel processes for scene segmentation based on the signal detection theory and appropriate generalizations of covering theory. Through dialog with the system, a user may participate in the acquisition of the structural description of the visual environment.

4.3.1 Scene Segmentation

The picture processing model attempts to develop the structure and semantics necessary to treat a general two-dimensional scene. To achieve this goal, we introduce the first characteristic of the model. The picture is initially mapped into a cellular mosaic: a segmentation of the scene into cells (e.g. regions bounded by simple curves) which then form primitives for further analysis. (See figure 11 A & B) The choice of primitives strongly influences the development and the capability of the model. Cells seem to be far superior to primitives consisting primarily of line-elements, which were used in earlier approaches. Line-elements were often chosen to attempt to obtain a model based on one-dimensional and context-free grammars, which are inadequate in many cases.

Scenes are represented internally in the system as a hierarchical structure of cellular mosaics. The cells, or regions, of the mosaics, considered as nodes with assigned attributes and names, are linked (according to selected binary relations) to form a labelled digraph representation of the scene. The decision rules for when to select a given binary relation or to classify two regions as having analogous attribute values are inferred by the system from instances of the visual scene in the instructional period.

The graph representation is chosen for global level picture processing. The recognition process can then be viewed as replacement rules operating on graphs. This process, which is analogous to parsing, is called "composite formation." A replacement rule forms a composite element from subelements of the graph. The new graph formed by new composite elements and new relations involving these elements is then

Fig. 11A PARTIAL MOSAIC REPRESENTATION.

_____ border adjacency relation
 _____ vertex adjacency relation
 Region Complexes (dashed lines)

Fig. 11B COVER FOR MOSAIC

the candidate for further composite formation. For successful recognition, this process should tend toward a "recognizable graph," (see figure 12) i.e. a graph which has somehow been specified as an acceptable goal.

The key property of the system is its ability to rapidly infer which factors of the binary relations and which multidimensional intervals of attribute values best articulate the scene for a given mode of system behavior and visual environment.

Toward the end of the current contract year, a thesis by John C. Schwebel will summarize the larger part of our work to date in this area. In addition, Michalski and McCormick are preparing a paper for the Paris meeting of IFAC Conference.

4.3.2 Structure Operation Language Support

We have specified a Structure Operation Language , SOL [4] in order to be able to simply and conveniently express operations on multi-graph structures which allow an arbitrary number of variables to be associated with any network element.

To justify the definition of yet another "programming language" and put its development into perspective, we cite the following facts: We are currently able to program picture processing algorithms which operate on the PAX plane representation of pictures as binary valued elements with neighborhood connectivity relationships. The next and most natural abstraction from a PAX plane representation is a graph-structure, (i.e. a hierarchical system of graphs). Many scene segmentation algorithms can be expressed most simply by operations on a graph structure. We have, and are continuing to develop, theory for the next higher level of picture processing operations represented as structure transformations.

Thus, the structure operation language is quite helpful for precisely specifying and for experimenting with heuristic scene segmentation strategies.

The language has a small number of root operations which are sufficient for expressing network transformations and has been specified with the goal of embedding it in PL/1. Figure 13 lists and illustrates the root operations of the language.

Fig. 12 COMPOSITE FORMATION

a) subgraph operations

LINK	subgraph
MOVE	pointer
ADD	element
DELETE	element

SELECT	element
DECLARE	element
ASSOCIATE	variable
FREE	variable

b) other operation types and operands

Fig. 13 SOL OPERATIONS

The language is still in the early stages of implementation.

A syntactic analyzer for the language is under development using the Translator Writing System of Alan J. Beals, Department of Computer Science, University of Illinois.

4.4 APPLICATIONS

The operational tools, hardware and software to begin a significant attack on several applications were begun. Automated cytology was emphasized this past year, as cells with their relatively simple morphology provide a ready short range target for the parallel image processing strategies of the Illiac III computer system.

4.4.1 SEM Micrographs

The Scanning Electron Microscope has been extensively employed in its applications to material science as well as subcellular biology. A JSM-3 SEM, Japan Electron Optics Laboratory Company was installed in the Material Research Laboratory, University of Illinois, Urbana, last summer. Mr. John B. Woodhouse of MRL is in charge of the facility. With his help, a program to explore the benefits of image processing automation for SEM imagery was initiated. The system design of an automated on-line scanning electron microscope has been in process.

One of our main goals is to study cells and subcellular populations as generated by the cytospectrometer. We have undertaken two preliminary tasks:

- (1) Slide preparation: With the services of Mrs. Beth Lepinski and Mrs. Grace Conway of Mercy Hospital, Urbana, Illinois, we have prepared a variety of samples such as red cells, leucocytes, urine crystals, bacteria, chromosomes, etc.
- (2) Evaporated coating techniques: The problem arises from preparation of a slide to be available to be observed both under conventional optical microscope and SEM. With the help and suggestions of Professor Francis Young of Civil Engineering Department, University of Illinois, Urbana, a thin coating of 200 Å of carbon and Au/Pt alloy has been tried as a compromise to solve this problem.

In addition a special specimen carrier for integrated circuits has been constructed, so that the circuit can be operated within the vacuum chamber of the SEM. Checkout of LSI circuitry of the future will likely entail the use of an electron beam probe (i.e. the scanning electron microscope) quite as today the technician uses a manually positioned contact probe. Our studies here are still preliminary.

4.4.2 Cervical Smears

A central problem in the construction of machines to automatically inspect and classify samples of cervical cells (e.g. Pap smears) has been the tendency of the cells to clump, fold and overlap in such a way as to confound the analysis algorithms proposed. This problem occurs whenever morphologically-based measurement is attempted on cells of this type, even after they have been suspended in a fluid medium and mechanically rearranged with flow or droplet techniques.

While non-morphological measurements such as total ultraviolet absorption or total light scatter* may prove to be sufficient for the cervical cell screening problem, it is felt that the ability to reliably locate and identify various cell types and substructures in the face of folding and clumping provides a firmer base for development of new techniques for automated cytology.

It was decided here to proceed by attempting to develop techniques to segment microscope slides of cervical cells prepared according to the standard method of Papanicolaou. Work on this problem has indicated the feasibility of using the Illiac III system organization to do digital processing of Pap smears without disrupting the form of the sample currently used by pathologists, an approach which reduces many difficulties in correlation of results and establishing follow-up of positives which arise when the cells are transported in other ways. In addition it is anticipated that feature extraction and pattern articulation procedures developed for this application will transfer easily to other cytology applications.

The picture processing strategy adopted here has been to approach the image in a top-down fashion because of the fairly limited number of different forms to be located, and because of the simple structure of the forms, which consist basically of blobs (dark, compact, symmetric regions) and conglomeration of blobs of various sizes and textures. For example,

* Kamentsky, L. A., and Melamed, Instrumentation for Automated Examination of Cellular Specimens, Proceedings of IEEE, Vol. 57, No. 11 pp 2007-2016.

large blobs with a coarsely grainy texture have a high probability of being malignant cell nuclei. Some preliminary results are shown in figures 14 through 17E. Figure 14 shows a photograph of a field of a Pap smear [60x]. Figure 15 shows the 128 x 128 element digitization of the field. (The 46 mm film scanner is being used temporarily while the microscope scanner is under checkout.) A filter was applied to this picture which detects blobs of a size just greater than a white cell (small dark spots in figure 16) while rejecting blobs composed of white-cell-sized subblobs. The output of this filter, shown in figure 16 is thresholded and locations of high filter output (figure 17A) are rescanned at higher resolution for texture analysis. (See figures 17B,C,D,E.)

The above investigations, the subject of a Master's thesis, are only preliminary and will be continued during the forthcoming contract period.

Fig. 14 PAP SMEAR FIELD

Fig. 15 128 x 128 DIGITIZATION
OF FIG. 14

Fig. 16 BLOB FILTER OUTPUT

Fig. 17A BLOB HITS CIRCLED

Fig. 17B SECOND RESCAN AREA

Fig. 17C SECOND RESCAN

Fig. 17D THIRD RESCAN AREA

Fig. 17E THIRD RESCAN

4.4.3 Brain Mapping

The purpose of this investigation is to develop structural inference techniques and to apply and test them in the environment of neuroanatomy of the brain.

It is intended that these techniques be applied first to a small sample populations of objects in the brain. In connection with current research in quantitative neuroanatomy*, methods are being developed for the automatic recognition of cellular and subcellular constituents rendered visible in cresylecht violet stain and Weil stain under a light microscope. An immediate application is to determine quantitative properties such as the number of neurons in specific domains of the brain and the geometrical dimensions of their nuclei and nucleoli. (See figure 18.) A further application is the delineation of the boundaries of larger domains in the brain, e.g. nuclei, fiber tracts, such that finally a brain map is obtained.

Our approach consists of a preprocessing and an inference phase. In the preprocessing phase, the picture is scanned (i.e. transformed into a grid of points differing in gray scale) and partitioned into closed regions. Two methods to obtain regions are presently being investigated. A region is determined as a connected set of grid points such that (1) they have almost uniform distribution of gray scale values or (2) they form a more complicated characteristic texture. After partitioning a picture into regions, we associate with each region applicable attributes and their values as well as binary relations (containment, adjacency, etc.) valid between pairs of regions.

A simple context-free grammar has been developed whose nonterminal values are class names used in neuroanatomy to designate objects such as cell-types, constituents of cells, etc. The variables are associated with attributes which in turn are connected by semantic rules. Using clustering and covering techniques that are already available or being developed, a variable of grammar is assigned to each region. The structure of the picture is then described by a derivation tree whose nodes are associated with each identified region.

Fry, William J., Quantitative Delineation of the Efferent Anatomy of the Medial Mammillary Nucleus of the Cat, J. Comp. Neur. V. 139, No. 3, July 1970, pp 321-336.

Fig. 18 FILTERED IMAGES OF BRAIN NUCLEI
(Codes to extract parameters
such as maximum diameter,
perimeter, etc. have been written)

4. 4.4 Cytospectrometer

A new and fundamental technology for the identification and sorting of cells and subcellular components is suggested. A principal ingredient of this technology is the transport of charged particles (cells, subcellular components or macromolecules) by an iterative system of electrostatic quadrupole lenses. These techniques, though extensively used in accelerator installations for the transport of nuclear and high energy particle beams, have not had comparable application to the guidance, identification and separation of macromolecular and cellular particle beams. It is the intention of this research to explore this potential and to devise a first working instrument (called here a "cytospectrometer") which can bridge the gap between these digitally-oriented beam techniques and more conventional methods in cytology emphasizing light and electron microscopy.

4. 4.4.1 Objective of a Cytospectrometer

To provide a two-dimensional spatial separation of constituent particles (cells, subcellular components or macromolecules) on the basis of a 2-parameter differentiation of the incident stream of isolated particles. Parameters available for selection include mass, charge, shape, differential spectral absorption/scattering, rate of discharge in an ionizing atmosphere, etc. The procedures used ideally should maintain the identity of subpopulations, such as all chromosomes from a given nucleus.

Applications of the cell beam technology that warrant investigation include:

- i) Applications to hematology (differential counts, etc.)
- ii) Pap smear classification
- iii) Separation and classification of bacteria
- iv) Isolation of genetic mutants
- v) Subcellular particle separation: e.g. cell nuclei differentiation
- vi) Karyotyping of chromosomes
- vii) Cancer cell discrimination, including possibly stem cell identification

These potential applications cannot be adequately investigated, obviously, with the level of support requested here, but we should be able to evaluate the feasibility of applying the beam transport technology to these areas.

Constituent subassemblies of a cytospectrometer facility are shown in figure 19. A digitally controlled hypodermic syringe is used to inject a liquid jet of particles suspended in a solute carrier (figure 20).

It will be critical to the success of the spectrometer system that the features extracted from an individual cell passing through the beam transport system reflect the internal structure of the cell and/or subcellular particle. For example charge or mass gives an obvious discrimination by size (and in particular, specific charge alone almost allows a complete differentiation of white blood cells). These parameters however, do not afford the same sensitivity as histochemical reactions monitored by optical transmission. In particular we view the droplet stream of cells as a potential candidate for ultra-rapid low resolution (certainly not in excess of 32×32) image processing. The requirements here, $\sim 10 - 100 \mu$ sec. per frame, transcend all known systems, but would appear technically attainable. We anticipate that the design of this on-line system will evolve from our experience in processing cells off-line with the Illiac III Computer System.

