C00-1469-0182
C00-2118-0007

RECEIVED BY DTIE AUG 3 1971

QUARTERLY TECHNICAL PROGRESS REPORT

January, February, March 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

January, February, March 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 61801

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Carl

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

TABLE OF CONTENTS

	Page
1. CIRCUIT RESEARCH	1
Summary	1
1.1 Bundle Processing	2
1.1.1 Bundle Signal Repeater and Restorer . .	2
1.1.2 Analog to Bundle Converter of Repeater	2
1.1.3 Repeater/Restorer Stages	2
1.1.4 SABUMA	6
1.2 APE	6
1.2.1 Output Decoding Circuits	6
1.2.2 The 1997/1023 Scaler	6
1.2.3 Machine Number to Signed Decimal Number Converter	6
1.3 Colormatrix	10
1.3.1 Achieving Thermal Stability	10
1.4 PENTECOST	15
1.4.1 Camera	15
1.4.2 Monitor	15
2. HARDWARE SYSTEMS RESEARCH	16
Summary	16
2.1 LASCOT	17
2.1.1 Light Source	17
2.2 OLFT	17
2.2.1 Design of Cooled Assembly	17
2.3 ORBIT	18
2.3.1 Modified Synchronization	18
2.4 Tricolor Cartograph	20
2.4.1 Control Panel Fabrication	20
2.4.2 New Logic	20
2.4.3 The Delay Line	20
2.4.4 Horizontal Video to Logic Converter . .	20
2.5 Transformatrix	23
2.5.1 Summary	23
2.6 BLAST	23
2.6.1 Screen Signal	23
2.7 Eidolyzer	26
2.7.1 Progress to Date	26
2.7.2 Hardware	28
2.8 Sematrix	31
2.8.1 Summary	31
2.9 LINDA	31
2.9.1 Summary	31
2.9.2 Project Status	31

	Page
2.10 RASER	37
2.10.1 Summary	37
2.11 Stereomatrix	37
2.11.1 Observer Position Detector	37
2.11.2 Coefficient Generator	37
2.12 PAGAN	51
2.12.1 Polyhedron Generation	51
2.12.2 Circuit Design	51
2.13 Scantrix.	59
 3. SOFTWARE SYSTEMS RESEARCH	 60
3.1 Numerical Processes	60
3.1.1 Sparse Matrix Inversion	61
3.1.2 Steady State Packages	65
3.1.3 Steady State Package Testing	66
3.2 Non-Numerical Packages	70
3.2.1 Compiler.	70
3.2.2 Filing System	70
3.2.3 Item Analysis	73
3.2.4 Global Analysis I and II	74
3.2.5 Elimination	78
3.3 Graphical Remote Access Support System (GRASS).	79
3.3.1 Disk Monitor System	79
3.3.2 Display Terminals	79
3.3.3 Information Retrieval Package	80
3.3.4 Disk Interface.	80
3.3.5 Monitors.	80
3.3.6 Remote Data Structure Utilities	83
3.3.7 Other Utility Software	86
3.4 Hardware	87
3.4.1 Computek Computer Graphics Terminal	87
3.4.2 PDP-8 System Engineering Log Summary	87
3.4.3 Disk	90
3.4.4 Multiplexor	90
3.4.5 Line Buffer	90
3.4.6 PDP-8/I, Stereomatrix Interface	90
 4. IMAGE PROCESSING AND PATTERN RECOGNITION RESEARCH:	
ILLIAC III	92
4.1 INTRODUCTION	92
4.2 PICTURE PROCESSING STRATEGIES	93
4.2.1 Show-and-Tell	93
4.2.1.1 System Objectives	93
4.2.1.2 System Elements	93
4.2.1.3 Documentation and Status	95
4.2.2 Scene Segmentation	98
4.2.3 Scanner/Monitor Evaluation	99
4.2.4 Scan-Display Devices	100

	Page
4.2.5 SMV Controller	101
4.2.6 Intermachine Link	101
4.2.7 Low-Speed Terminal Network	102
4.3 Parallel Processing Strategies	103
4.3.1 Synthesis of Interval Coverings for Pattern Recognition	103
4.3.2 Signal Detection Theory	103
4.3.3 Global Image Straightening	105
4.3.4 Pattern Articulation Unit	105
4.3.4.1 Iterative Array	105
4.3.4.2 Control	105
4.3.5 PAX II Language Support	105
4.4 Graph Transformation Strategies.	107
4.4.1 Graph Transformational Languages	107
4.4.2 Taxicrinic Processor	107
4.5 Applications	109
4.5.1 SEM Micrographs	109
4.5.2 Cytospectrometer	110
4.5.3 Pap Smears	111
4.5.4 Brain Mapping	111
4.6 Computer Systems Support	116
4.6.1 IBAL Assembler	116
4.6.2 Operating System	116
4.6.3 Arithmetic Units	116
4.6.4 I/O Processor	117
4.6.5 Channel Interface Units	117
4.6.6 Device Controller	117
4.6.7 Diagnostic Procedures	117
4.7 Documentation	119
4.7.1 External Documents Issued	119
4.7.2 Logic Drawings Issued	120
4.7.3 Engineering Drafting Report	120
4.8 Administration	121
4.8.1 Personnel Report	121
4.8.2 Computer Usage Log Summaries	122
5. NUMERICAL METHODS, COMPUTER ARITHMETIC AND ARTIFICIAL LANGUAGES	123
5.1 Computerized Mathematics	123
5.2 Problems in Computational Geometry	124
5.3 PDP-11	124
5.4 Factorization Methods Used in the Solution of Partial Differential Equations	125
5.5 Graph Algorithm Research	128
5.6 Stability Charts of Stiffly Stable Methods for Digital Simulation	129
6. SWITCHING THEORY AND LOGICAL DESIGN	130

	Page
7. SOUPAC SECTION	132
3. MATRIX MULTIPLICATION OPERATION MINIMIZATION	134
9. PARALLELISM EXTRACTION	135
9.1 Masterprogram	135
9.2 Assignment Statement Subprogram	136
9.3 Do Loop Subprogram	137
9.4 Tree-Height With Weighted Operands and Graph Schedules	139
9.5 Simulation Processor	141
9.6 Memory Hierarchies	142
9.7 Microprogramming and File Processing	143
9.8 Text-Searching Project	143
9.9 D-Machine Micro-Programs	144
9.10 S-Language Assembler	144
9.11 Debugging	146
10. COMPUTER SYSTEMS ANALYSIS	149
10.1 Computer Network Modeling	149
10.2 Center Throughput Analysis	149
11. ILLIAC IV	150
Report Summary	150
HARDWARE	152
11.1 Off-Line Diagnostics	152
11.1.1 PE Test	152
11.1.2 Card Test	152
11.2 Program Conversion	152
11.3 ILLIAC IV Maintenance	153
11.4 Financial	153
SOFTWARE	154
11.5 Operating System	154
11.5.1 Operating System I	154
11.6 Compilers and Translators	154
11.6.1 Glypnir	154
11.6.2 Cockroach	154
11.7 Assembler	155
11.8 Interactive Communications and Graphics . . .	155
11.8.1 Interactive Communications	155
11.8.2 Graphics	156
11.9 Mathematical Subroutines Special Function Library	156
11.10 B5500/B6500 Operations	157
11.11 Softward Documentation	157

	Page
APPLICATIONS	158
11.12 Numerical Analysis	158
11.12.1 Partial Differential Equations	158
11.12.1.1 Numerical Solution of Problems in Hydrodynamics.	158
11.12.1.2 Algorithm Development.	159
11.12.2 Matrix Inversion and Solution of Linear Algebraic Equations.	159
11.12.3 Eigenvalues	160
11.12.3.1 Eigenvalues and Eigen- values of Symmetric Tridiagonal Matrices	162
11.12.3.1.1 QR Algorithm with Origin Shift	162
11.12.3.1.2 Inverse Iteration to Find Eigen- vectors	163
11.12.4 Polynomial Root Finder.	164
11.12.5 Identification of Non-Linear Differential Equations : : :	165
11.12.6 Estimation and Filtering.	170
11.12.6.1 Non-linear Least Squares Problem	170
11.12.6.2 Numerical Solution of the Non-linear Matrix Riccati Equation	170
11.13 Linear Programming.	171
11.14 Long Codes.	172
11.15 Signal Processing	172
11.16 Education	173
11.16.1 ILLIAC IV Seminars	173
ADMINISTRATION.	176
11.17 Administration and Services	176
11.17.1 Financial Report.	176
REFERENCES, THESES, and DOCUMENTS	177
 12. GENERAL DEPARTMENT INFORMATION.	 179
12.1 Personnel	179
12.2 Bibliography.	180
12.3 Colloquia	182
12.4 Drafting	183
12.5 Shops' Production	183

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.)