DETECTION

CAN OCCUR
ANYWHERE BETWEEN
C AND J

TARGETING

CAN OCCUR
ANYWHERE BETWEEN
D AND J

INTERMEDIATE DEFLECTION NOT SHOWN

METHODS: CHARGE
MASS
TIME-OF-FLIGHT/POSITION
LIGHT, TRANSMISSION/REFLECTION
FLUORESCENCE/TRACER
ELECTRON SCATTERING
IONIZATION

Fig. 19 PROTOTYPE CYTOSPECTROMETER
BLOCK DIAGRAM

Fig. 20 CYTOSPECTROMETER DROPLET STREAMS

4.5 COMPUTER SYSTEMS SUPPORT

Basic support and graded evolution of the Illiac III Pattern Recognition Computer has been carried out under the present contract. This support has been central to the successful execution of the above core research program.

4.5.1 System Documentation

An extensive effort has been made to document the Illiac III Computer System. Key to this documentation has been the final issuance of the first two volumes of the Illiac III Reference Manual [5], [6]. The first of these includes a complete bibliography of available hardware documentation. Volumes III and IV have been prepared and will be issued this summer.

In engineering drawings, over 400 logical design sheets were prepared this past contract period in **final** form and signed off.

4.5.2 Design and Fabrication

The level of design work completed by the miniscule engineering (and maintenance) staff this past year is outstanding: 461 integrated circuit cards of control logic were laid out, and predominantly wired (under subcontract). A breakdown by area of the machine follows:

CONTROL CARDS OCTOBER 1 - JUNE 1

UNIT	# OF CARDS WIRED	# OF CARDS DESIGNED (NOT WIRED)	TOTAL
TP	73	54	127
PAU	73	2	75
SMV	81	31	112
AU	0	42	42
IU	0	30	30
SEQ. TEST.	25	7	32
PDP8/e	9	18	27
CIU	16	0	16
TOTAL	277	184	461

4.5.3 Control Point Strategy and Its Automated Diagnosis

The design technique known as Control Point Strategy has been used in the design of the Illiac III Pattern Recognition Computer. The control of this machine is functionally partitioned and physically divided into small, etched cards of integrated circuits. The testing of these cards is the immediate object of this research. Broader implications of this work are also discussed below.

4.5.3.1 Introduction

A control can be considered as being basically synchronous, asynchronous, pseudo-asynchronous or micro-programmed and the design techniques used are a function of these classifications.

The Department of Computer Science at the University of Illinois has been, historically, for the most part involved in the design and construction of asynchronous computers. The fact that this department has been primarily interested in parallel processor computers is strongly related to that involvement. An asynchronous control possesses some advantages over synchronous controls due to the relative independence of the speed of the different units. However, an asynchronous control "does require hardware which greatly increases the design complexity and renders a logical design problem of flow chart realization." Because of this complexity, this design has remained mostly "an art and not a science." (Swartout, 1963).

Some work [7], [8], [9], toward a formalization of this problem has previously been done and methods have been proposed to standardize the design. The solution proposed in [10] introduces a building block called a control point.

Work of the past year [10] has served to formalize and enlarge the control point concept. We introduce there a model called a control point net which yields a unique topological representation of a given pseudo-asynchronous system from which the control point design is easily derived.

4.5.3.2 Developments of the Past Year

A formal model of the pseudo-asynchronous control strategy has been developed [10]. Here a detailed analysis of the concepts of task, event, and transition in pseudo-asynchronous systems is given. A close study of what are called "external conditions" shows that transitions exhibit a very definite structure with respect to these conditions. This property is used to define a basic building block, called a control point, which logically implements the fundamental functions assumed by a transition.

The thesis of Rey [10] provides a brief review of how the control point strategy is actually realized in terms of logic circuits in the Illiac III Control. Given a control point design, a method based on the above formal model is then proposed for generating appropriate test patterns. The essential idea here is to partition the control point design into "independent subnetworks", to find tests for these subnetworks and to concatenate these tests according to a set of rules which take into account the connectivity properties of the Control Point Network. A first algorithm has been developed which realizes this concatenation.

The problem of physically applying the tests to individual cards was then attacked. The maintenance facilities provided by the Illiac III implementation of the control points, although simple, appear to be extremely time-saving for the debugging process. Since the actual test in our case is to be monitored by a mini-computer (PDP8/e), the design of a special interface, called the Sequence Tester, is needed. The Sequence Tester has two functions: (1) it submits input patterns to the card and (2) preprocesses the card response before transmitting it to the PDP8/e. The processing function facilitates the fault location.

4.5.3.3 Current Status

A detailed logical design of the Sequence Tester logic is essentially complete and fabrication has begun. The interface to the mini-computer (PDP8/e) should be largely complete before the end of the current contract year.

Test generation procedures have been implemented in a first FORTRAN program [11].

REFERENCES

- [1] Read, John, "Show-and-Tell System Specifications," Department of Computer Science Report No. 429, University of Illinois, March 1971.
- [2] Green, D. M., Swets, J. A., "Signal Detection Theory and Psychophysics," John Wiley & Sons, Inc., New York, 1966.
- [3] Michalski, R. D., McCormick, B. H., "Interval Generalization of Switching Theory," Department of Computer Science Report No. 442, University of Illinois, May 3, 1971 (In printing).
- [4] Schwebel, John C., "Picture Segmentation by Structural Inference," Thesis in preparation.
- [5] "ILLIAC III REFERENCE MANUAL, VOL. I: The Computer System," edited by B. H. McCormick, and B. J. Nordmann, Jr., Department of Computer Science Report No. 433, February 17, 1971.
- [6] "ILLIAC III REFERENCE MANUAL, VOL. II: Instruction Repertoire," edited by B. H. McCormick and B. J. Nordmann, Jr., Department of Computer Science Report No. 434, February 26, 1971.
- [7] Petri, C. A. "Kommunikation Mit Automaten," Schriften des Reinisch-West filischen Inst. Instrumental Math., and der Universitat Bonn, Nr. 2, Bonn 1962.
- [8] Holt, A. W., Shapiro, R. M., Saint, H. and Warshall, S., "Final Report for the Information System Theory Project," Applied Data Research, Inc., ADR ref. 6608, February 1968.
- [9] Patil, S. S., "Coordination of Asynchronous Events," M. I. T. MAC.TR, 72, June 1970.
- [10] Rey, Christian A., "Control Point Strategy and Its Automated Diagnosis," Thesis work printed as Department of Computer Science Report No. 450, University of Illinois,, June 1971 (In printing).
- [11] Michalski, R. D., Tareski, V. G., "Experiments with the Algorithm A^Q for the Interval Covering Synthesis," Department of Computer Science Report in progress.

4.6 DOCUMENTATION

4.6.1 External Documents Issued

Report No. 442	Michalski, R. S., McCormick, B. H., "Interval Generalization of Switching Theory", May 3, 1971.
Report No. 450	Rey, Christian A., "Control Point Strategy and Its Automated Diagnosis", June 1971.
Report No. 461	Michalski, R. S. "A Geometrical Model for the Synthesis of Interval Covers," June 24, 1971. (In printing)
Report No. 463	Rey, Christian A., "Control Point Design Using Modular Logic," June 1971 (In printing).
File No. 862	Borovec, R. T., "IMAGE 8: PROGRAMMED I/O FACILITIES (The External Bus)," June 9, 1971.
File No. 863	Borovec, R. T., "IMAGE 8: The Real Time Clock", June 16, 1971.

Seminars: Sight Sensory Systems

"Intersection Detection and Sofa Problems," Kiyoshi Maruyama,
Department of Computer Science, University of Illinois,
April 14, 1971.

"The Pattern Articulation Unit of Illiac III: Past, Present,
and Future," Richard T. Borovec, Department of Computer Science,
University of Illinois, April 21, 1971.

"Speech Recognition Survey," Bernard J. Nordmann, Jr.,
Department of Computer Science, University of Illinois,
May 5, 1971.

"Fault Diagnosis of Combinational Circuits and Its Application
for the Check-out of Control Point Logic," Christian A. Rey,
Department of Computer Science, University of Illinois, May 19, 1971.

"Prediction of Electronic Device Feature Behavior by Pattern
Recognition Methods", Professor F. N. Pokrovsky, Visiting Professor
at Purdue University from Moscow Power Engineering Institute,
May 26, 1971.

4.6.2 Logic Drawings Issued

The following new logic drawings have been drawn during the past quarter:

Sequence Tester Logic	40 drawings
Image 8 (PDP8/e Interface) Logic	19 drawings
Transfer Memory II	46 drawings
T.P. Control Logic	82 drawings
A.U. Control Logic	5 drawings

The remaining A.U. Control Logic drawings will be processed in the immediate future.

4.6.3 Engineering Drafting Report

During the last quarter a total of 450 drawings, including new logic drawings, drawing changes, layouts, flow-charts, thesis, report drawings and drawings related to Opto/Mechanical design of Illiac III, have been processed by 2118 drafting section.

S. Zundo

4.7 ADMINISTRATION

4.7.1 Personnel Report

Senior Staff

Professor Bruce H. McCormick - Principal Investigator

Assistant Professor - R. S. Michalski

Professional Staff

Robert C. Amendola

Richard T. Borovec

John S. Read

Research Engineering Assistant

S. Paul Krabbe

Electronic Engineering Assistant

Joseph V. Wenta

Digital Computer Technician II

George T. Lewis

Drafting

Stanislavs Zundo

Secretarial

Mrs. Roberta Andre'

Research Assistants

Jerry Chen

Walter Donovan

Lakshmi Goyal

Richard P. Harms

S. N. Jayaramurthy

Ahmad Masumi

Kiyoshi Maruyama

Bernard J. Nordmann

Peter Raulefs

Christian Rey

John C. Schwebel

Val Tareski

Kuo Wen

4.7.2 Computer Usage Log Summaries

1. The PDP8/e, which arrived early in June, was used approximately 100 hours, mostly for installation and checkout of interfaces to the Illiac III and IBM 360/75.
2. The Scanner-Monitor-Video System was used as follows:

Production & demonstration	223 hours
Preventive maintenance	16 hours
Corrective maintenance	12 hours
3. IBM 360/75 utilization totaled \$3,718.12

J. Read

U.S. ATOMIC ENERGY COMMISSION
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR
DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO.

C00-2118-0015

2. TITLE

QUARTERLY TECHNICAL PROGRESS REPORT - APRIL-JUNE, 1971

3. TYPE OF DOCUMENT (Check one):

☒ a.

Scientific and technical report

☐ b. Conference paper not to be published in a journal:

Title of conference _____

Date of conference _____

Exact location of conference _____

Sponsoring organization _____

☐ c. Other (Specify) _____

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a.

AEC's normal announcement and distribution procedures may be followed.

☐ b.

Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.

☐ c.

Make no announcement or distribution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

Professor Bruce H. McCormick

Principal Investigator

Illiac III Project

Organization

Department of Computer Science

University of Illinois

Urbana, Illinois

Signature

Date

July 7, 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

☐ a. AEC patent clearance has been granted by responsible AEC patent group.

☐ b. Report has been sent to responsible AEC patent group for clearance.

☐ c. Patent clearance not required.

5. ILLIAC IV

(This work was supported in part by the Department of Computer Science, University of Illinois, and in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center, under Contract No. USAF 30(602)-4144.)

REPORT SUMMARY

During the quarter the majority of the efforts in off-line diagnostics were concentrated in conversion of the card test generation system from the B5500 to the B6500. Automation Technology, Inc. maintenance engineers have been actively participating with Burroughs personnel in the on-line debugging of ILLIAC IV.

Work on a new technique to allow easier test generation through use of Roth's D-Algorithm has progressed well. Work next quarter will concentrate on larger logic partitions.

Progress has been made in the ILLIAC IV FORTRAN and operating system. Despite manpower shortages and an erratic B6500, all languages have been brought to what is essentially a refining stage. The operating system is on the verge of complete integration where it can be used to completely simulate the present ILLIAC IV system. Documentation of software has been extended and refined as evidenced by a library of macros and subroutines.

Two hardware interfaces between the PDP-11 and the ARPA network IMP were completed and debugged. Development of the ARPA Network Terminal System (ANTS) proceeded with near completion of the software coding, and staffing personnel for a graphics effort was begun.

Application efforts have continued in solving partial differential equations, solutions of systems of linear equations, eigenvalues, polynomial root finding and time series analysis. Education efforts included two seminars held during the quarter.

Project expenditures through June, 1971:

Burroughs Corporation \$27,477,000.00

University of Illinois 7,486,608.87

HARDWARE

5.1 Off-Line Diagnostics

5.1.1 PE Test

PE Control Logic Tests are as yet incomplete. Manpower originally scheduled to complete this area during the quarter was diverted to program conversion.

5.1.2 Card Test

Work on Card Test Generation was halted during the quarter while program conversion to the B6500 was taking place. Work is expected to resume in this area during the next quarter.

5.2 Program Conversion - ATI

The majority of the efforts in off-line diagnostics were concentrated in conversion of the card test generation system from the B5500 to the B6500. The problems of poor reliability and poor or misleading documentation, common to other areas of the project, had an impact in this area also. However, operation of the B6500 version of the system is expected during the next quarter.

5.3 Development - ATI

Work on a new technique to allow easier test generation through use of Roth's D-Algorithm has progressed well. Initial programming is complete and has been tested using PE cards. Work next quarter will concentrate on larger logic partitions, such as the CU cards and how to make them operationally available as a new aid in test generation.

5.4 ILLIAC IV Maintenance

ATI maintenance engineers have been actively participating with Burroughs personnel in the on-line debugging of ILLIAC IV. The

major areas which had active participation during this period were: PE debugging and modification, checkout of repaired PE/MLU cards, CU debugging, modification to spare cards for the CU, and the complete checkout/modification to the I/O area. The specification for a confidence test has been prepared, and the program is over 50% completed. Also the spare part break out needed for the operation of ILLIAC IV has been completed and supplied to the University.

5.5 Financial

At this point in the contract, ATI is approximately seven percent (7%) below the budgeted cost estimate.

SOFTWARE

5.6 Operating System

This quarter the major effort of the operating system group has been directed toward integrating, instrumenting, and stressing those parts of the operating system that reside in the B6500. The operating system has been integrated with the ILLIAC IV simulator to the extent that the Job Partner may communicate with a simulation of OS4 through a simulated TMU. Work presses forward on a broad band data path analogous to that of the BIOM and ILLIAC IV disk. A full simulation of the working of the I/O subsystem will not be attempted, however.

The operating system has been made "half-load proof"; i.e., it has been made largely impervious to B6500 failures. This is done by storing vital data and program status on the B6500 disk so that the operating system is almost always restartable.

The instrumentation added to the operating system this quarter allows the system to log the amount of CPU and I/O time spent in each operating system module by each ILLIAC IV job. It aids in allowing detailed job histories to be produced as a listing for the benefit of the user and of those maintaining the system. It allows the computer operators to interrogate the operating system to determine the status of any particular ILLIAC IV job, and lets the operator manipulate the operating system in case of hardware or software failure.

The ILLIAC IV resident module of the operating system, OS4, is being rewritten to take advantage of hardware interrupt changes and more flexible I/O capabilities.

Preliminary talks have been held with Burroughs Corp., Paoli, to discuss the integration with the On-line Diagnostic Programs (IDIAP) of the operating system.

Work has also begun on a special set of utility programs to effect format conversion between the various B6500 binary word formats and those of ILLIAC IV. These so-called "symbionts" each have a special Job Partner, and execute in the reserved area of the array. They do not

necessarily have any relationship with the current ILLIAC IV job and are, in reality, part of the data processor which currently executes on the B6500 only. By transferring this array oriented module to ILLIAC IV a considerable load will be removed from the B6500.

By the end of next quarter a preliminary version of the ILLIAC IV operating system will be ready to run as soon as the hardware debugging schedules permit.

5.7 Compilers and Translators

5.7.1 Cockroach

A working version of Cockroach (ILLIAC IV FORTRAN) is now available in the form of a Cockroach-to-Glypnir translator. This version includes functions, subroutines, and arrays of up to 64 in the "row" dimension. It has been successfully used to write and simulate programs by non-University users.

During the last two months of the quarter, the Cockroach group's efforts have been directed toward a quantitative study of the storage and execution efficiency of various storage allocation schemes. These schemes are for handling arrays whose row dimension is greater than the 64 which have been proposed in the past. The storage scheme is due to be completed this quarter.

Additional documentation for this language is being completed.

5.7.2 Glypnir

Some difficulty was experienced in transferring the Glypnir compile from the B5500 to the B6500. This was largely due to differences in MCP intrinsics between the two machines and their documentation.

Most of the quarter was devoted to reorganizing the compiler in preparation for including facilities which would allow the binding of separately compiled Glypnir subroutines with a Glypnir program. Effort was also spent in "cleaning up" the present compiler to make it faster and more reliable. Various operations of the compiler were timed and studied with a view to making a major speedup at a later date.

5.8 Assembler (ASK)

The Assembler has undergone a major revision during the quarter. Its peak speed has been more than doubled to over 2000 cards per minute; its output listing has been made more legible and can be printed faster, and a lot of dead wood has been cut from the compiler itself. In addition, a considerable number of "bugs" have been fixed. No changes have been made in the language.

5.9 Simulator (SSK)

Work on the simulator was concluded during this quarter. After transferring it to the B6500, a confidence test was written to test nearly all SSK functions. This is now in use. The simulator is tested for correct functioning each time there is a change of Burroughs software. No more work will be done on the simulator unless absolutely necessary.

5.10 Interactive Communications and Graphics

5.10.1 Interactive Communications

During the reporting period, two hardware interfaces between the PDP-11 and the ARPA network IMP were completed and debugged. The first interface will remain at the University of Illinois terminal site. The second interface was sent to Paoli, Pennsylvania. The second PDP-11 system and appropriate interfaces for connection to the B6500 at Paoli were checked out and delivered to Paoli for installation during the next reporting period.

Additional equipment received during this period included: Gould Electrostatic Printer/Plotter and a Computek storage scope, Model 400/15, terminal system. These two devices were successfully interfaced to the PDP-11 and programming efforts for support packages were started.

Development of the ARPA Network Terminal System (ANTS) proceeded with near completion of the software coding. Initial debug and checkout operations on the network NCP were also started. Subsidiary programming efforts were started to develop a version of ANTS for the Paoli PDP-11 (PANTS) to allow a specialized remote access link between

the University and the Paoli B6500. Completion of this part of the project is scheduled for August, 1971.

5.10.2 Graphics

During the quarter, staffing of personnel for a graphics effort was begun. A level of 2 1/2 full-time employees was reached, and efforts were started to provide for implementation of support packages for graphics devices such as the Gould printer/plotter, Computek storage scope, network access to graphics utilities sites such as Utah, BBN, MIT, etc. In addition, work was started on a basic graphics structure system to provide a generalized support structure linking all Center graphics activities and activities on the ARPA network.

5.11 Library

A small library of ILLIAC IV subroutines and macros, together with a program to aid the general user in collecting them in proper sequence, has been made available. These include SIN, COS, ARCTAN, LN, SQRT, and EXP in 32- and 64-bit mode. ITU routines and macros are also available. However, more work needs to be done before these can be considered truly "professional."

APPLICATIONS

5.12 Numerical Analysis

5.12.1 Partial Differential Equations

5.12.1.1 Block Jacobi Method

Work on the Block Jacobi method is being continued to solve the elliptic type partial differential equations for given boundary conditions.

Let R be a rectangular region with boundary S , and let $g(x)$ be a continuous function defined on S . Find a function $u(x,y)$ which is continuous on $R + S$, satisfies on R the following equation:

$$a(x,y) \frac{\partial^2}{\partial x^2} u(x,y) + b(x,y) \frac{\partial^2}{\partial y^2} u(x,y) + c(x,y) \frac{\partial}{\partial x} u(x,y) + d(x,y) \frac{\partial}{\partial y} u(x,y) + e(x,y) u(x,y) = f(x,y) \quad (1)$$

and satisfies on S the condition $u(x,y) = g(x,y)$ (Dirichlet condition). Sometimes $\frac{\partial u}{\partial \eta}$ ($\frac{\partial}{\partial \eta}$ means the normal derivative--Neumann condition) or linear combination of u and $\frac{\partial u}{\partial \eta}$ is prescribed on S instead of u itself. The following are analytic functions of X and Y in R and satisfy in $R + S$ the conditions $a > 0$, $b > 0$ and $e < 0$: a , b , c , d , e and f .

The corresponding finite difference equations on $n \times n$ mesh points in R are given by,

$$c_{1,i,j} u_{i-1,j} + c_{2,i,j} u_{i+1,j} + c_{3,i,j} u_{i,j-1} + c_{4,i,j} u_{i,j+1} + c_{5,i,j} u_{i,j} + c_{6,i,j} = 0 \quad 1 \leq i,j \leq n \text{ (= even number)} \quad (2)$$

where i, j are indices of mesh points.

In point-iterative methods the value of each component $u_{ij}^{(n)}$ is determined by an explicit linear formula. That is, the n th approximation of the (i,j) component can be expressed by using already computed values of the other unknowns. In block-iterative methods, on the other hand, groups of components of $u^{(n)}$ are modified simultaneously by solving a system of linear equations. Therefore individual components are defined implicitly in terms of the others in the same group.

The Block Jacobi method is one of the block-iterative methods and is equivalent to the successive line over the relaxation method (SLOR) except that the acceleration parameter is changed at each step in the former case.

We write the system of equations (2) in matrix notation as

$$\begin{bmatrix} D_1 & F_1 & & & \\ E_1 & D_2 & F_2 & & \\ & E_2 & D_3 & F_3 & \\ & & & \ddots & \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ \vdots \\ \vdots \end{bmatrix} \quad (3)$$

$$\text{or in short } A_u = k \quad (3')$$

where D_i 's are tridiagonal matrices, E_i 's and F_i 's are diagonal matrices, $u'_i = (u_{i1}, u_{i2}, \dots, u_{in})$ and $k'_i = (k_{i1}, k_{i2}, \dots, k_{in})$ which consists of boundary values and $C_{6,i,j}$'s, (Components of D, E, F consist of C_m 's ($m = 1, 2, \dots, 5$)).

If we interchange the order of components of u in (3') so that the mesh points with odd i 's are in the upper half of u and those with even i 's are in the lower half of u without changing j 's (in other words, to separate odd rows from even rows), we have the following system of linear equations instead of (3):

where α_m is the mth acceleration parameter for odd lines.

2) Solve

$$\begin{bmatrix} D_2 & & & 0 \\ & D_4 & & \\ & & D_6 & \\ 0 & & & \end{bmatrix} \begin{bmatrix} u_2 \\ u_4 \\ u_6 \\ \vdots \\ u_n \end{bmatrix}^* = \begin{bmatrix} k_2 \\ k_4 \\ k_6 \\ \vdots \end{bmatrix} - \begin{bmatrix} E_1 & F_2 & & 0 \\ & E_3 & F_4 & \\ & & E_5 & F_6 \\ 0 & & & \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \\ u_5 \\ \vdots \end{bmatrix}^{(m+1)} \quad (7)$$

for $\begin{bmatrix} u_2 \\ u_4 \\ u_6 \\ \vdots \end{bmatrix}^*$ Note that $\begin{bmatrix} u_1 \\ u_3 \\ u_5 \\ \vdots \end{bmatrix}^{(m+1)}$ is used instead

of $\begin{bmatrix} u_1 \\ u_3 \\ u_5 \\ \vdots \end{bmatrix}^{(m)}$ on the right hand side of (7).

Then obtain the (m+1)st approximation by extrapolation

$$\begin{bmatrix} u_2^{(m+1)} \\ u_4^{(m+1)} \\ u_6^{(m+1)} \\ u_8^{(m+1)} \\ \vdots \end{bmatrix} = \begin{bmatrix} u_2^* \\ u_4^* \\ \vdots \end{bmatrix} + \alpha_{m+1/2} \begin{bmatrix} u_2^* - u_2^{(m)} \\ u_4^* - u_4^{(m)} \\ \vdots \end{bmatrix}$$

where $\alpha_{m+1/2}$ is the mth acceleration parameter for even lines.