Summary

Bernard Tse has taken over the work in Bundle Processing and describes in his report the combination "repeater/restorer". SABUMA, the earlier bundle project, is being finished off by Trevor Mudge. Yiu Wo's APE report deals with conversion between different number representations. Colormatrix is finally glowing: Hadjistavros has replaced the pellet resistors with microtransistors as heating elements, and he has also modified the method of thermally biasing and insulating the elements. Finally, Panigrahi comments on the camera and monitor specifications for Pentecost.

M. Faiman, editor

1.1 Bundle Processing (Project No. 21)

1.1.1 Bundle Signal Repeater and Restorer

Since the last quarter, it has been decided that, owing to their structural similarities, the bundle signal repeater and the bundle signal restorer should be incorporated into one system. The design of a system of 6 repeater/restorer stages with a bundle size of 63 wires is planned. The block diagram of such a system is shown in Figure 1.

1.1.2 Analog to Bundle Converter of Repeater

The analog input of the bundle signal repeater is to be converted to bundle representation. From this analog signal 63 wires with "+" and "-" signals have to be generated. A design for this A/B converter is shown in Figure 2.

1.1.3 Repeater/Restorer Stages

The mapping scheme used for the system is shown below:

	Repeater	Restorer
"+"	signal level "1"	signal level "1" in bundle 1 of input
"0"	broken wire, no signal	wire with level "0" in bundles 1 and 2 of input
"-"	signal level "0"	signal level "1" in bundle 2 of input

The purpose of the bundle signal repeater/restorer stages is to detect the wires with no signal information, namely the "0" wires, and assign to these wires "+" or "-" signals randomly. A design for this purpose is shown in Figure 3.

Bernard Tse

Figure 1. Block Diagram of Bundle Signal Repeater/Restorer

Figure 2. Analog to Bundle (A/B) Converter

Figure 3. Repeater/Restorer Stage for Two Wires A and B

1.1.4 SABUMA

All printed circuit cards for this project are complete except for the error detectors. The display has been wired but lacks a panel as yet. The only major construction remaining is to build the bundles. The system can then be put under test.

Trevor Mudge

1.2 APE (Project No. 25)

1.2.1 Output Decoding Circuits

In the APE machine, values of the input variables are normalized in the range of -1 to 1. Furthermore, all variables are mapped into machine number representation having a range from 0 to 1 in 10-bit binary accuracy. After they are processed by APEs, the results are sent out by the APEs in machine number representation. For easy read-out of the computation results, it becomes necessary to have the output value in machine representation converted back to signed decimal representation. This is done in the control unit of the APE machine by the 1997/1023 scaler together with the machine converter. These units will now be described.

1.2.2 The 1997/1023 Scaler

The output signals from the APEs are in pulse width modulation. The pulse width ranges from 0 to a maximum of 1023 system clock periods. The scaler is to map a number in this range into a number ranging from 0 to 1997, linearly, to the limit of the truncated least significant digit. The idea here is to convert the 10-bit machine number into a 3-digit decimal number plus sign. Figure 1 shows the block diagram of the scaler. What it does is simply to delete one pulse for every forty input pulses. A fast clock having a frequency twice as high as the system clock is used in the input of the scaler to compare with the pulse-width modulated signal from the output of the APEs. With such an arrangement the number of output pulses is that of the input scaled by a factor of 1997/1023.

1.2.3 Machine Number to Signed Decimal Number Converter

Recall that the mapping between these two representations is given by

$$x_i = 1 - 2x_{im} \quad \begin{matrix} -1 < x_i < 1 \\ 0 < x_{im} < 1 \end{matrix}$$

where x_i denotes variable value and x_{im} denotes machine number value.

$0 < X_{MD} < 1023T_S$; T_S = Period of System Clock

X_{MD} ; Machine Number in
Pulse Width Modulation

C_S = System Clock

$\bar{C}$ = A Carry Output; $\bar{C} = \begin{cases} 0 & \text{When the Content of the Counter is 39} \\ 1 & \text{Otherwise} \end{cases}$

Figure 1. The 1997/1023 Scaler

The function of the converter is to implement the above relation. This can be carried out by a circuit shown in Figure 2. The decade up-down counters are preset to 999, and the counter is set to count down to begin with. The input is fed by the 1997/1023 scaler. When the contents of the counter reach 000, the counters are switched to count up. If the counting process ends during the count down period, the number is negative, otherwise positive. The magnitude of the signed decimal number is given by the contents of the counters for both positive and negative cases. The full range of the input is 1997 pulses. This corresponds to an output range from -999 to +998. If the decimal point is considered located to the far left, the range of this number is suitable for displaying the variable value between -1 and +1 in 10-bit accuracy.

Yiu Wo

Figure 2. The Machine Number to Signed Decimal Converter

1.3 Colormatrix (Project No. 27)

1.3.1 Achieving Thermal Stability

A 5×7 matrix using microtransistors has now been constructed. X-Y addressing is used with the X line corresponding to the emitters, and the Y lines corresponding to the bases, as shown in Figure 1. The corresponding horizontal (X) and vertical (Y) drivers are shown in Figures 2 and 3. Due to the proximity of the transistors in the array, the thermal interaction between neighboring elements causes the active (and thus warmer) elements to heat the non-active elements. Moreover, this average heat, which is added over and above that supplied by the thermal bias, is dependent on the particular character being displayed. Clearly, this additional heat shifts the average temperature above the threshold temperature and makes the displayed characters ambiguous or, at best, destroys the contrast of the display. In the worst case, this shift, ΔT , is approximately $0.5 - 0.6^{\circ}\text{C}$. Thus, some sort of thermal isolation is necessary for the elements and, in addition, a sink must be provided for this excess heat. It has been determined that heat loss occurs primarily by two means: by conduction through the metallic leads of the transistors (about 10-20%), and by convection to the air (90-80%). Little can be done about the former, but an attempt was made to reduce the latter by introducing a screen consisting of a thermal insulator laminated to a thermal conductor. The thermally conducting layer was maintained at a much lower temperature than the bias temperature to act as a sink. The thermally insulating layer surrounded the transistors, so as to isolate them and, at the same time, not sink too much heat from them. Using this scheme an improvement of about 50% was noted, that is, ΔT was reduced to $0.2-0.3^{\circ}\text{C}$.

Thus, a new approach has been taken. In order to solve this problem, it will be necessary to operate the thermal bias at 1°C below threshold rather than at threshold. Thus, the average increase of $0.2-0.3^{\circ}\text{C}$ will be insufficient to bring the non-active elements into the thermally active region. This will, of course, be detrimental to the response time. To overcome this we may thermally bias at 1°C below threshold and then apply a pulse to all

Figure 1. Microtransistor Array

Figure 2. Horizontal Driver

Figure 3. Vertical Driver

elements which is sufficient to bring them up to threshold. When a character is displayed, the active elements receive an additional pulse, and the non-active ones receive no pulse at all. Thus, the active elements are driven up in temperature, and the inactive ones are allowed to "fall" down in temperature. In effect this is a "negative" heat pulse.

One could, of course, operate directly from 34°C by supplying a pulse to the active elements sufficient to bring them up the full 2°C while supplying no pulse to the inactive elements. This has disadvantages, however, in that it is more difficult to adjust the threshold temperature, and, as mentioned before, the response time is slower. Thus, the first method is preferable although more complex.

This method was implemented by dividing the output pulses from the SN7445 decoder (Figure 4), which formerly had a duration $2T$, into two pulses of duration T . Elements then receive none, one or two pulses, corresponding to the non-active, standby and active states.

In order to decrease the turn-on time, an additional group of horizontal drivers in Figure 4 (Group B) have been added in parallel with the normal drivers. These supply an additive power pulse during transition periods from one character to another. The duration of the period for which these additional drivers are active is adjustable. The results using this new scheme have been very good.

Stavros Hadjistavros

Figure 4. Revised Colormatrix Block Diagram

1.4 PENTECOST (Project No. 31)

1.4.1 Camera

The camera system of PENTECOST has been decided upon. For field-sequential color presentation, it is essential for the camera pick-up tube to have a high speed response, so that there is no carry over from one color field to the other. From cost and performance considerations, it was found that a lead-oxide vidicon (Plumbicon) could satisfy these requirements. Ordinarily, plumbicons do not have good spectral sensitivity in the red region; so we would buy a plumbicon designed to be used in the red channel of color television systems. Red-extended plumbicons were rejected because of cost considerations. The plumbicon tube will be employed in a black and white vidicon/plumbicon camera.

1.4.2 Monitor

The black and white CCTV monitor and the Penetron tube have been received. The various modifications of the video monitor so as to accommodate the Penetron tube are being considered now.

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.)