α_i 's are given either in the form of a table or calculated recursively from the previous value at each step ($\alpha_m = f(\alpha_{m-1/2})$).

Note that in either step a system of linear equations for each row can be solved independently of others. This is a very nice feature for parallel computation; one PE solves one row. The usual Thomas algorithm is used to solve linear equations of tridiagonal matrices.

Each PE contains two adjacent rows so that all 64 PE's are enabled when $N = 128$. The original version in which each PE contains one row has been implemented in Glypnir and has been run successfully. The program for the new storage scheme is being programmed and debugged.

5.12.1.2 Numerical Solutions of Problems in Hydrodynamics

A computer program using the Lax-Wendroff finite difference scheme, which has been tested with Burger's Equation [1], is under development for calculating the compressible, inviscid, transonic flow around a circular cylinder. A consistent formulation of the problem, which is eminently suitable for parallel computation, has been successfully arrived at. This is done in such a manner that, for shockless flow, no additional problems are posed in satisfying the boundary conditions at the body surface as well as at the far field free stream boundary. The same instructional stream is fed to each arithmetic unit whether it is

computing boundary point values or interior point values. Thus no computing time is lost at the boundary points. This leads to high machine efficiency. In addition, a computational algorithm that permits a change in the mesh sizes with a minimum of change in the instructional stream to the processing units is under investigation.

The computational technique is based on the asymptotic method of calculating steady state solutions using the time dependent formulation [2]. The two-dimensional problem is computed in cylindrical polar coordinates. Variable mesh sizes are employed to accommodate the rapid variation of the physical parameter, especially at the forward stagnation point.

Preliminary results from the computer program indicate that the formulation currently employed does yield a solution which exhibits properties similar to the solution for inviscid flow obtained by Moretti [3]. The present formulation, however, possesses the added advantage of being amenable to extension to three dimensions and more complicated body shapes for practical applications.

5.12.2 Solutions of Systems of Linear Equations

During this quarter, the ALGOL versions of three subroutines were converted to B6500 ALGOL, debugged, and documented. They are now available for use. The three routines are:

- (1) Triangular decomposition of a square matrix
- (2) Gaussian elimination
- (3) Matrix inversion (triangular decomposition method).

The ASK versions of the above 3 routines are partially debugged. Changes are being made currently in the storage allocation scheme and I/O scheme. When these changes are completed, the routines will be able to handle matrices up to 1500 x 1500 (as opposed to 630 x 630), and the I/O wait time will be significantly reduced in most cases. Debugging will continue as soon as these changes are completed. A subroutine to perform the conjugate gradient method has also been started.

We have devoted ourselves to two main activities concerning

solution of systems of linear equations and matrix inversion. First, B6500 ALGOL subroutines, namely HOUSEHOLDER and MODIFY, have been coded, debugged, and tested on the machine. A double precision version of HOUSEHOLDER has been implemented to test several ill-conditioned matrices. The second effort has been to document existing programs according to ILLIAC IV subroutine standards. The documents are available for the ALGOL programs, and the ASK documents are being prepared. The HOUSEHOLDER upper triangularization method as described in QPR 1970 [4] has been modified for purposes of decomposing an $n \times n$ matrix A of rank r into the product of two matrices $A = BC$ where B is an $n \times r$ matrix and C is an $r \times n$ matrix.

5.12.3 Eigenvalues

5.12.3.1 The Eigenvalue Problem

In this quarter, the following three ALGOL programs have been coded and tested successfully:

- (1) Solution of systems of complex linear equations (CU decomposition)
- (2) Inverse iteration for finding eigenvectors
- (3) Eigenvector solution of the matrix Riccati equation.

Actually, programs (1) and (2) are a part of program (3), but for the purpose of completeness and for their independent utility, they are tested separately with numerical examples. These will be documented in ALGOL and GLYPNIR in the next quarter.

For problem (3), both Eberlein's and QR's methods [5] are used, depending on whether the matrix of coefficients of the corresponding linear dynamic system is diagonalizable.

Two methods of transforming a matrix into Upper Hessenberg form,

- a) using elementary unitary transformations (HOUSEHOLDER's method)
 - b) using elementary stabilized transformations,
- were investigated. It was concluded that the second method is efficient

only for core-contained matrices and has been coded in B6500 ALGOL (File MMTI/UH/ALG) (debugged and running) and in ASK (MMTI/UH/I4ASK) (being debugged). For non-core contained matrices, in order to keep disk memory transfer minimum, the HOUSEHOLDER's method is being coded in ASK. Macro definitions for using the ILLIAC IV disk as a direct-access device are being coded to facilitate access to rows and columns stored on disk and their transfer to core memory. The main problem is to have the double QR-algorithm for non-core contained matrices running in ILLIAC IV, taking advantage of the parallelism of the machine by choosing an appropriate value of the origin shift. For this purpose the algorithm has been coded in B6500-ALGOL (MMTI/QR/ASG) (debugged and running) to experiment with different matrices, and if possible, empirically determine the best origin shift and the suitability of this choice for the non-core contained matrices and the storage scheme both in ILLIAC IV disk and memory.

For the purpose of numerical stability, the matrices have to be balanced. The method described by Parlett and Reinsch in their paper "Balancing a Matrix for Calculation of Eigenvalues and Eigenvectors" is to be coded as a subroutine which produces the balanced matrix.

5.12.3.2 Eigenvalues and Eigenvectors of Symmetric Tridiagonal Matrices

5.12.3.2.1 QR-algorithm for Symmetric Tridiagonal Matrices

QR-algorithm for finding the eigenvalues of symmetric tridiagonal matrices written last quarter has been modified into an independently compiled procedure on the B6500. The procedure has been debugged and tested for different-size matrices. The calling sentence is

STREVA (D,E,N,LMT,EVA) ; .

The complete document of this procedure has also been completed. This algorithm is now being written for the ILLIAC IV in GLYPNIR.

5.12.3.2.2 Inverse Iteration for the Corresponding Eigenvectors

This algorithm written last quarter has been modified into an independently compiled procedure on the B6500, debugged, and tested. The calling sentence is STREVC (C,B,N,W,ML,MACHEPS,Z) ; . The complete document of this procedure has also been finished. This algorithm is now

being written for the ILLIAC IV in GLYPNIR.

5.12.3.2.3 Iterative Method for Solution of the Eigenproblem

Gradient methods for finding the maximal (or minimal) eigenvalue and its corresponding eigenvector for large scale symmetric matrices are being explored for the ILLIAC IV. The algorithm can be described as follows..

Consider the Rayleigh quotient as a function f of n variables $\bar{x}$, i.e.,

$$f(\bar{x}) = (A\bar{x}, \bar{x})/(\bar{x}, \bar{x})$$

then the gradient

$$\nabla f = g = 2(A\bar{x} - f\bar{x})/(\bar{x}, \bar{x})$$

can be readily evaluated. Maximize (or minimize) the function f by gradient methods, we get the maximal (or minimal) eigenvalue of matrix A , and the final value of vector $\bar{x}$ is its corresponding eigenvector.

5.12.4 Polynomial Root Finding

Two polynomial root-finding routines have been coded, debugged and documented for B6500 ALGOL. The first is a procedure finding real roots by using Sturm sequences. The second finds all roots, real and complex, using root squaring and sub-resultant procedures.

A real root finder has been coded in ASK and will be debugged. A document on this routine will come out August 11, 1971.

5.12.5 Time Series Analysis

5.12.5.1 Identification and Estimation of Stochastic Models

Algorithms for identifying Stochastic models and estimating their parameters have been studied in this period. Programs are being written in ALGOL. Comparisons are to be made among several methods, such as the maximum likelihood method, the linear least-squares method and other non-linear methods, to estimate the parameters of the models.

5.13 Linear Programming

The past quarter has again seen an assembler-simulator package insufficient to facilitate code debugging. Each attempted simulation primarily served to assist in SSK debugging. With this in mind, a brief, intensive joint effort between LP and the ILLIAC IV Project's simulator maintenance group was undertaken to produce an SSK exerciser. While the work was considered a success, at quarter's end the simulator remained nearly unusable.

Unable to proceed with INVERT debugging, attention returned to the SETUP and SETDOWN codes--those programs massaging data from "external" storage conventions into the format of the ILLIAC IV LP package, and back again. Over the quarter, the early B5500 routines were translated from XALGOL to ALGOL. Major recoding was done to accommodate a workload shift from the B6700 to the ILLIAC IV. As the period closed preliminary flow-charts were being developed to extend the ILLIAC IV's load to include label identification and data allocation--the primary work remaining on the B6700.

Major SETDOWN codes were written and syntactically debugged, with synthetic test-data being developed. In developing the SETDOWN procedures, some output formats were modified to facilitate translation of solution information. These modifications are being incorporated into the LPS code and documentation. It now appears quite possible for virtually the entire SETUP/SETDOWN package to be placed on the ILLIAC IV, with the primary constraint to the realization of this being time and manpower available.

Some time has been spent this quarter exploring the possibility of a collaborative effort with the group of Professor A. Charnes at University of Texas. Particular areas of mutual interest may be algorithm development for mathematical programming and the solution of large linear programming models. This effort will continue into the next quarter.

5.14 Signal Processing

The following ALGOL procedures have been documented and tested for use on the B6500 computer:

- (1) a plotting procedure called GRAPH,

- (2) a procedure called FHANN for computing the impulse response and power of a band-pass filter,
- (3) a one-dimensional fast Fourier transform procedure called FFT,
- (4) a one-dimensional inverse fast Fourier transform procedure called IFT,
- (5) a one-dimensional conventional Fourier transform procedure called FURY,
- (6) an auto correlation procedure called AUTO.

5.15 Education

5.15.1 Teaching

During this quarter, activities included teaching the graduate course on ILLIAC IV for the University of Illinois, a two-day ILLIAC IV seminar for Lawrence Radiation Laboratories (LRL) and a one-day seminar for attendees of the Fluid Dynamics Workshop given by the staffs of the Center for Advanced Computation, Coordinated Science Laboratory and ILLIAC IV Project at the University of Illinois.

5.15.1.1 CS-491: Architecture, Applications and Languages for a Parallel Computer

The Spring Semester of CS-491 had 14 students; 4 took the course for one unit, 5 for 1/2 unit and 5 audited the course. The course material covered was substantially the same as in previous semesters (refer to outline in previous QPRs) but with slightly more emphasis on the ILLIAC IV programming languages--ASK, GLYPNIR, and FORTRAN.

The content next fall will stress more heavily the operation of the ARPA Network as ILLINOIS becomes more and more a working node of the Network.

In order to make CS-491 accessible to a larger class of students at the University of Illinois, the course will be changed to the 300 level (can be taken for graduate credit or by advanced undergraduates). The new course designation will be CS-397: Special Topics in Computer Science--The ILLIAC IV Computer System. Following is a

course description:

CS 397 Special Topics in Computer Science--

The ILLIAC IV Computer System

Review of conventional and unconventional organization hardware structure, diagnostic and test/repair equipment, programming languages, operating system and selected applications. ILLIAC IV will be accessed remotely in the solution of class problems. One-half to one unit.

Prerequisite: Consent of instructor.

5.15.1.2 Two-Day Seminar for LRL

On May 13 and 14 a two-day seminar on ILLIAC IV was given to three employees of LRL. The seminar outline follows:

May 13, 1971

I. Overview

10:00 am - 12 noon

A. Background Conventional and Unconventional Organizations

1. Overlap

a. Buffer

b. Pipeline

2. Replication

a. General Multiprocessor

b. Recentralize Memory

c. Recentralize ALU - Intrinsic Multiprocessor

d. Recentralize CU - Vector or Array Processor

3. Both 1) and 2) above

B. ILLIAC IV is a Vector Processor

C. Hardware Structure

1. Organization Chart

2. ILLIAC IV Array - General Description

3. Some Illustrative Problems

4. ILLIAC IV I/O System

5. ILLIAC IV System

6. Some Indicators of Powers of ILLIAC IV

D. Diagnostics and Test/Repair Equipment

- E. Software
- F. Applications

II. Programming Languages 2:00 pm - 5:00 pm

A. ASK

1. Background - Review, Notation, Conventions
2. Sample Problems
 - a. Summing an Array of Numbers
 - b. Finding the Largest Value in an Array of Numbers

May 14, 1971

~~B. ALGOL 1:00 pm - 1:30 pm~~

~~C. GLYPNIR 1:30 pm - 3:00 pm~~

5.15.1.3 One-Day Seminar for Fluid Dynamics Workshop

On June 15 a one-day seminar on ILLIAC IV was given to 26 attendees of the Fluid Dynamics Workshop. Following is the seminar outline:

I. Overview

9:00 am - 12 noon

A. Background Conventional and Unconventional Organizations

1. Overlap
 - a. Buffer
 - b. Pipeline
2. Replication
 - a. General Multiprocessor
 - b. Recentralize Memory
 - c. Recentralize ALU - Intrinsic Multiprocessor
 - d. Recentralize CU - Vector or Array Processor

B. ILLIAC IV is a Vector Processor

C. Hardware Structure

1. Organization Chart
2. ILLIAC IV Array - General Description
3. Some Illustrative Problems
4. ILLIAC IV I/O System
5. ILLIAC IV System
6. Some Indicators of Powers of ILLIAC IV

- D. Diagnostics and Test/Repair Equipment
- E. Software
- F. Applications
- G. Current Status

II. Programming Languages

2:00 pm - 5:00 pm

A. ASK

1. Background - Review, Notation, Conventions
2. Sample Problems
 - a. Summing an Array of Numbers
 - b. Finding the Largest Value in an Array of Numbers

5.15.2 Consulting

A small amount of time was spent this quarter in one-to-one consulting with people from the Center for Advanced Computation, ILLIAC IV, and people outside the University of Illinois concerning the ILLIAC IV and B6500.

5.15.3 Training

Education has taken on the responsibility of training two Affirmative Action employees to become Computer Programmers. In order to teach the trainees, a master-apprentice program is being used--the trainee meets with a different experienced programmer for one hour each day. Each experienced programmer teaches the trainee as much of the crafts of programming as he feels the trainee needs at his current level of development. The method of teaching ranges from lectures, to assigning problems, to working directly with the trainee.

Additionally, the trainees are learning the BASIC language for the PDP-11 and ALGOL for the B6500 and are solving a prescribed set of problems designed to instill the fundamentals of algorithmic problem-solving. Each problem is flow-charted, coded in both BASIC and ALGOL and finally run on the appropriate machine to solution.

5.15.4 Technical Writing

5.15.4.1 An Introductory Description of the ILLIAC IV System: ILLIAC
IV Document No. 225

The first version of ILLIAC IV Document 225 will be ready for distribution early next quarter. Following is
ing instructions for the document:

Abstract

Written specifically for an applications programmer, the book presents a tutorial description of the ILLIAC IV System. The initial version contains three chapters--Background, Hardware Structure, and The Assembly Language--ASK. Many illustrative problems are used to educate the beginner in the use of the ILLIAC IV System.

Reading Instructions

This book was written for an applications programmer who would like a tutorial description of the ILLIAC IV system before attempting to read the reference manual. As a tutorial, the level of detail presented in this book is fairly general; particular information can be found in the Burroughs Reference Manual "ILLIAC IV Systems Characteristics and Programming Manual."

In order to use this book most effectively, the chapters should be read in order. The reader who wants a very quick look at the capabilities of ILLIAC IV can skim just the summaries of parts A, B and C of Chapter I and begin reading on page II-55. He may then read pages II-1 through II-19; skipping the detailed description of the ILLIAC IV Array. Pages II-40 through II-72 are optional; the reader should at least look at them and decide for himself. As much of Chapter III as possible should be read--the instruction repertoire, more than anything else, defines the capabilities of a computer. A valid answer to the question "What is ILLIAC IV?" would be to hand the questioner a description of each instruction in the repertoire.

For a more complete understanding, however, the reader should come back and read the sections he skipped on the first pass. It is the nature of ILLIAC IV, to a degree much greater than the conventional computers, that its hardware structure is bound up very closely with its capabilities. It is therefore necessary that the reader spend the time

necessary to understand the architecture of ILLIAC IV.

The Table of Contents in the front of the book is in an abbreviated format while each chapter will be preceded by a finer Table of Contents. Each chapter will also have its own Glossary which can be detached for handy reference in addition to a cumulative Glossary at the end of the book. As these become available they will be distributed to you.

Chapter I presents the background concepts necessary for an understanding of ILLIAC IV. A short section is devoted to the historical development of digital computers, and their evolution is described in terms of the problems that had to be solved. After conventional computer organizations are described, unconventional ones are presented as design options to speed up the operation of a computer. Two design philosophies, overlap and replication, represent two major methods used to increase the computer's operational speed. Overlap is effected by the buffer and pipeline mechanism and replication is embodied in the general multiprocessor. ILLIAC IV is shown to be a variant of a general multiprocessor using buffering and a modified pipeline mechanism in the instruction execution section.

Chapter II describes the architecture or the hardware structure of ILLIAC IV. The ILLIAC IV Array is discussed in broad terms followed by some illustrative problems which point out some of the similarities and differences between problem-solving on sequential and parallel machines. The problems also serve to illustrate how the hardware components are tied together. Following is a more detailed description of the ILLIAC IV Array, then another illustrative problem (Laplace's equation describing steady-state temperature distribution in two-dimensions) followed by some data allocation considerations; the ILLIAC IV I/O System is discussed briefly, and some conclusions and opinions end the chapter.

Chapter III presents the Assembly Language ASK in a functional and pragmatic way: a problem is described and then only those ASK instructions necessary for the solution are described. In this way the five problems introduce forty ASK instructions and the flavor of the assembly language which, from a programmer's standpoint, is an indication

of the capabilities of ILLIAC IV itself. The five problems are:
Summing an array of numbers, Finding the largest value in an array of
numbers, Matrix multiplication, Matrix transpose, and Laplace's equation
described in Chapter II.

Chapters IV through XI and the Glossary are not included in
this edition but will be supplied as soon as they become available.

5.15.4.2 IEEE Article

Education is coordinating and contributing to a future article
to be presented in the IEEE Proceedings under the tentative title "The
ILLIAC IV Supercomputer." The first draft will be sent to the IEEE by
the end of the next quarter.

ADMINISTRATION

5.16 Administration and Services

5.16.1 Contract Status

Burroughs has advised that the schedule for completion of the ILLIAC IV Computer has slipped from June to August and September.

<u>Item</u>	<u>From</u>	<u>To</u>
Diagnostic Programs	June 1971	September 1971
Deliver One Quadrant ILLIAC IV System. . . .	June 1971	September 1971
Commence Final Acceptance Testing of ILLIAC IV	June 1971	August 1971
Documentation	June 1971	September 1971
Engineering Drawings	June 1971	September 1971
Test Equipment	June 1971	September 1971

5.16.2 Financial Report

Budgeted expenditures--Fourth Quarter, Fiscal Year 1971

	<u>April</u>	<u>May</u>	<u>June</u>	<u>Total for Quarter</u>
Burroughs	\$136,234.00	\$136,295.00	-0-	\$272,529.00
University	215,302.00	215,302.00	\$215,302.00	645,906.00

Actual expenditures and commitments--Fourth Quarter, Fiscal Year 1971
(through May, 1971)

	<u>April</u>	<u>May</u>	<u>June</u>	<u>Total for Quarter</u>
Burroughs	\$199,000.00	\$175,000.00	*-0-	\$374,000.00
University	330,810.19	341,524.85	**0-	672,335.00

Total budgeted and actual expenditures to date:

	<u>Total Budgeted Expenditures Through June, 1971</u>	<u>Total Actual Expenditures Through May, 1971</u>
Burroughs Corporation	\$27,687,946.00	\$27,477,000.00
University of Illinois	7,673,203.92	7,486,608.00

*NA - Report of Expenditures from Burroughs covering June not received
as of July 6, 1971.

** - Report of University expenditures for June not completed as of
July 6, 1971.

REFERENCES

- [1] Rajan, S., "Numerical Solution of the Partial Differential Equations of Gas Dynamics," In preparation, Center for Advanced Computation, University of Illinois at Urbana-Champaign.
- [2] Magnus, R., and Yoshihara, H., "Inviscid Transonic Flow Over Airfoils," A.I.A.A. Journal, Dec. 1970.
- [3] Moretti, G., "Transient and Asymptotically Steady Flow of an Inviscid, Compressible Gas Past a Circular Cylinder," Polytechnic Inst. of Brooklyn Report.
- [4] ILLIAC IV Quarterly Progress Report, July, August, and September, 1970. "Matrix Inversion and Solution of Linear Algebraic Equations." ILLIAC IV Document No. 236, DCS Report No. 415 (October 15, 1970).
- [5] ILLIAC IV Quarterly Progress Report, July, August and September, 1970. "The Eigenvalue Problem." ILLIAC IV Document No. 236, DCS Report No. 415 (October 15, 1970).

THESES

- Alsberg, P.A. "OSL/2 An Operating System Language." Ph.D. Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.
- Beals, A.J. "The Automatic Generation of Deterministic Parsing Algorithms." Ph.D. Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.
- LaFrance, J.E. "Syntax-Directed Error Recovery for Compilers." Ph.D. Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.
- Nakamoto, H. "On-Line Diagnosis of ILLIAC IV." Master's Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

DOCUMENTS

Gary, J. "PDELAN--A Programming Language for the Solution of Partial Differential Equations." ILLIAC IV Document No. 229, Department of Computer Science File No. 853, ILLIAC IV Project, University of Illinois at Urbana-Champaign (July 31, 1970).

Koga, Y., K. Naemura and C. Tanaka. "Logic Simulation for the ILLIAC IV Processing Element." ILLIAC IV Document No. 223, Department of Computer Science File No. 849, ILLIAC IV Project, University of Illinois at Urbana-Champaign (January 22, 1971).

Schuster, S.A. "The Tuning of Buffer Parameters for the ILLIAC IV Data Management System." CAC Document No. 3, Center for Advanced Computation, University of Illinois at Urbana-Champaign (January 15, 1971).

Tanaka, C., L. Abel, and K. Naemura. "Parallel Logic Simulator and Its Use for Test Generation." ILLIAC IV Document No. 222, ILLIAC IV Project, University of Illinois at Urbana-Champaign (July 13, 1970).

6. NUMERICAL METHODS, COMPUTER ARITHMETIC AND ARTIFICIAL LANGUAGES

(Supported in part by the National Science Foundation under Grant No. US NSF GJ-812).

6.1 Computerized Mathematics

The research described in previous progress reports were collected into a Ph.D. thesis which was accepted by the Graduate College in June. The author received his Ph.D. on June 19, 1971.

We were able to show completeness of a very general form of the naming rule (see previous report). However, such a general form of the rule is not computationally feasible. Thus, the existence of a complete and practicle form of the naming rule is still open. We were able to demonstrate a restricted form of the naming rule which was sufficient for an admittedly small number of sample problems.

(L. J. Henschen)

6.2 Computational Geometry

During this quarter, a DCL Report* was written and submitted for publication. A summary follows:

An intersection detection procedure for polyhedral objects by means of face-to-face intersection analysis and its application for a path-finding problem in geometrically constrained space.

(K. Maruyama)

*Maruyama, K., "A Procedure for Detecting Intersections and Its Application," Department of Computer Science Rept. No. 449, 1971.

6.3 Educational Timesharing System on the PDP-11

ETS is designed to provide an interactive computing facility for non-sophisticated or beginning assembly language users. It will operate in a timesharing mode with 12-16 students who are connected to the processor via KSR-33 Teletypes. During its first semesters of operation it will support a teaching package called GIZMO (a question-answer routine used to introduce fundamentals to the student) and an on-line programming facility similar to DOS. Students will be able to write a program with an on-line editor, assemble it with a modified PAL-11R and run it under a powerful interpreter-debugging package. Concurrent with this operation, ETS will provide batch-type assemble and go facilities for students whose programs are on card decks.

The basic monitor is a revision of DEC's RSTS (formally BTSS) monitor (before the addition of the editor-compiler). The assembler is a modification of PAL-11R VOOOA. The editor, interpreter, and GIZMO are written locally. Our broad philosophy rests on making the system as simple as possible for the students while remaining compatible with DEC software specifications-conventions.