Summary

M. N. Cooper has concluded that a laser will be required for the LASCOT project, the efficiencies of other types of light source being too low. Progress in the design of the Curie temperature OLFT system is summarized by Doug Sand. Ed Carr's report deals with his redesign of the local oscillator in the ORBIT receiver to achieve synchronization with the transmitter. This project is now almost complete. Don Hanson has details on changes to the Tricolor Cartograph. The BLAST report of Larry Wallman describes problems of amplifying the screen signal. Art Simons has a description of how Eidolyzer's World Model works. The Mudge-Kodimer duo are now building some of the hardware for Sematrix, described last time, and have reached the stage of being able to locate block positions on the table. Dick Blandford's LINDA report deals mostly with his newly designed sweep circuits. Progress with Stereomatrix is written mostly by Dick Cheng, who gives detailed drawings of the Coefficient Generator and Cursor. Dick Partridge describes circuits and polyhedron generation for PAGAN. Finally, Sik Yuen is considering the use of LEDs in the Scantrix project.

M. Faiman, editor

2.1 LASCOT (Project No. 09)

2.1.1 Light Source

The requirement on the light source of LASCOT was for an output beam having maximum divergence of 0.1° and a minimum diameter not exceeding 2.5 mm. The source is required to put out a power of approximately 0.8 watt in blue (440 nm), 0.4 watt in green (510 nm) and 0.3 watt in red (630 nm). Preliminary calculations indicated that for an isotropic source of size s , producing an output beam diameter d with a divergence angle of Φ_3 radians

$$\frac{\text{total luminous flux in output beam}}{\text{total luminous flux emitted by source}} = \frac{1}{16} \left(\frac{d}{s} \cdot \Phi_3 \right)^2$$

substituting the following appropriate values: $d = 3$ mm, $s = 4$ mm and $\Phi_3 = 0.1^\circ$, this ratio is about 10^{-7} .

Thus the idea of using an isotropic (or nearly isotropic) source, such as a projection lamp or arc lamp and a collimator, was dropped. The use of a laser is the only logical choice.

Investigations are under way in selecting the most suitable laser.

M. N. Cooper

2.2 OLFT (Project No. 12)

2.2.1 Design of Cooled Assembly

Most of the design details for the vacuum chamber have been specified, and the materials required are being obtained. The modified electron gun is being ordered and the deflection-focus yoke has arrived. Remaining are the crystal assembly (with negotiations for its construction near completion) and the optical components. Design of the electronic circuits has begun with the temperature control circuits now being tested.

Discussions with other research groups working with cooled KDP devices indicate that near the Curie temperature crystal cracking sometimes occurs. To examine this problem and also to test the cooling design, an "intermediate" system is being built, which is essentially the new cooling assembly alone, contained in the old vacuum chamber and using

most of the old electronics. Further comments will be included in a subsequent report.

Doug Sand

2.3 ORBIT (Project No. 15)

2.3.1 Modified Synchronization

Until now, the ORBIT receiver has been synchronized with the transmitter by matching a local oscillator of 9.45MHz with the master oscillator. This has proved to be quite difficult. The reduced bandwidth of the system, compared with that of standard TV, made it necessary to count 600 periods of the local oscillator and compare the end of count with the arrival time of the next incoming video line. Any difference was fed back to correct the local oscillator frequency. The circuitry to implement this proved to require critical adjustment, with the result that the synchronization was not always true.

However, the exact value of 9.45MHz is only important for specifying the exact picture width -- an unnecessary parameter, since this width can be adjusted over a wide range on all receivers. The only real requirements on the local oscillator are that (i) it be no less than 9.45MHz to be able to provide 600 time slots in each horizontal line; (ii) its frequency be stable; and (iii) it start in the same phase on arrival of each incoming line of video. This task can be readily accomplished by a very stable, triggerable multivibrator.

The circuit that has been developed is shown in Figure 1, and is actually a retriggerable monostable. The CLEAR AND TRIGGER signal is derived from the synch waveform generator (see last report) and the four inverters provide the delay necessary for retriggering.

Ed Carr

Figure 1. ORBIT Receiver Clock

2.4 Tricolor Cartograph (Project No. 16)

2.4.1 Control Panel Fabrication

Part of the last quarter, was spent in the shop fabricating a suitable control panel out of sheet steel. The switches and slide potentiometers came and the switches are in the process of being assembled and mounted on the control panel. Control wiring was designed to run between the switch panel and the card rack.

2.4.2 New Logic

New logic was designed to make the new switches compatible with the previous logic. A gating circuit was designed and assembled on a PC card. This circuit is the minimal Moore designed circuit for gating one frame of video onto the disk. It is shown in Figure 1.

2.4.3 The Delay Line

The delay line from Corning arrived and was tested in the mode in which it will be used in the project. It seemed to work very well. As yet, no design has been finalized. It will be used in the vertical video to logic converter.

2.4.4 Horizontal Video to Logic Converter

Finally, a bandpass differentiator was designed and assembled on a PC card. This card is used in input of outlines from the black-board. It is shown in Figure 2.

Don Hanson

Figure 1. One Frame Gate

Figure 2. TCC Bandpass Differentiator

2.5 Transformatrix (Project No. 17)

2.5.1 Summary

The second phase of Transformatrix is now complete. Details are to be found in L. Ryan's Ph.D. thesis, "System and Circuit Design of the Transformatrix Coefficient Processor and Output Data Channel", to be published as D. C. L. Report No. 435.

M. Faiman (ed.)

2.6 BLAST (Project No. 19)

2.6.1 Screen Signal

In order to eliminate some of the noise associated with transmitting the very small screen signal down from 18kV, an amplifier has been built which floats at the anode potential. In this way, the signal from the screen is amplified before it is transmitted by the isolation transformer to ground. The common mode noise associated with the high voltage power supply is not amplified as much as the screen signal. The signal, which is obtained using the circuit in Figure 1, is approximately 5mV in amplitude. The electron beam has a constant current for this figure. The screen signal is not uniform in amplitude across a single horizontal line. This is thought to be due to variations in thickness or density of the phosphor coating on the screen. The variation, then, is because the beam does not penetrate the phosphor to the conducting stripes, but rather is a capacitive effect with the phosphor acting as dielectric.

In order to eliminate another source of noise, namely, secondary electrons falling back on the screen, the screen is biased about 10 volts negative with respect to the second or accelerating anode. With this biasing only about 10% of the emitted secondaries fall on the screen. The remaining 90% are collected by the second anode. ¹This biasing is accomplished by simply connecting second anode lead to the positive supply for the amplifier, as seen in Figure 1.

The amplifier, μ A733, outputs are connected to emitter followers whose emitters have windings of a transformer in series. In this way any voltage change at the output of the μ A733 is immediately transformed

¹Fundamentals of Television Engineering, Glenn M. Glasford, 1955, McGraw-Hill

Figure 1. Screen Signal Amplifier

into a current change in the winding and is thereby detected in the secondary of the transformer. The input resistor at pin 1 of the $\mu A733$ was determined experimentally.

The immediate goal is to amplify the signal more and to use this signal to synchronize the chopper.

Larry Wallman

2.7 Eidolyzer (Project No. 23)

2.7.1 Progress to Date

As suggested in the previous report, a test jig has been constructed to check out all of Eidolyzer with the exception of the input subsection. Using the test box to simulate the Scan Array, Color Detector, and Row Analyzer (see Figure 1), the remaining subsystems of Eidolyzer have been checked and work perfectly. Furthermore, the prototype card containing the simulated sections has been tested and approval has been given to build the remaining fifteen of these cards. To date, four more have also been built and checked out. The entire system should be constructed and functional during this next quarter.

Figure 1. Eidolyzer Block Diagram

2.7.2 Hardware Design

Having demonstrated that the system, with the exception of the input subsection, works as planned, it is appropriate at this time to describe in detail the most important part of the system -- the World Model.

The World Model is constructed with eight standard size printed circuit boards. (See Figure 2) In addition to timing signals, there are sixteen inputs to the World Model per region. There are four possible colors for a region, five possible colors for details, and seven significant combinations of temperature and humidity. The outputs consist of the name for a region from the following seven: SKY, WATER, ROCK, DIRT, SNOW, SAND, GRASS, or zero, one or two names for details from the following twelve:

WATER, SUN, BARN, UFO, GRASS, ISLE, ROCK, DIRT, CLOUD, ICE, SAND, SNOW.

The timing signals provide the necessary sequencing. By considering only certain information at certain times, it is possible to make a decision based solely on that information and modify it later due to newly-arrived information. This process can be repeated as many times as the decision-maker thinks it necessary. The more iterations, the more information considered, and the more accurate the answer. Depending on what accuracy is desired, the decision-maker is free to accept an answer at any point in the process. It is this very technique which is the realization of the layered, hierarchial type structure and which distinguishes the World Model from a simple read-only-memory.