The monitor itself will be 6K words in size, leaving a 6K user area. However, it seems unreasonable to expect to operate with any efficiency where it is necessary to give the entire user area to a single task (PAL-11R will require 6K). A minimum job-mix seems to be of the order of 1 assembly and 2 edits (2K each) or equivalent (GIZMO = 2K, INTERPRETER = 2-3K). Thus a 10K user area would be needed. For this we need an additional 8K of core memory. (We presently have 8K and are in the process of adding an additional 4K).

As it is necessary to constantly maintain student interest, using the KSR-33 for long listings is not feasible. Output of PAL-11R must go to a line printer. Also, the line printer is to be the output device for batch operation. DEC has given us a line printer which will be installed as soon as possible.

The editor and GIZMO are both operational and are being refined for use this fall. Lessons for GIZMO are also being written. Plans call for 6-10 lessons of 60 questions to be written.

(D. B. Gillies)
(T. Chen)
(D. W. Oxley)
(A. Davis)
(S. Hansen)
(M. Stone)
(R. Atkinson)

6.4 Debugging Interpreter/Trace System (DITS)

DITS is being developed by the Computer Science Department of the University of Illinois as an educational tool for machine language programming for the PDP-11. It will be used interactively, primarily by students learning systems programming. By focusing on this class of user we hope to develop a very powerful debugging system, and gain insight into languages and debugging techniques in general.

Since student jobs are short, and are nearly all debugging runs, and since our PDP-11 is a fast dedicated resource it is feasible to run most, if not all such jobs interpretively. As well as interpreting every instruction and then executing it, DITS can perform certain tests, and maintain use tables and other statistics, automatically or as a result of user commands. The more it knows of the programmers intentions, the more effectively it can monitor whether the program carries these out.

The goal is to present the student with a virtual machine which is instrumented, introspective, intelligent and interactive.

Many such systems are possible, and their design and testing is a promising research area. The choice of one to try first should be made within the general context of what is possible and desirable. Notice that syntax errors detected by an assembler or compiler resemble conventional computer-aided instruction since the computer can verify mechanically and directly from the form of the answer whether it follows the syntax rules. DITS is an example of a different and more demanding instructional use of the computer, namely the student designs a complex structure (in this case a program) whose correct functioning cannot be deduced from its form alone--it must be obeyed or simulated. This type of instrumented simulation requires a large number of instruction executions, each of a very simple type, and can be done more economically by a mini-computer or a network of mini-computers than by a large centralized computer.

Clearly, the more information available to DITS, the more sophisticated its checks can be. Such information can be passed through programming conventions, extensions to the assembler (both what it will accept and what it will tell DITS), user declarations and commands before and during interpretive execution, and user response to DITS typed enquiries. User commands in DITS correspond to a more or less conventional on-line debugging system except that more information is maintained by DITS and can be examined retrospectively by the programmer. Other DITS activities, using its knowledge derived from programming conventions, declarations, stylized comments, etc. are intended to flag suspicious or incorrect actions and outcomes automatically. The user is free to suppress or modify such testing or printing. Furthermore, the user has the option to run under DITS, partly under DITS or independent of DITS as he chooses.

6.5 PDP-11 Hardware Progress Report

Considerable progress has been made on PDP-11 hardware over the past quarter. An 8-bit extension to the processor status (PS) word and a clock were added to the PDP-11 processor. Detailed design of the relocation hardware was completed by Chuck Hyde and preliminary drawings were produced. The drawings were sent to C. Hyde for verification, were modified slightly, and have been returned. Priorities have pushed back the target date for working relocation hardware to Mid-September.

Production runs of the TTY interface boards were received from the printed circuit shop. A total of 54 boards were stuffed and soldered by the PDP-11 staff. Inventory and status of the TTY interface boards as of July 25 is shown below:

	<u>ON HAND</u>	<u>Not Working</u>
Transmitter/Receiver Boards (Double Height)	20*	6
Interrupt Boards (Single Height)	20*	2
Addressing Boards (Single Height)	20*	3

* Includes 2 DEC boards

Five TTY's are currently on line and seven additional units have been allocated for PDP-11 use. These units will require some checkout and rewiring and will be installed when this work is completed.

The design of the interface for an Ampex 4K memory mod is complete and the boards are being wire-wrapped. The additional 4K should be operational by the 1st of August. This will bring total core up to 12K.

DEC has donated a Data Products Model 2410 line printer to the project. This is a 132-column drum type impact printer with a speed

range of 245-1110 LPM. The printer should arrive about the 1st of August and will require about a week for checkout and installation. Prospects are grim for obtaining an additional 8K core, a card reader, and perhaps another disk platter before the fall term begins. Humidity and noise control in Room 123 are still significant problems.

(J. D. Miller)

6.6 Factorization Techniques Used in the Solution of Partial Differential Equations

We continued to work on a new algorithm which uses factorization methods to solve the large sets of algebraic equations that arise in the solution of partial differential equations by implicit numerical techniques.

Factorization methods solve the equation

$$AX = q$$

as follows.

An auxiliary matrix, B, is added to A so that (A+B) factors into known sparse triangular matrices L and U. The solution X is then given by the limit of the sequence $\{X_n\}$ generated from

$$(A+B)X_{n+1} = (A+B)X_n - \tau_n(AX_n - q).$$

The values of τ_n are determined from the values of the extreme eigenvalues of $(A+B)^{-1}A$.

The algorithm developed last quarter simultaneously solves $AX = q$ and generates an approximate eigenvector of $(A+B)^{-1}A$ corresponding to an extreme eigenvalue. This approximate eigenvector is used to calculate an approximate eigenvalue of $(A+B)^{-1}A$. The approximate eigenvalue is then used to improve the sequence of τ_n 's.

Let a be an approximation to the smallest eigenvalue of $(A+B)^{-1}A$ and b be an approximation to the largest eigenvalue of $(A+B)^{-1}A$.

The approximate eigenvectors y_n satisfy

$$y_{n+1} = P_n((A+B)^{-1}A)y_1$$

where $P_n(X) = T_n\left(\frac{a+b-2X}{b-a}\right)/T_n\left(\frac{a+b}{b-a}\right)$ and $T_n(X)$

is the Tchebychev polynomial of degree n . The approximate eigenvalue is given by

$$U = \frac{\langle Ay_{n+1}, y_{n+1} \rangle}{\langle (A+B)y_{n+1}, y_{n+1} \rangle}.$$

It was shown that the convergence of y_n to an eigenvector corresponding to the largest eigenvalue is most rapid if a is equal to the smallest eigenvalue and b is equal to the second largest eigenvalue. It was also shown that the convergence of the eigenvalue was optimal if a and b are as just defined and

$$U = \frac{\langle Ay_{n+1}, y_{n+1} \rangle - \frac{1}{T_n^2(y)} \langle Ay_1, y_1 \rangle}{\langle (A+B)y_{n+1}, y_{n+1} \rangle - \frac{1}{T_n^2(y)} \langle (A+B)y_1, y_1 \rangle}$$

where $y = \frac{a + b}{b - a}$.

This result unfortunately has no practical application in the algorithm developed since it is assumed that nothing is known about the eigenvalue. Thus a and b cannot be chosen as above.

It was shown that for some special cases the elements of the auxiliary matrix defined by Stone's symmetric factorization [1] approach limits as the dimension of the matrix increases.

(M. A. Diamond)

[1] Stone, H. L., Private Communication, April, 1969.

7. THEORY OF DIGITAL COMPUTER ARITHMETIC

(Supported in part by the National Science Foundation under Grant No. US NSF GJ 813.)

7.1 Evaluation of Some Elementary Functions in Radix 16

During the quarter we have studied basic aspects of radix 16 iterative procedures, based on a continued product formulation, in order to determine the efficiency of this approach. Our interest has been restricted to four operations, namely division, multiplication, natural logarithm and exponential. It has been previously shown [1] that the computational procedures based on use of a continued product representation offer compatible algorithms with similar hardware requirements for elementary functions, making this computational approach very attractive in the view of new technological possibilities.

In a continued product form numbers are represented as

$$\prod_{k=0}^m (1 + r^{-k} s_k), \text{ where } r \text{ is the radix and } s_k \text{ is a properly chosen}$$

coefficient. In general, the hardware structure required for basic recursions in all algorithms under consideration, can be divided in three parts: adder complex, shifting network, and mechanism for selection of coefficients s_k . For radix 16 methods, the decrease of hardware requirements for the shifting network approximately equals the increase of adder complexity with respect to radix 2. Since the selection of s_k 's is sufficiently simple after the first 3 iterative steps, the increase in speed may justify implementation of radix 16 methods, provided that the selection of s_k 's during the initial steps does not require a complicated procedure. It has

[1] B. DeLugish, "A Class of Algorithms for Automatic Evaluation of Certain Elementary Functions in a Binary Computer", University of Illinois, DCS Report No. 399, (Ph.D. Thesis), June 1, 1970.

been shown that starting selection rules can be implemented without any substantial increase in hardware. The requirement for ROM capacity for the set of precomputed constants which are needed in algorithms for natural logarithm and exponential, is increased 3 times compared to radix 2 methods. This is still acceptable the required size of ROM being approximately $2.5(m + 5)$, words where m is the word length.

(M. D. Ercegovac)

7.2 Continued Fraction Arithmetic

During the quarter effort was directed toward the solution of the set of quadratics

$$x^2 + b_k x - c_k = (x-u)(x-v) = 0$$

such that $\frac{1}{2} \leq u \leq 1$. The problem, specifically is, given b_k and c_k , find

u . The method consists of expanding u in the form of a periodic continued fraction and gradually increasing the period of the fraction.

In an earlier report^{*}, v was restricted to a small range and the square rooting problem was only soluble for $c_k = 0$. On reflection it was seen that such a restriction on v is not necessary and hence a subset of the set of quadratics, namely all those with $b_k = 0$, (that is, the square rooting problem) was made the target of investigation.

With the given limits on u the triangular region enclosed by $c_k - b_k = 1$ and $c_k - \frac{1}{2} b_k = \frac{1}{4}$ was the region of the c_k, b_k plane that is amenable to solution by this method.

^{*}Quarterly Progress Report, Department of Computer Science, University of Illinois, June-September, 1970, pp. 132-137.

The recursion is given by

$$b_{k-1} = q_1 c_k$$

$$c_{k-1} = 1 - q_1 (b_{k-1} - b_k)$$

$$b_{k-n} = q_n c_{k-n+1} - q_{n-1} c_{k-n+2} + b_{k-n+2}$$

$$c_{k-n} = -q_n b_{k-n} + q_n b_{k-n+1} + c_{k-n+2}$$

$$P_n = q_n P_{n-1} + P_{n-2}$$

$$Q_n = q_n Q_{n-1} + Q_{n-2}.$$

The problem then is the selection rules for partial denominators q_i .

With $q_i \in \{\frac{1}{2}, 1\}$, there was no freedom of choice of q_1 and the dividing line had an intercept of $2/3$ (infinite binary fraction) on the $b_k = 0$ axis and hence redundancy in the digit set was required.

A choice of $q_i \in \{\frac{1}{4}, \frac{1}{2}, 1\}$ was made. The region of c_k, b_k plane soluble is now enclosed by $c_k - 1.56 b_k = (1.56)^2$ and

$c_k - 0.39 b_k = (0.39)^2$. In particular this means that the square rooting problem can be solved for $0.15 < c_k < 2.4$.

A set of selection rules for the choice of partial denominators has been developed, with extensive numerical simulations indicating their validity. A complete proof of convergence of the algorithm with these selection rules is the future aim of the investigation.

(Kishor Trivedi)

8. SWITCHING THEORY AND LOGICAL DESIGN

(Supported in part by the National Science Foundation under Grant Number U.S. NSF-GJ-503.)

When the branch-and bound method is applied to logical design, the method generates feasible solutions successively (the last one is an optimum network). We have been trying to improve the efficiency of the method, by examining whether each feasible solution contains redundant gates or interconnections. If they do, we can skip intermediate feasible solutions, shortening computation time. Preparation of a computer program to draw networks of feasible solutions facilitates discovery of procedures for this redundancy check.

Integrated circuitry provides logical designers with extra means such as wired logic. We started to look at the logical design with wired logic capability.

Ph.D. thesis research by T. K. Liu has been in good progress. Logical design with MOS is explored.