There are four timing signals used in the World Model. The first pre-sets a bank of flip-flops corresponding to the presupposed knowledge of landscapes associated intrinsically with the World Model. This means that, with no other information available, a brown region is hypothesized to be DIRT, a white region to be SNOW, and so on. The second signal gates in the region color information so that the first interpretation can be made. The third timing signal gates in the detail color information. The details are now chosen by a logic array which allows the detail information to flow to a reasonable conclusion through paths biased favorably or negatively by the original region

WORLD MODEL FOR ONE REGION

Figure 2. World Model for One Region

hypothesis. For example, a white detail would tend towards SNOW, SAND, ICE or CLOUD but in a region already hypothesized to be SKY only the latter makes much sense. Furthermore, a CLOUD detail tends to increase the "confidence level" of a SKY region and would, in turn, provide some positive feedback to the first level decision. On the other hand, if a SAND detail were found in a region already hypothesized to be DIRT the confidence level would be decreased and a measure of negative feedback would be supplied to the first level decision. Finally, the fourth timing signal is used to add the temperature and humidity information to the decision-making process. This information is supplied by the human observer by means of switches located on the front panel. The four temperature switches read:

HOT, WARM, COOL, COLD

and the four humidity switches read:

ARID, DRY, DAMP, WET.

These switches serve a dual purpose. Their more important function is to provide another level of context for the interpretation, thereby exercising the world model further. A significant combination of temperature and humidity can influence the hypothesis of a region or detail directly, and these in turn can influence the other. Furthermore, the choice of a detail can influence the choice of another detail so that what we have is a fourth level of information capable of influencing third and second level decisions, with both these decisions then capable of influencing other third and second level decisions. The result of all this is a fluidic, nesting-type arrangement where the answer which finally filters through represents the most probable answer based on the presented information. The other function of the temperature and humidity switches is to reflect somewhat the state of mind of the observer. Having just come in out of the bitter cold, a person is perhaps more likely to see a white region as SNOW than a person who has been inside all day. By providing the observer with the opportunity to select a temperature and humidity condition, an environment closer to that of the observer is likely to result.

Art Simons

2.8 Semantrix (Project No. 24)

2.8.1 Summary

The position location scheme described in the last report has been designed. Presently, circuit boards for the sense amplifiers are under construction. A handwired system detecting a 6×6 coordinate matrix can be demonstrated. Construction of the major mechanical portions of Semantrix has begun.

Denny Kodimer
Trevor Mudge

2.9 LINDA (Project No. 28)

2.9.1 Summary

The goal of this project is to build a machine which is capable of recognizing a number of simple line drawings. A given line drawing will be displayed on a cathode ray tube via a flying spot scanner. The drawing will then be expanded so that its edges pass through a ring of photocells around the CRT. The information from the photocells will be clocked into registers and decoded to produce an identifying output.

2.9.2 Project Status

Past work has been focused on building a pattern generator and determining a suitable expansion technique. Two pattern generators were built and expansion of each pattern on the CRT was performed by lowering the accelerating voltage across the tube. Results were promising although intensity and focusing problems were encountered during expansion of the patterns.

In January 1971, a flying spot scanner was obtained and work during the past quarter has been directed at building sweep circuits, a one-stage video amplifier, and overcoming the intensity problem introduced by the scanning technique. Figure 1 shows the Synch. Separator Circuit. The vertical synch signal is obtained by integrating the composite signal, while the horizontal signal is obtained by differentiating the composite signal. Figures 2 and 3 show the Vertical and Horizontal Sweep Circuits, respectively. The multivibrators provide for vertical and horizontal sweep even though the synch signal is lost. (Note

Figure 1. Synch. Separator Circuit

Figure 2. Vertical Sweep Circuit

Figure 3. Horizontal Sweep Circuit

that similar circuits were used in Artrix.) Figure 4 shows the Video Amplifier which feeds a video signal to the cathode for intensity modulation.

Since a flying spot scanner is now being used in place of a pattern generator, it is necessary to raise the voltage across the CRT from 3 to 5kV. Preliminary tests indicate this will solve the intensity problem.

If further problems are not encountered, future work will be directed at determining an optimum decoding technique free of pattern orientation.

Dick Blandford

Figure 4. Video Amplifier

2.10 RASER (Project No. 29)

2.10.1 Summary

A method of expanding the display by a factor of two or four has been devised, using shift registers and data selectors. Full details will be given in the next report.

Tak Katoh

2.11 Stereomatrix (Project No. 30)

2.11.1 Observer Position Detector

The encoder disc code was programmed and drawn up and, after a 4 to 1 photoreduction, was etched on a .005" thick sheet of stainless steel. The shaft mount and mirror platform for the disc were also designed and fabricated. This subassembly awaits the reader mount for final testing.

An experimental stand was designed and built to test the reader electronics during encoder disc fabrication. Tests showed that the circuitry worked correctly. But they also indicated that the optic fiber diameter should be reduced in order to improve the resolution of adjacent binary numbers on the disc.

The encoder disc reader mounts have been designed and will be built next quarter.

Work to be completed during the next quarter will include design and construction of the detector cover and mount, fabrication of the necessary printed circuit boards for digital and analog circuitry, and interconnection of the subassemblies.

Chuck Pirnat

2.11.2 Coefficient Generator

All circuits except the master sequencer of the coefficient generator have been designed. The overall block diagram is shown in Figure 1. The 10-bit digital multiptier and multiplier control circuits are shown in Figures 2 and 3. Both of these have been tested. Completed during the past quarter are: (i) the translation control circuit, Figure 4, which enables the three-dimensional figure to be moved along x, y or z axes; (ii) the summing circuit, Figure 5, which adds/ subtracts the sine product to/from the cosine product; (iii) the sign detector, Figure 6; (iv) the coordinate selector, Figure 7, for the

Figure 1. Stereomatrix Coefficient Generator

Figure 2. 10-Bit Digital Multiplier

Figure 3. Multiplier Sequencer

Figure 4. Translation Control (A)

Figure 4 (Cont.). Translation Control (B)

Figure 5. Summing Unit

Figure 6. Coefficient Sign Generator

Figure 7. Coordinate Selector

operator's control panel. All circuits are being tested and are ready for fabrication. The most complicated part of the chassis wire wrapping which is the computing control section, has been completed.

Things remaining to be done are (i) memory loading; (ii) master sequencer designing; and (iii) wire wrapping of the rest of the system.

2.11.2 Cursor

All of the cursor circuits have been designed and more than half have been built. The system block diagram is shown in Figure 8. The most complicated subsystem is the analog matrix universe transformer which operates on a 4×4 matrix with $t_{41} = t_{42} = t_{43} = 0$ and $t_{44} = 1$; this is shown in Figure 9. The coincidence detector and circle generator circuits are shown in Figures 10 and 11.

Dick Cheng

Figure 8. Cursor Block Diagram

Figure 9. Analog Matrix Inverse Transformer

**COINCIDENCE DETECTOR
CURSOR, STEREOATRIX**

Figure 10. Cursor Coincidence Detector

Figure 11. Cursor Circle Generator

2.12 PAGAN (Project No. 32)

2.12.1 Polyhedron Generation

As described in the last quarterly report, PAGAN will generate polygons in cylindrical coordinates by appropriately varying the radius, R , as a function of the angle Θ . Figure 1 is a block diagram of polygon generation. Also discussed in the last quarterly report was the desirability of digital synchronization among internal waveforms. Therefore, Θ and Φ are derived digitally and converted to analog signals. A polar to rectangular transformation, computed using analog modules, yields x and y outputs.

If Z is incremented each time the polygon is outlined, a prism results. Furthermore, if r , the radius of the inscribed circle, is decremented when Z is incremented, then a pyramid is formed. Figure 2 shows the creation of solids having a regular polygon cross section. The proper increments and limits for each figure are determined by the digital Pattern Programming unit.

2.12.2 Circuit Design

At the end of the fourth quarter, a preliminary breadboarding of PAGAN was being used to evaluate D/A converters and operational amplifiers. During this past quarter, those items were ordered and have been delivered.

Also from the preliminary circuitry, three cards, shown in Figures 3, 4, 5 were submitted for printed circuit layout. Samples of the cards have recently been fabricated. These boards, as well as a couple of Stereomatrix designed cards, are used in PAGAN's polygon generation.

The multipliers and divider needed were ordered during the quarter. The divider module is actually a multiplier placed in the feedback loop of an operational amplifier circuit. Thus, frequency response and accuracy are inversely proportional to denominator magnitude. The divider module was chosen after considering the range of the divisor, $\cos \Phi$.

In the past quarter, the actual construction of PAGAN began. Card racks and power supplies were acquired and assembled into a cabinet. Wire lists have been drawn up and physical wiring of the back panel is about to begin.

Generation of a Regular Polygon of "S" Sides

Figure 1. Generation of a Regular Polygon of S Sides

Figure 2. Generation of Polyhedral Surfaces

1469 - 484
4 Bit Adder-Subtractor

Figure 3. PAGAN 4-Bit Adder - Subtractor

Figure 4. PAGAN Θ , Φ D/A Conversion

Generation of Polyhedral Surfaces

Figure 5. Sinusoidal Transconductor

2.12.3 Point Spacing

With accurate analog components and equal angle increments, there will be uniform point spacing on a circular cross section. However, as the circle is shaped into a polygon, the point spacing will no longer be truly uniform. Referring to Figure 6, if the angle increments remain equal the point spacing will vary corresponding to $\tan \Phi$. If the spacing is to remain uniform, the angle must be incremented in proportion to the arctan of that spacing.