S. MUROGA

Our program for drawing networks of gates by Calcomp was extended to include NAND, EXCLUSIVE-OR, and EXCLUSIVE-NOR gates. Also, the number of allowable external variables was increased from 5 to 9. A user's manual has been prepared to allow the use of the program by other researchers.

Network transformations were studied with the idea of transforming (within a reasonable time) any network which realizes a given function into an optimal or near optimal network which realizes the same function.

Only minor results have been obtained thus far.

(J. Culliney)

There are several logic families of integrated circuits which have capabilities of wired logic. A NOR gate with wired OR capability (or a NAND gate with wired AND capability) is widely used because it is realized by DTL, some kind of TTL and ECL. By utilizing the wired logic capability, the number of gates required to realize a given logic function can be reduced. Properties of a network using this kind of logic gates were obtained. By integer programming we could obtain all minimal networks of three-variable functions using NOR gates with wired OR capability (all minimal networks using NAND gates with wired AND capability are obtained by a simple transformation). This kind of minimal networks were not known before.

(Y. Kambayaski)

An interface program between the program ILLOD(NOR-B) and Jay Culliney's CALCOMP program (see Quarterly Report for Jan., Feb., March, 1971 by J. Culliney) was developed and debugged. Using the solutions on magnetic tape obtained by ILLOD(NOR-B) as the input data, this program can draw the corresponding network of each solution. The user of this program can specify the problem numbers or the solution numbers under a problem number. In the former case, all solutions of those problems will be drawn; and in the latter case only the specified solutions of that problem will be drawn.

(H.C. Lai)

How to realize a negative function by a single MOS cell which consists of the minimum number of FET's has been investigated. This problem is similar to the classical relay network synthesis problem. However more restrictions can be added due to some properties of the MOS cells. If bridge connections are prohibited in the cells, this problem can be formulated into a tree type AND/OR gate network, and solved by the branch-and-bound method. The structure of optimum cells is being studied. It is hoped that some properties of the optimum cells can be incorporated with the branch-and-bound method in order to speed up its convergence.

(T.K. Liu)

Publications during this quarter

Nakagawa, T., and H.C. Lai, "A Branch-And-Bound Algorithm for Optimal NOR Networks (The Algorithm Description).

Nakagawa, T., and S. Muroga, "Comparison of the Implicit Enumeration Method and the Branch-And-Bound Method for Logical Design," Department of Computer Science, University of Illinois, Report No. 455, June, 1971, 94 pages.

Nakagawa, T., "A Branch-And-Bound Algorithm for Optimal AND-OR Networks (The Algorithm Description)," Department of Computer Science, University of Illinois, Report No. 462, June, 1971, 39 pages.

9. MACHINE AND SOFTWARE ORGANIZATION STUDIES

The following is a collection of related work aimed at improved designs for computer and software systems. We are interested in parallel processors, small primary memories, effective use of rotating memories and some questions concerning user languages for problems including typical Fortran type calculations and certain file processing problems.

9.1 Fortran Parallelism Detection (C. Cartegini, S. Chen, J. Han)

The function of the scanner has been refined. One conflict was the handling of variable names inside a DO-loop used simultaneously as a fixed subscript in an array-name. A similar conflict was the occurrence of the DO-loop index as a single variable name. Identical local names in decimal form have been provided. The scanner can now support more general Fortran programs with the exception of continuation cards.

For the purposes of a refined and more general analysis of Fortran programs the detection of IF-trees has become an important feature. The purpose of IF-tree detection is to handle them in a different way so that their occurrence does not mean an overall loss of efficiency. It may be recalled that the definition of an IF-tree supposes a high ratio of IF (logical, arithmetical) and GO TO statements to assignment statements. The new program which is being written should allow a more effective handling of the data so that retrieval will not cause the searching of large tables. The structure used for storing the data pertaining to each single block is a list-structure. The features for passing the structure elements from one procedure to another are being tested.

Some refinements of the DO Loop Subprogram have been made during this quarter. This includes a modification of the basic program and data structure in order to be able to handle more cases in general DO loops, such as the index variable used in an arithmetic expression within the same loop and any non-index variable used as a subscript. This reprogramming structure could lead to the possibility of extracting more parallelism within DO loops by backsubstituting the non-index subscript with its original form. For example, by transforming `<DO I=1,10; N=I+1; A(I)=A(N)+K>` to the form `<DO I=1,10; N=I+1; A(I)=A(I+1)+K>`, the parallelism which exists in the statement `A(I)=A(I+1)+K` can be exploited explicitly by the detecting algorithm employed in the subprogram. For this purpose, a subroutine which can do this simple first-level backsubstitution has been implemented and tested, and will be integrated into the main subprogram.

Also, some preliminary experiments on real application programs have been done by combining the DO Loop Subprogram with the Masterprogram and Assignment Statement Subprogram. During the experiment, some unexpected cases raised in a real program were found and debugged. A larger group of application programs taken from the University of Illinois subroutine library will be under experiment next.

For the complete analysis of DO loops, a further step is being taken to consider IF statements within DO loops. Some strategies have been considered. Since the detecting logic used in the reprogrammed subprogram could be compatible to this goal, the results are still expected to be feasible.

For assignment statements the implementation of the tree height reduction algorithm has been programmed in PL/1 and tested (see the last quarterly

report). In order to cooperate with the masterprogram and DO-loop subprogram to measure Fortran programs for parallel processing, some modifications are desirable such as omitting the building of syntatic trees for assignment statements to obtain less program execution time. Now a new PL/1 program has been programmed and tested. This program consists of manipulations of strings of numbers and related numbers. Those are minimum tree height, effective length, multiplicative length and absolute holes associated with each term in the assignment statement. They are required to compute the minimum tree height and minimum number of PE's required to execute the assignment statement within the time of the tree height. In addition to this, the number of operations for each level are counted in such a way that Hu's algorithm can be applied to obtain the minimum number of PE's for a tree graph within the time limit of the tree height.

For blocks of assignment statements the tree height reduction algorithm is being expanded to accommodate the back substitution and recursion algorithms properly to obtain a lower tree height and more parallelism.

By the application of back substitution the sequential relations among assignment statements may be changed into a parallel relation, i.e. they may be independent of one another and can be executed simultaneously in parallel machines.

By the application of recursion, iterative assignment statements in DO-loops might be handled as a set of independent assignment statements, rather than the iterations of sequential operations. Sometimes in this way we might gain the advantage for parallel operations.

9.2 Weighted Node, Directed, Acyclic Graph Schedule by m-Machines (P. Kraska)

We have developed a scheduling algorithm for m machines on a weighted node, directed, acyclic graph which is probably optimal, or near optimal, this quarter; the analysis is not yet complete. However, the following lemma was found which provides a lower bound on the number of machines required to process a graph in the critical time, W_q , of a graph G :

Let G_R be the relaxed graph of G (i.e., settle each node so that it is tightly connected to the terminal node, n_t) and label each level of independent nodes $1, 2, \dots, q$ where n_t is at level 1. Let $P(j)$ be the sum of the node weights (of independent nodes) at level j and W_j be the maximum longest path (i.e., the critical path) of node-weight sums of all nodes in level $q+1-j$ to the starting nodes. W_q is the critical path time of G_R (and G) to the terminal node, n_t .

Lemma: If $n < \max_k \frac{\sum_{i=1}^k P(q+1-i)}{W_i}$ then it is impossible to process all nodes with n machines in W_q time.

We now sketch the proposed scheduling algorithm. Suppose we have found m , the smallest number of machines required to process the graph in W_q time ($m > n$). For each node $n_i \in G$ calculate W_{it} , the largest node-sum on all paths from n_i to n_t the terminal node. Of the initial set of starting nodes assign up to m with the largest W_{it} to the machines; as time advances and k machines become free ($k \leq m$), update the set of starting nodes and again assign up to k of these with the largest W_{it} to

the machines. The rationale for the algorithm's optimality is that the time-dependent critical path is guaranteed to emanate from among the nodes being processed at all times, for a sufficiently large m .

Methods of factoring common subexpressions from tree-height reduced parses of arithmetic expressions, thereby transforming the parse-tree to a graph, were also investigated. An interesting fact was uncovered about the number of operations required when we factor common subexpressions from polynomials which have been parsed in accordance with Muraoka's folding method [1] which produces a tree-height near the limit of $1+\log_2[n]$. That is, we write

$$P_n(x) = a_0 + a_1x + a_2x^2 + (a_3 + a_4x)x^3 + (a_5 + a_6x + a_7x^2)x^5 + \dots$$

and we find that the number of operations required, y , is bounded above, i.e.,

$$y < 2n + \log_f(n) + 1,$$

where f is the Fibonacci ratio 1.618... We may write this more conveniently as $y < 2n + 4.79 \log(n) + 1$. This means, for example, that if $n=10^6$ then less than 30 operations are required in excess of $2n$, the minimum as prescribed by Horner's rule. (The folding method tree-height for $n=10^6$ is 30 compared with 21, the minimum.)

9.3 Simulation Processor (E. W. Davis)

The control unit, which resolves run time precedence questions, has been designed at the register level. A simulation of the control unit has been written in GPSS. This simulator will be used to study the performance of the unit on simulation programs being gathered from industry and university sources.

Organization of the processing units is being studied. Simulation programs have been found to consist of simple assignment statements intermixed with decision statements such that the ratio of IF's to assignments is approximately one. For this reason the design of the processors will include special logic for program control.

9.4 Memory Hierarchies (D. Gold)

Recent results indicate that any array operation may have its data mapped on the disk in such a way to yield no disk latency when accessing it. This is true even if only a subset of the original data is used and if the data is accessed differently on successive iterations.

The significance of the above is that for the array operations of Fortran-like programs, zero latency data mappings can be obtained for a small primary memory system. It is, however, still necessary to obtain a means of abstracting the relevant array index information from the original (source code) Fortran program.

9.5 Microprogramming and Control Unit Design (L. A. Hollaar)

The production version of the Burroughs microprogrammed machine was delivered during June. Work prior to this time consisted of preparing for its arrival. The indicator and switch panels were removed from prototype 3 and mounted in the new machine's rack; by switching the cables plugged into them, either machine can be used.

The individual boards of the new machine were tested by Burroughs, but the interconnections on the backplane were not. At the present time, the only wiring errors encountered were caused by human error or incorrect input to the automatic wire-wrap program; no mistakes by the wiring machine have been found.

The various input/output buses and control lines have been added to the backplane of the machine. Currently, the machine can be connected to four peripheral devices at one time.

Work has started on the connection of the microstore, which consists of four 1024 by 16 bit Cogar memories, to the control lines of the machine. The cycle time of the memory and the machine will be 400 nanoseconds (2.5 MHz).

After the microstore has been installed and tested, the various sections of the machine will be debugged, a core memory will be added, and the card reader and printer (currently in use by the PDP-11 project) will be interfaced. Following that, work on an automatic microstore loader, the disk interface, and a broadband link to the IBM System/360 is planned.

9.6 S-Language Assembler (E. Polley)

The string instructions for the assembler are in the process of being implemented into the language. Meanwhile test programs consisting of only word instructions are being run to locate any bugs in the code for those instructions.

In the final form the assembler will punch object decks (on the 360) which will be loaded into the D-machine for interpreting. Object code is currently being printed for each instruction but is not as yet punched. As soon as some longer programs are written in the S-Language timing estimates will be available on the speed of the assembler.

9.7 Text Searching (W. Stellhorn)

Implementation has begun on a preliminary version of the interactive language for text searching. The initial program, to be used for experiments with search and data organization techniques, timing requirements and user

reactions, will contain most of the features desired in later versions for controlling the search process and displaying the results.

The user will be allowed to request a search for any desired character string(s), and he may control the context of the search (full text, title, author, etc.). He may also specify that two or more strings should occur together within a given context (sentence, paragraph, article, title, etc.) or that documents which contain any of several strings or groups of strings are to be retrieved. Similar flexibility will be provided for controlling the output from a search. The user may request printing of any or all of the following: full text, selected bibliographic information, or the sentence or paragraph in which a "match" was found.

Background investigations are continuing in a number of pertinent areas including automated information retrieval, interactive languages and the production of digitized text.

Several items of B5500 software support for the D-machine have been received from the Burroughs Corporation and have been partially tested on local facilities and made available for use. These include a simulator for assistance in developing and testing microprograms, and several versions of the microcode assembler.