Polar coordinates were decided upon because of the simplicity and generality in creating figures. Since an objective of the PAGAN project is building a relatively simple and inexpensive pattern generator, point spacing compensation will not be attempted. Although the nonuniform spacing may be evident in a triangle, it is inconspicuous in a polygon of five or more sides.

Richard L. Partridge

Figure 6. Point Spacing

2.13 Scantrix (Project No. 35)

More sophisticated methods of fabrication have made light emitting diodes cheaper and, if this trend continues, it is very likely that in a very short period of time there might be available diodes which cost about 30 cents each. This has encouraged us to give up the idea of a rotating mirror to scan the whole line of frozen TV signal and instead build a frame of 128×128 light emitting diodes with two alternating registers driving each line of diodes in sequence. One apparent advantage of this approach is that it eliminates the bulky and complicated structure of a four-foot long rotating mirror system while at the same time makes a flat display possible.

About six diodes of different makes and different characteristics are now under study. It is hoped that some circuits will be designed before September when the financial situation gets better and the project will no longer be delayed.

Sik Yuen

3. SOFTWARE SYSTEMS RESEARCH

All of the packages necessary for the transient analysis of arbitrary networks have been completed this quarter and work has just begun on linking them together.

A multiplexor enabling up to eight storage tube displays to be driven by the PDP-8/I has been completed and is operational. The new disk controller is now operational.

Publications this quarter include:

Gear, C. W. "The Automatic Integration of Ordinary Differential Equations," Communications of the ACM, 14, #3 (1971), pp. 185-190, along with submission to the Algorithm section of CACM).

Gear, C. W. "The Simultaneous Numerical Solution of Differential-Algebraic Equations," IEEE Transactions on Circuit Theory, TC-18, #1 (1971).

Gear, C. W. "A Graphical Search for Stiffly Stable Methods (with C. Dill)," Journal of the ACM, 18, #1 (1971), pp. 75-79.

Gear, C. W. "Transient and Steady State Numerical Solution of Differential-Algebraic Equations," IEEE Conference Proceedings, 1 (1971), pp. 384-386.

C. W. Gear, Professor
and Principal Investigator

3.1 Numerical Processes

The set of packages which performs the numerical and associated processing for transient analysis has been completed. They have been linked together, and linked to use the output from the compiler stage described in section 3.2.1. A rough flow of the whole system is shown in Figure 1.

The separate packages are discussed below.

3.1.1 Sparse Matrix Inversion (J. Deogun, K. Ratliff)

The group of routines which generates code to invert sparse matrices has been combined with the programs (MAIN and DIFSUB) which integrate ordinary differential equations. The function of the sparse programs is to generate matrix inversion code for the Jacobian of a system of equations. This result is used by DIFSUB in the corrector step of the integration process. The combination of routines has run successfully with equations used previously for test purposes. Figure 2 gives a diagram of the relationship between these two sections.

Within the sparse programs an improvement in the method of storing matrix element values has been implemented. If a matrix element is not an integer, its value must be stored in a floating point array (PW) of arbitrary length. These elements are either fixed constants or symbolic variables. If sparse is generating code to do arithmetic involving both integers and nonintegers, it must first store the integers as floating point constants. Rather than using a new location for each of these, the first twenty locations of PW are initially set to contain the floating point form of all integers with absolute value ≤ 10 (zero excluded). When one of these values is required, a subroutine is called to determine which location contains the appropriate constant and thus a new location need not be used.

The section of the sparse program which selects the order of the pivots has been changed experimentally in an attempt to find an optimal pivot selection algorithm. The basis for comparison of algorithms is the number of instructions generated for subroutines MATMUL and MATINV.

The following methods have been tested with the sparse program.

DEC PDP-8

Figure 1

Figure 2

1. Minimum Row--First Column: The row with the smallest number of non-zero elements is used as the pivot row. The first eligible element in this row is the pivot.
2. Minimum Row--Minimum Column: The row with minimum nonzero element count is pivot row. The element in that row with least nonzero element in its column is the pivot.
3. Minimum Product: The element with the minimum product (elements in row x elements in column) is the pivot.
4. Minimum Column--Minimum Row: Same as method 2 on the transpose of the matrix.

All these methods are dynamic in that at each pivot selection step, the choice depends only on that part of the updated matrix which is in unpivoted rows and columns.

Each method was tested with several examples. However, examples considered were comparatively smaller than those which are expected to occur in practice. Still many striking results and improvements in sparse were achieved.

Pivot selection method 4 was revealed to be the best algorithm of those tested. The word "best" is used here in a restricted sense and means more efficient in a larger number of examples. Method 2 was found to be equivalent to method 4 in efficiency. Method 3, minimum product, was the second best choice as an algorithm, but it must be noted that in two or three examples minimum product proved to be far better than the others.

The strategy of minimum row-first column proved to have very little value as far as efficient pivot selection is concerned. It was, however, invaluable in exposing bugs in sparse. Since this method led to the insertion of many new elements consecutively during the Gauss elimination, errors were detected that would have gone unnoticed in better algorithms which keep the number of new elements to a minimum.

Currently a larger example (50 x 50) is being tested with method 4 and it is felt that this example should be tested with the other algorithms as well.

3.1.2 Steady State Packages (B. van Melle)

The present package is a modification of the numerical integration package. While it works, we are not happy that it is as efficient as possible.

Tests were conducted with the numerical package in an effort to determine a criterion for selecting certain variables in DIFSUB in such a way as to bring about the swiftest convergence in the steady state problem. In particular, the relation between

$$D = \sum_{J=1}^N |DY(J)|$$

(where $DY(J)$ is the amount by which the J -th equation is not satisfied) and H (the step size) was examined, as well as the order of the method used.

The tests were conducted by running the problem up to a given order, and then varying the value of H . At each step, several values of H were tried, and the one which produced the greatest reduction in D was selected. This process was continued until D was sufficiently small.

The results for the various examples are given in the following tables. For a given order, the best step size and the accompanying reduction in D are listed for several values of D . The results for smaller values of D are identical to those for the last entry, since the method stabilized for small D . Best results were generally achieved with the low orders, since the method often failed to converge for the higher orders with reasonably large values of H .

The results indicate that the second-order method is good for D larger than .001, with a step size in the vicinity of 1.6, except for rather large D , which require a smaller step size. The linear method with step size of 1.0 is by far the best for small D . The transition from the second order to the first order method is indicated when D is somewhere in the range .01 to .001.

Further experiments are to be made to try to determine a good strategy. In the meantime work has continued on the use of the present package.

3.1.3 Steady State Package Testing (W. Chung, B. van Melle)

The following things have been done as an effort to combine and test the several routines which would constitute a numerical package as a whole.

1. Change of EPS.
2. Reconstruction of Main Program.
3. Test of CMPILE.
4. Test of SETUP.

1. Change of EPS

The error test criterion formula was changed from $EPS = 10^{(2.3*D - 4.3)}$ to $EPS = \exp(2.3*D - 4.3)$ which was shown by experiments to result in the faster convergence in most cases of the steady state problem.

The comparison of two EPS expressions is given in the table below for several examples (EX1 to EX4).

EPS=	10**(2.3*D-4.3)			EXP(2.3*D-4.3)		
	NS	NFNS	NW	NS	NFNS	NW
EX1	28	96	19	28	96	19
EX2	52	176	26	28	96	19
EX3	43	168	30	35	132	27
EX4	86	490	76	62	363	59

COMPARISON OF TWO EPS'S

The bound for EPS when D is greater than 1.0 was set to 0.1 since EPS = .05 gave slow convergence or no convergence in the new test problem.*

However, this EPS value remains a problem which needs more consideration.

2. Reconstruction of Main Program

The steady state solution of EX6 was obtained after 130 steps of DIFSUB call by using the old main and subroutines.** This was tried to check the subroutine CMPILE which was run together with other new subroutines.***

The first half of the main program was rewritten to match the new subroutine set and have the calling sequence in proper order.

*EXAMPLE 6

```
SUBROUTINE DIFFUN(T,G,DY,Y,YL,HINV)
REAL*8 T(1),G(1),DY(1),Y(7,1),YL(1),HINV
COMMON NFNS,NW
NFNS = 1 + NFNS
DO 1 I = 1,4
1  DY(1) = Y(2,1)*HINV + G(1)*(Y(1,1)-1.0) - (Y(1,1)-1.0)**2
DY(5) = Y(1,5)*(G(1)*T(3)/(T(3)+G(5))-Y(1,1)) + Y(1,6)
DY(6) = Y(1,6) - (Y(1,2)+Y(1,3)+Y(1,4)-Y(1,1))*0.500
DY(7) = 2.0*Y(1,7) - Y(1,7)**3 - Y(1,6) - T(2)
DY(8) = YL(1) - Y(1,7) - Y(1,6) + 1.000
DY(9) = YL(2) - YL(1) - (Y(1,1)+Y(1,3)+Y(1,4)-Y(1,2))*0.5 + 1.0
RETURN
END
```

```
SUBROUTINE S1(T,G)
REAL * 8 T(1),G(1)
G(5) = 1.0 + G(1)
RETURN
END
```

```
SUBROUTINE S2(T,G)
REAL*8 T(1),G(1)
T(2) = DEXP(-T(1))
T(3) = DEXP(-G(1)*T(1))
RETURN
END
```

** MATSET, DIFSUB, MATINV, MATMUL

*** DIFSUB, SETUP, SPARSE, COMPIL

EXAMPLE 1

$$F(x) = \begin{cases} \sin(x_1 x_2) - x_2/\pi - x_1 \\ (1 - \frac{1}{4}\pi) (\exp(2x_1) - e) + ex_2/\pi - 2ex_1 \end{cases}$$

D	BEST H FOR THE GIVEN ORDER AND THE ACCOMPANYING REDUCTION IN D						BEST ORDER FOR GIVEN VALUE OF D	
	1st ORDER		2nd ORDER		3rd ORDER		ORDER	H
	H*	RED	H	RED	H	RED		
1.0	1.1	.51	0.9	.37	0.9	.46	2	0.9
.3	2.0	.35	1.4	.06	1.3	.42	2	1.4
.1	4.0	.21	1.6	.04	1.3	.40	2	1.6
.03	4.0	.20	1.6	.03	1.3	.39	2	1.6
.01	4.0	.20	1.6	.03	1.1	.17	2	1.6
.003	1.0	.0007	1.8	.12	1.3	.04	1	1.0
.001	1.0	.0006	0.8	.60	1.3	.04	1	1.0
.0003	1.0	.0005	0.8	.60	1.3	.04	1	1.0

* H--the step size. Values tried for H ranged from .2 to 4.0
RED--reduction in D, which is equal to (new value of D)/(odd D)

EXAMPLE 2

$$F(x) = \begin{cases} x_1^2 - x_2 + 1 \\ x_1 - \cos(\pi x_2/2) \end{cases}$$

D	1		2		3		4		BEST ORDER	H
	H	RED	H	RED	H	RED	H	RED		
1.0	0.9	.46	1.6	.16	1.6	.24	0.5	.64	2	1.6
.3	1.1	.40	1.6	.15	1.8	.23	1.6	.11	4	1.6
.1	1.4	.36	1.6	.10	1.8	.25	1.8	.17	2	1.6
.03	2.0	.30	1.8	.06	1.6	.31	1.8	.14	2	1.8
.01	4.0	.17	1.6	.03	1.4	.35	1.8	.08	2	1.6
.003	4.0	.24	1.6	.02	0.9	.33	1.8	.03	2	1.6
.001	1.0	.002	1.6	.02	1.3	.05	1.8	.02	1	1.0
.0003	1.0	.0007	0.8	.60	1.3	.05	1.3	.39	1	1.0

EXAMPLE 3

$$F(x) = \begin{cases} 4 + x_1 + x_2 - x_1^2 + 2x_1x_2 + 3x_2^2 \\ 1 + 2x_1 - 3x_2 + x_1^2 + x_1x_2 - 2x_2^2 \end{cases}$$

D	1		2		3		4		BEST ORDER	H
	H	RED	H	RED	H	RED	H	RED		
2.0	0.9	.62	0.7	.52	0.7	.50	0.4	.77	3	0.7
1.0	1.8	.42	1.3	.12	0.6	.65	0.6	.65	2	1.3
.3	4.0	.21	1.4	.04	1.3	.43	1.3	.32	2	1.4
.1	4.0	.15	1.6	.05	1.3	.40	1.6	.08	2	1.6
.03	4.0	.20	1.6	.04	0.9	.33	1.6	.07	2	1.6
.01	1.0	.007	1.6	.03	1.1	.17	1.8	.05	1	1.0
.003	1.0	.007	2.0	.22	1.3	.04	2.0	.19	1	1.0
.001	1.0	.006	0.8	.60	1.3	.05	1.3	.40	1	1.0

EXAMPLE 4

$$F(x) = \begin{cases} x_1^2 + x_2^2 + x_3^2 - 5 \\ x_1 + x_2 - 1 \\ x_1 + x_3 - 3 \end{cases}$$

D	1		2		3		4		BEST ORDER	H
	H	RED	H	RED	H	RED	H	RED		
2.0	1.1	.41	1.3	.22	1.3	.30	1.6	.15	4	1.6
1.0	1.3	.39	1.8	.09	1.8	.27	1.4	.25	2	1.8
.3	1.8	.32	1.8	.04	1.6	.31	1.8	.14	2	1.8
.1	3.0	.22	1.6	.03	1.4	.35	1.8	.09	2	1.6
.03	4.0	.24	1.6	.01	1.4	.37	1.8	.03	2	1.6
.01	4.0	.21	1.6	.02	1.0	.26	1.8	.01	4	1.8
.003	1.0	.004	1.6	.02	1.3	.05	1.8	.02	1	1.0
.001	1.0	.003	0.8	.60	1.3	.05	1.3	.40	1	1.0
.0003	1.0	.003	0.8	.60	1.3	.05	1.3	.40	1	1.0

3. Test of CMPILE

To check whether CMPILE generates the correct codes for DIFFUN, S1 and S2, two programs were run, one using CMPILE and the other using hand coded subroutines DIFFUN, S1 and S2 without CMPILE. Through this some trivial errors in CMPILE were found and corrected, then finally two programs produced the same results. The fact that results of these programs were identical with that of previous program (old version) means that CMPILE, SETUP, and SPARSE are working nicely.

4. Test of SETUP

The SETUP written by K. Ratliff has been used through the above tests. Finally, another version of SETUP by A. Whaley was available and tested to see if it is working properly. They gave the same solution, however, after different numbers of function calls. (154 steps for the old SETUP and 168 for the new one.) The difference is caused by the different techniques used in SETUP to determine when a partial derivative is a constant. After some changes, the new set up produces a simpler matrix for the Jacobian.

3.2 Non-Numerical Packages (A. Whaley)

3.2.1 Compiler

The compiler, which has been working, was improved to generate more efficient object code. Both general registers and floating point registers are obtained for use from subroutines which allocate the register which is unused for the longest period of time. These changes will make the produced object code about 5/6 as long as that previously obtained. The former version only used one general register and one floating point register. The routine which handles the operators +, -, *, / is careful to take advantage of the possibility that some of the operands are already in registers. For + and *, if the second operand is in a register and the first is not, the sense of the operation is reversed (i.e., A+B to B+A) to allow the operation to be performed immediately. Intermediate results are never stored unless their register is required for some other operation.

3.2.2 Filing System

The filing system is fairly well debugged at this point and seems to be working satisfactorily. A command to

delete records was added. As the previous write-up was not very clear, and since that time most of the rules have changed, a hopefully complete write-up of the filing system follows:

```
XINIT <XLIST>
```

Here XLIST should be replaced with the name of an XLIST parameter area. In this special case, the XLIST is not used as a parameter list but as a scratch work area. This command allows a user to access the files residing in an OS dataset. The dataset is defined by a DD card with the name DISK. An example defining a temporary data set is as follows:

```
//DISK DD UNIT=DISK,SPACE=(TRK,10)
```

Additional parameters:

```
XINIT <XLIST> INIT,FORMAT
```

FORMAT will format the dataset for use, placing eight blocks on each track of 793 bytes each. INIT and FORMAT initialize the disk to have no files existing and all space free and available. Formatting is only required before the first time that a dataset is used.

After XINIT is complete, the user will discover in R1 the address of something called the XDSCB. This contains all the pointers, temp storage, etc., for the re-entrant filing system program. This address must be in every XLIST to be used. It may be placed there from register one as follows:

```
XLINK <XLIST> XDSCB=(R1)
```

It may also be obtained from another XLIST that already has this address:

```
XLINK <XLIST> LIST2=XLIST
```

```
OPEN <XLIST> NAME=  
      BUF=  
      NREC=  
      ABEND=  
      BUFLN=  
      TEST=
```

This command opens a file. Afterwards, until it is closed, it may be accessed by referring to the same <XLIST>. The NAME parameter is for specifying the file name. If the file did not previously exist, it will be created. Abnormal conditions for this macro are SEQ which is signaled if the file appears to already be open.