9.8 Debugging (M. Kaplan)

Implementation of the debugging package described in the last Quarterly Technical Report continued with the target date for implementation on the University of Illinois EXPRESS system as the beginning of the fall semester.

Version I Release 2 of WATFIV, which went on both the HASP and EXPRESS systems in June contains some new debugging features provided by Waterloo. These include a flow-of-control trace, the ON ERROR GO TO statement and DUMPLIST. (Details may be obtained from the consultants.) The first two of these are similar to statements in the proposed debugging package (LINETRACE and ON ERROR) and make the implementation of these unnecessary.

REFERENCES

- [1] Muraoka, Y., "Parallelism Exposure and Exploitation in Programs," Ph.D. Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 424 (February, 1971).

10. COMPUTER SYSTEMS ANALYSIS

(Supported in part by the National Science Foundation
under Grant No. US NSF GJ 28289.)

The goal of this research is the development of analytical tools for system modeling and analysis of real time computer networks. The particular network being investigated is that of a geographically distributed network of computers. A queueing theory model for this computing system based on the essential characteristics of the network, and priority assignment rules for efficient job processing at each of the computing centers of the network are being investigated.

10.1 Computer Network Modeling (W. Barr)

Our current effort in this area is aimed at developing a queueing theory model for a multiserver system with a finite length queue. To date we have considered nonpriority and nonpreemptive priority queue disciplines. We are currently investigating preemptive priority and dynamic priority queues.

Additionally, we are working to modify existing mathematical models of communication nets so that they more accurately represent our network. Cost functions for communication within the net are being developed. Once our model is completely defined we will look at criteria for inter-center load transmitting for optimization of throughput for the entire network. We also plan to investigate techniques for minimizing congestion within the network.

10.2 Center Throughput Analysis (J. Fitzgerald)

Our research in this area is aimed at analyzing Illinet, a geographically distributed computing center which provides online express, teletype timesharing, and remote batch entry services to a network of users at the University of Illinois. We are currently determining the essential characteristics of the various computers within the center.

We plan to employ GPSS (General Purpose Simulation System/360) in conjunction with our queueing theory model to facilitate studying the effects of priority assignment, job dispatching, and load regulation within the network.

(E. K. Bowdon)

11. SOUPAC AND STATISTICS

(Statistically Oriented Users Programming and Consulting)

April - June 1971

11.1 SOUPAC Consultants

A version of Soupac called Soupex is available on Express. This statistical system has the capability for:

- Means and Standard Deviations

- Pearson product moment correlations with optional missing data facility

- Principal axis factors and Varimax rotation

- One-way frequency counting

Soupex is limited to 30 variables and all the other limiting requirements of the Express System. It was felt users would appreciate the instant turnaround and simplified program cards of such a system.

A further use of express to codecheck a standard Soupac program deck is ready and will soon be available to users. The facility called Soupscan will check for keypunch and other errors and print diagnostics. Users may then correct their programs and run them with data (if any) on Hasp. Soupscan is a codechecking procedure and does not execute a Soupac program.

Soupac has been implemented at Cleveland State University in Ohio, at the National Headquarters of Girl Scouts and Megsystems Corporation in New York City and at the Bureau of Labor Statistics in Washington, D.C.

The Soupac Staff began maintaining a Quadratic Programming program from the Rand Corporation made available through the Agricultural Economics Department. A series of Multidimensional Scaling programs from the Psychology Department is also available through the Soupac Consultants.

During the period (April to June 1971) 13,451 jobs were run in Soupac. This is an average of over 4,460 jobs per month or nearly 150 jobs per day.

11.2 Research

Several goodness of fit measures for factor analysis were studied and applied to the results of maximum-likelihood factor analysis. In order to do this, 96 sample correlation matrices were drawn, twelve from each of eight population-correlation matrices with known factors. Each sample correlation matrix was factored by the maximum-likelihood technique, and the results were tested for goodness of fit. Each of the 96 matrices had 20 variables, and they were arranged in a $2 \times 2 \times 2 \times 3$ ANOVA design with three observations per cell. The four independent variables were number of factors (3 or 7), level of communality (high or low), model (whether or not the factor analytic model held in the population), and sample size (100, 400, and 1600).

Results showed that ρ_2 was a fairly good measure of the goodness of fit of the factor analytic model in the population. Although the sample values were somewhat large for small sample sizes, they approached the population values as the sample size increased. Two other measures, c_1 and r_1 , were used to compare the obtained sample factors with the population factors. These two measures can be used whenever a hypothesis factor matrix can be written down. They showed that fairly good agreement with an hypothesis can be obtained, even though the factor analytic model does not exactly hold in the population. This would require that the experimenter have variables with high communality and have a low ratio of factors to variables.

In general, better fit was obtained when the factor analytic model held exactly in the population, when there were three rather than seven factors, when the variables had high communality, and with larger sample sizes.

The above work was reported in DCS Report No. 451 An Investigation of the Goodness of Fit of the Maximum Likelihood Estimation Procedure in Factor Analysis. Some of the results are now being used in a comparison of several factor analytic techniques.

12. GENERAL DEPARTMENT INFORMATION

12.1 Personnel

The number of people associated with the Department in various capacities is given in the following table:

	<u>Full-time</u>	<u>Part-time</u>	<u>Full-time Equivalent</u>
Faculty	20	3	21.77
Visiting Faculty	4	0	4.00
Graduate Research Assistants	2	55	29.33
Graduate Teaching Assistants	0	28	13.11
Professional Personnel	9	1	9.50
Administrative and Clerical	19	0	19.00
Nonacademic Personnel (Monthly)	17	1	17.50
Nonacademic Personnel (Weekly)	<u>0</u>	<u>32</u>	<u>18.91</u>
TOTAL	71	120	133.12

*This report does not include personnel employed by the Computing Services Office.

The Department Advisory Committee consists of Professor J. N. Snyder, Head of the Department, Professors E. K. Bowdon, D. F. Cudia, K. W. Dickman, H. G. Friedman, C. W. Gear, D. B. Gillies, D. J. Kuck, B. H. McCormick, S. Muroga, T. A. Murrell, J. Nievergelt, J. R. Phillips, W. J. Poppelbaum, S. R. Ray, E. M. Reingold, J. E. Robertson, P. E. Saylor, D. L. Slotnick, and D. S. Watanabe.

12.2 Bibliography

During the second quarter, the following publications were issued by the Laboratory:

File Numbers

- (1) Borovec, Richard T., "IMAGE 8: The Real Time Clock," File No. 863, June 16, 1971

Report Numbers

- (1) ILLIAC IV Quarterly Progress Report, January, February, and March 1971, Report No. 456, April 15, 1971
- (2) Maruyama, Kiyoshi, "A Procedure for Detecting Intersections and Its Application," Report No. 449, May 1971
- (3) Michalski, R. S., "A Geometrical Model for the Synthesis of Interval Covers," Report No. 461, June 24, 1971
- (4) Michalski, R. S., and B. H. McCormick, "Interval Generalization of Switching Theory," Report No. 442, May 3, 1971
- (5) Nakagawa, Tomoyasu, and Hung-Chi Lai, "A Branch-and-Bound Algorithm for Optimal Nor Networks," (The Algorithm Description), Report No. 438, April 1971
- (6) Nakagawa, Tomoyasu, and Saburo Muroga, "Comparison of the Implicit Enumeration Method and the Branch-and-Bound Method for Logical Design," Report No. 455, June 1971
- (7) Ratliff, K., "Sparse Matrix Inversion," Report No. 443, June 1971
- (8) Reingold, Edward M., "Notes on AVL Trees," Report No. 441, May 10, 1971
- (9) Reingold, Edward M., "A Destructive List Copying Algorithm," Report No. 454, June 1971

Theses

- (1) Alsberg, Peter Allyn, "OSL/2: An Operating System Language," (Ph.D.), Report No. 460, June 10, 1971
- (2) Beals, Alan James, "The Automatic Generation of Deterministic Parsing Algorithms," (Ph.D.), Report No. 458, June 21, 1971
- (3) Bracha, Amnon, "A Symmetric Factorization Procedure for the Solution of Elliptic Boundary Value Problems," (Ph.D.) Report No. 440, April 30, 1971

12.2 Bibliography (cont.)

- (4) Garton, Richard Dean, "A Hardware Realization of a Decomposition Algorithm," (M.S.), Report No. 457, June 1971
- (5) Guimaraes, Celso John Frazao, "WEED: A Wonderful Equation Elimination Device," (M.S.), Report No. 444, June 1971
- (6) Henschen, Lawrence Joseph, "A Resolution Style Proof Procedure for Higher-Order Logic," (Ph.D.), Report No. 452, June 1971
- (7) Jacobson, Robert Clarence, "Path Integral Calculation of the Three-Particle Direct and Exchange Contributions to the Pair Distribution Function and of the Third Virial Coefficient for He^4 Gas," (Ph.D.), Report No. 445, June 1971
- (8) LaFrance, Jacques Emmett, "Syntax-Directed Error Recovery for Compilers," (Ph.D.), Report No. 459, June 21, 1971
- (9) Mark, Barbara Drahos, "Machine Independent Compilation of PL/I: Pass I Symbol Manipulation," (M.S.), Report No. 439, June 1971
- (10) Montanelli, Richard G., Jr., "In Investigation of the Goodness of Fit of the Maximum Likelihood Estimation Procedure in Factor Analysis," (Ph.D.), Report No. 451, June 1971
- (11) Rey, Christian A., "Control Point Strategy and Its Automated Diagnosis," (M.S.), Report No. 450, June 1971
- (12) Ryan, Lawrence D., "System and Circuit Design of the Transformatix Coefficient Processor and Output Data Channel," (Ph.D.), Report No. 435, June 1971
- (13) Simons, Arthur, "Eidolyzer: A Hardware Realization of Context-Guided Picture Interpretation," (Ph.D.), Report No. 448, June 1971
- (14) Wang, Paul Jang-Ching, "Machine-Independent Compilation of PL/I: Compilation," (M.S.), Report No. 453, June 1971
- (15) Yamada, Hirohide, "Emulation of Disc File Processor," (M.S.), Report No. 436, June 1971

12.3 Colloquia

"Probabilistic Generative Grammars," by Mr. Stephen Soule, Committee on Information Sciences, The University of Chicago, April 1, 1971.

"Formal Models for Memory Management Problems," by Professor Peter J. Denning, Department of Electrical Engineering, Princeton University, April 12, 1971.

"ATC (Air Travel Control) Terminal Automation," by Mr. Francis S. Carr, Department of Transportation, Federal Aviation Administration, Washington, D.C., April 19, 1971.

"The Posterior Intrinsic Systems of the Brain as a Jet Bundle," by Professor William C. Hoffman, Department of Mathematics, Oakland University, Rochester, Michigan, April 26, 1971.

"Lunar Surface Holography," by Dr. Donald H. Close, Hughes Research Laboratories, Malibu, California, May 3, 1971.

"Switching and Memory Effects in Amorphous Semiconductors," by Professor Hellmut Fritzsche, The James Franck Institute, The University of Chicago, May 10, 1971.

"Flow Analysis and Register Allocation," by Mr. Kenneth Kennedy, New York University, May 13, 1971.

"Information Work and Information Entropy," by Dr. R. R. Johnson, Vice President, Engineering, Burroughs Corporation, May 17, 1971.

"On the Algebraic Complexity of Matrix Multiplication," by Mr. Charles M. Fiduccia, Brown University, May 24, 1971.

"Models of Computation--Mathematical vs. 'Realistic' Properties," by Professor Edward Robertson, The University of Wisconsin, Computer Science Department, June 3, 1971.

12.4 Drafting

During the second quarter, a total of 465 were processed by the general departmental drafting section:

Large Drawings	122
Medium Drawings	91
Small Drawings	151
Layouts	3
Report Drawings	70
Changes	26
Miscellaneous	2
Total	<u>465</u>

(M. Goebel)

12.5 Shops' Production

Job orders processed and completed during the second quarter of 1971 are as follows:

	<u>AEC 2118</u>	<u>AEC 1469</u>	<u>Other</u>
Machine Shop	3	16	2
Electronic Shop	3	109	16
Chemical Shop	3	89	10
Layout Shop	0	72	8

(F. P. Serio)