BUF is for specifying a buffer address. This parameter is not used for the `open`, but any parameter coded on any X macro will be there for execution of other X macros unless overwritten. (This statement applies only to BUF, NREC, BUFLen, and NAME.) BUFLen is the length of the buffer. NREC is the number of records to be read or written during XREAD or XWRITE. Severe abnormal conditions that are not tested will result in a branch to a location called ABEND (e.g. XYZ) the parameter ABEND=XYZ may be coded.

```
XREAD    <XLIST>,NREC=
          BUF=
          BUFLen=
          ABEND=
          TEST=
```

The first eight bytes of the buffer provide the line number of the first record to be read. Remaining records are read in a sequential fashion. Each record is placed one after the other and begins with the appropriate eight character line number and a two byte length of the remaining data. The source record format must be provided by the user when doing a write. Abnormal conditions which may occur are SHORT for a too-short buffer, RNF for a record not found, EOF for end of file, PARM for parameter error, and SEQ for sequence error (i.e., file not yet opened). For a more elaborate discussion of how to code the TEST parameter, see the previous write-up of the filing system. If abnormal conditions occur, <XLIST> + 34 contains the number of records successfully transferred.

```
XWRITE    <XLIST>,NREC=
          BUF=
          BUFLen=
          ABEND=
          TEST=
```

Write the number of records specified, each having a record name and remaining data length. Abnormal conditions that may be checked are: PARM, SEQ, and RF (record already exists). For explanations, see XREAD. Abnormal conditions cause the operation to be terminated. Number of records successfully transferred is in <XLIST> + 34 (halfword).

```
XDELETE    <XLIST>,NREC=
          BUF=
          BUFLen=
          ABEND=
          TEST=
```

Causes one record, whose name is in the first eight bytes of the buffer, to be deleted from the file. Abnormal conditions are RNF, record not found.

```
XDEST  <XLIST>,NAME=
        NREC=
        BUF=
        BUFLen=
        ABEND=
        TEST=
```

The <XLIST> named must not be associated with an open file. The file name by NAME= (here or previously) will be destroyed. Abnormal conditions are FNF (file not found) and SEQ (<XLIST> is being used with an open file).

```
<XCLOSE> <XLIST>,NREC=
        BUF=
        BUFLen=
        ABEND=
        TEST=
```

Closes open file associated with this <XLIST>. Abnormal conditions are SEQ if the file is not open. Also causes any blocks which have not yet been actually moved from core to disk to be so moved.

```
XTERM  <XLIST>
```

Undoes an XINIT: freemains control blocks used by the filing system, closes the OS dataset, etc.

3.2.3 Item Analysis (J. Koch)

The item analysis routine takes the data structures generated in the PDP-8 and produces a block which is used as input for Global Analysis. In Item Analysis such errors as illegal syntax in equations or declarations are detected and messages are sent to the user. Item was modified so it now eliminates from the node/connection table used by Global Analysis a terminal connected to itself, duplications of terminal connections and loops of connections. Item was also modified to be consistent with what is expected as input for Global Analysis. It uses the graphics filing system to save copies of the structures sent up from the PDP-8. These are retrieved

from the filing system when the command 'ITEMIZE name' is given. Thus, once an element definition has been sent from the PDP-8, it can be used without having to send it up with each network that contains it. The filing system is also used to save the output from Item Analysis in ITEM LIB where it is then called in by Global Analysis. Primitive elements and networks have now been created in the PDP-8, sent up to Item Analysis and passed on through Global Analysis successfully. Item Analysis uses a routine called PARSE to parse the equations. This routine has been modified to handle larger equations and will be able to handle floating point constants in the next quarter.

3.2.4 Global Analysis I and II (S. Wilkins)

A summary of the goals of the Global Analysis in its final form:

Figure 3 is a chart showing the position of Global Analysis I and II in the calling sequence of the Simulation and Modeling System. Global I can be activated by Item Analysis after storing data in the filing system or by the user whenever he wishes to analyze a previously defined network. Once Global I is activated no modifications to the user's data are allowed during the analysis. The only exception is that the user may supply different sets of values to variables global to the whole network, since these do not affect the structure of the network.

Figure 3

Figure 4

Figure 6

GLOBAL I

(Every network is stored as an element)
Each element that occurs in the network
need be analyzed only once.

The element is added to the Element Name Table and copied into the Element Area with certain modifications. All symbolic references to terminal types are replaced by internally assigned names. All numeric constants are placed in the constant table and are referenced by their position in the table. The element is assigned a numeric name.

This includes descriptions of subpictures and the construction of the network.

Each subpicture or element in the network must be analyzed and assigned an internal numeric name and then terminal types on all the connections must be checked for consistency.

The equation routine searches all equations on each subpicture for references to terminal variables and assigns to them internally defined names.

The equation routine is also used for all equations associated with the network.

Figure 5

Figure 4 is a flowchart of the routines comprising Global I. The function of 'Init' is to

1. Initialize a getmain area* in which various tables are built for use by Global II.
2. Setup the initial call to 'Element' with the name of the network to be analyzed.

'Element' is a recursive routine that reads element and terminal definitions from the filing system and builds tables in the getmain area. 'Element' operates on the network in a top-down manner.

Figure 6 is a flow chart of the routines in Global II. The function of Global II is to produce a set of equations equivalent to the network being analyzed. 'Init' initializes the getmain area* in which tables and equations for use by 'Elimination' are constructed, and sets up the call to 'Element.'

'Element' analyzes the network for equations in a topdown manner just as in Global I. Both 'Element' and 'Net' are recursive routines. 'Element' calls 'Net' when an element is a network and 'Net' calls 'Element' to analyze each subpicture element in the network. A detailed flow chart of Global II is given in File #824.

The main tasks of Element are

1. To assign internal names to global and internal variables and parameters.
2. To assign parameter values to the proper parameters.
3. To call 'Output' to modify and output the element and parameter equations.

The main tasks of 'Net' are

1. To assign unique terminal variable names at each node in the network using Kirchoff's current law to determine the assignment of I-type variables and Kirchoff's voltage law to determine the assignment of E-type variables.
2. To generate the equations necessary to show the relation between I-type variables at each node.
3. To call 'Output' to modify and output any network equations.

* The data structures constructed by Global I and II are described in detail in previous reports.

4. Call 'Element' for each subpicture in the network supplying the correct E- and I-variable assignments for the subpicture terminals and any parameter values.

3.2.5 Elimination (J. Frazao)

A package now called WEED has been written. WEED is an algorithm for symbolically simplifying systems of equations in treestructured form and classifying variables for the later numerical analysis stage. Through a consistent method of elimination, WEED produces a condensed system of equations and an Equivalence Table (EQV) for variables, which allow for a more efficient compilation and numerical integration than would otherwise be possible.

A number of passes are made through the system of equations, each time advantage being taken of simplifications made in previous passes. As each equation is processed, all expressions which are only functions of constants are evaluated and replaced by the result. Expressions of the form

$E+0$, $0+E$, $E*0$, $0*/E$, $E*/1$, $1*E$, $E+0$
 $E-1$, $1-E$ and $d/dt(S1 \text{ or } S2 \text{ expression})$

are simplified as they are detected, and unary minuses are eliminated, combined with other unary minuses or carried up the tree wherever possible.

Equations which become trivial relations, such as
 $\text{variable} = \text{constant}$

or

$\text{variable} = \pm \text{variable}$

are eliminated by setting appropriate pointers in the EQV, and the remaining equations are placed into three sets:

1. S1-equations, of the form:

$\text{variable} = \text{fn}(\text{global vars \& consts only})$

2. S2-equations, of the form:

$\text{variable} = \text{fn}(\text{globals, consts and TIME})$

3. General equations (those which are neither S1 or S2), of the form:

$\text{expression} = 0.$

In addition, as variables are encountered in the equations, they are classified into four sets according to their usage:

1. Set S1: variables defined by S1-equations.
2. Set S2: " " " S2-equations.
3. Set L : variables used only in linear algebraic expressions.
4. Set M : all other variables.

By performing these classifications, each set of variables and each set of equations can receive special treatment in the numerical integration phase, thus allowing for a more economical program execution.

The equations are reprocessed until there are no more changes in the EQV (up to a maximum of three passes). All variables are then renumbered sequentially, with all S1-variables first, then S2, L and M variables. The new names are substituted in the equations, and these in turn are compressed and passed on (in their three different sets), along with the EQV, to the numerical integration routines.

3.3 Graphical Remote Access Support System (GRASS)

3.3.1 Disk Monitor System (C. Hyde)

The new disk monitor on the new disk interface became operational this quarter and has performed very well. (c.f. section 3.4.3).

3.3.2 Display Terminals (R. Haskin)

ACID, the display terminal controller, has undergone several important modifications this quarter.

The first of these was changing IOT assignments to accommodate the terminal multiplexor. Sections of code which previously simulated this multiplexor were removed and replaced by the actual device IOT's. After the multiplexor hardware was tested, ACID was tried for the first time with two terminals, and worked perfectly.

Another modification was made to support the 8/I teletype as an operator's console. The function of the console

is to put the terminals on and off line so inactive terminals will not interfere with the system.

Strategy changes were made in the methods used to display pictures from the PDP-8. The most important of these was changing the program so that the keyboard is enabled for input between segments of display files rather than only after the entire display file has been received. This enables all pictures to be sent as segments, which eliminated the time spent waiting on completion of the last picture regeneration. The time it now takes to regenerate a reasonably complex picture (standard frame, about 20 nodes and 150 lines) is 1 to 1.5 seconds, which is considerably faster than previous expectations.

3.3.3 Information Retrieval Package (M. Michel)

The local IR package has been in use all quarter as part of GLASP and has run reliably. Copies of the program and documentation have been given to a research group at Griffiss Air Force Base, Rome, New York, to help them in development of their own graphics facility.

3.3.4 Disk Interface (C. Hyde)

The new disk interface became operational this quarter and has been put into full production use, both for the disk monitor system and for GLASP (c.f. section 3.4.3).

3.3.5 Monitors (M. Michel)

Multi-terminal support by both GLASP and GRASP is now operational (!).

GRASP (360 Remote Monitor)

The top-most monitor level, G8OPERAT, has been in use all quarter. All commands work as specified, and no abnormal terminations have occurred.

The second-level, multi-terminal monitor, SPACT, has also been in use all quarter. All commands work as specified, and two new ones have been added. '!DUMP' takes a snap of the current task attached to the requesting terminal; '!DSPACT' takes a snap of SPACT itself and the filing system module. A few bugs turned up at the beginning of the quarter. These were corrected and subsequently, SPACT has successfully handled a number of different user tasks on different terminals

simultaneously. These include the simulation and modeling system (specifically the module ITEM), and several utility packages (REPLYRE, COMUNE, LSD, etc., c.f. section 3.3.6). The user-level macros and routines for 2701 communication to the terminals (S8READ, S8WRITE, S8SETUP, etc.) have been in use successfully all quarter.

2701 Data Link

Usage increased again this quarter as debugging of user-written packages running under the monitor increased. Operation has been very good, although dense transmission of very long records has not occurred. One intermittent problem has begun lately (a premature time-out status at the 360 end) but has not happened often enough (yet) to be pin-pointed.

GLASP (PDP-8 Local Monitor)

GLASP has continued to be reliable this quarter, and no changes have been made. Multi-terminal operation was begun and worked perfectly the first time tried.

GLASP Service Routines

All services have been in heavy use this quarter except for Inktronic printing. No bugs have been found.

Additions to handle creation, integration, and display of subpictures, subpicture instances, and mnemonics were made, debugged, and put into production use during the quarter.

GLASP Program Segments

GENDRW was expanded with additional functions including basic subpicture handling, remote picture transmission, parameter assignment, etc. These new functions were provided mainly by making calls to GUT1 and GUT2 for access to "common" routines.

COMUNE was rewritten, expanded, and renamed REACT. It is now a very flexible tool for easy communication of commands and data structures to and from the remote CPU (c.f. section 3.3.6 on COMUNE and LSD). In general, data structures (pictures) and mnemonic definitions are transmitted to the 360 for archival storage. On request, these items can be variously

1. Fetched back to the PDP-8 and automatically stored on the local filing system.

2. Fetched back but not stored.
3. Displayed on the user's console.

Note that 1 and 2 cause the fetched item to replace the user's current working data structure in the PDP-8. After a fetch, the user can enter GENDRW and modify the item. But 3 only causes a display of the item, the current PDP-8 data structure remains unaltered. Any of 1, 2, or 3 are used in conjunction with LSD for examining, storing, and retrieving items from the remote permanent library. Mode 3 is mainly for use in conjunction with tasks like COMUNE for allowing access to the terminals in a "transparent" manner (e.g. the system appears as a terminal hooked to the remote CPU without the PDP-8 and PDP-8/I).

GLASP Status

Phase 0.5 has been reached: a skeletal system for stand-alone or remote interactive graphics support has been created. At this point, there is a great need for producing complete documentation for the elements of the system now implemented. In addition, although all aspects of this very initial version are operational, full testing and complete debugging has not yet taken place. While a rapid development and implementation of features occurred last quarter and this quarter, the coming quarter will be devoted to clean-up, consolidation, and debugging. Visibly moving above phase 0.5 will probably not occur until some time into the third quarter of this year.

Simulation and Modeling (SAM) System Support

GRASS is now being used actively for the development of the SAM System. Primitives and various networks using these primitives are created locally and transmitted directly to the 360 filing system (by LSD or ITEM) or are saved locally on magnetic tape for later transmission. ITEM is now running as an interactive user task in the 360, using the 2701 transmission facilities of the monitors for I/O to the terminals, using the 360 filing system for network (picture) and analysis storage, and calling the next level of SAM (Global).

A batch driver for ITEM was written and put in use this quarter, so that some types of debugging can be done from the system batch job stream.

A back-up tape of the 360 filing system is being maintained in case of system failure.

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

Data Structure Communications Package

COMUNE, the communications and data structure handling program, was debugged further this quarter, and several changes have been made, including the ability to send data structures to the filing system and to plot them. Also, several calling sequences to current routines have been changed.

Changes to Current Routines

INIT: INIT has been removed and all routines in the COMUNE package are self-initializing.

TEXT: CALL TEXT(IX,IY,'TEXT',N)

Text creates an entry in the text block consisting of the text string 'TEXT.' The first character starts at point (IX, IY) (in increments from the lower left hand corner of the picture). N is the number of characters in the string. If IX = IY = 0 and N is positive, the new text line appears immediately below the previous one. If IX = IY = 0 and N is negative, the new text line is concatenated with the previous one.

SEND: CALL SEND(IBLK,ICODE)

Send Block IBLK (IBLK = -1 means send all initialized blocks) to the PDP-8. ICODE is an INTEGER*4 array dimensioned 5 whose elements are as follows:

ICODE(1)--1 = reinitialize blocks when done sending
 0 = retain blocks

ICODE(2)--1 = replace block(s) sent in PDP-8
 data structure
 0 = retain current PDP-8 data structure

ICODE(3)--1 = do not display the block(s)
 0 = display them

ICODE(4)--1 = erase screen before displaying block(s)
 0 = add blocks to current screen image

ICODE(5)--1 = put up frame before displaying block(s)
 0 = do not put up frame

MASK: CALL MASK(IMASK)

IMASK is an integer array, dimensioned 11, whose elements are as follows:

IMASK(1)--1 = do not echo a user-typed text line on the display
0 = echo each user-typed text line

IMASK(2) through IMASK(10)--for screen segments 1 through 9
1 = accept joystick input
0 = ignore joystick input

IMASK(11)-1 = accept a user-typed text line
0 = ignore a user-typed text line

Hard Copy Output

Several modifications to the hard copy output routines have been made. The routines now take the data for the pictures to be plotted from the filing system, allowing a HASP job to be run to do the plotting off line. Programs in the graphics system can request hard copy output by first storing the data structure of the picture in the filing system, and then putting the name of the file containing the picture into file PLOTLIB in the filing system. COMUNE and LSD now have facilities in them to request hard copy output.

Graphics Library Maintenance Program

A program to provide facilities for storage and retrieval of data structures in the 360 filing system has been written and debugged. It allows pictures to be sent from the PDP-8 and saved, or fetched from the filing system and sent to the PDP-8. It also has provision for allowing pictures to be plotted by the hard copy output routines. This program, the Library Service Discographer (LSD), has the following facilities:

Saving of Items

SA[VE] <picture,name>
<mnemonicname>.<mnemonic id #>

Save allows a picture or mnemonic to be sent from the PDP-8 and saved in the filing system. The procedure for doing this is to first type in the save command, and then, after LSD replies 'SAVE READY', type in the PDP-8 command '##<picturename>' which will cause the picture to be sent to the 360. The 'mnemonic id number' is a one digit number from 0 to 7, which allows several versions of a mnemonic (such as horizontal and vertical resistors) to be stored under the same mnemonic name. The new picture or mnemonic replaces any previous occurrence in the filing system.

Fetching of Items

```
FE[TCH] <picturename>  
        <mnemonicname>.<mnemonic id>
```

Fetch retrieves the specified picture or mnemonic and sends it to the PDP-8, where it replaces the current picture in the data structure. LSD then replies with a message to indicate whether or not the picture was found.

Display of Items

```
DI[SPLAY] <picturename>  
          <mnemonicname>.<mnemonic id>
```

The action of DISPLAY is similar to that of FETCH except that the picture will be displayed at the terminal, and the current picture in the PDP-8 data structure will not be replaced.

Deletion of Files

```
DE[LETE] <picturename>  
         <mnemonicname>.<mnemonic id>
```

Deletes the picture or mnemonic instance in the library.

Plotting Files

```
PL[OT] <picturename>  
       <mnemonicname>.<mnemonic id>
```

PLOT puts the name of the item to be plotted into a file 'PLOTLIB' in the filing system. When the hard copy output program is run later from a HASP job, the item will be plotted.

Copying Files

```
CO[PY]  <picturename 1>
        <mnemonicname 1>.<mnemonic id 1>
        <picturename 2>
        <mnemonicname 2>.<mnemonic id 2>
```

The first item is copied into the second file, anything previously in the second file being overwritten.

Auto, Dauto

```
AU[TO]
DA[UTO]
```

The commands turn on and off the 'automatic save' feature in the PDP-8. When AUTO is in effect and a picture is sent to the PDP-8 via a FETCH or DISPLAY command, the picture is saved in the PDP-8's local IR. DAUTO prevents this from occurring.

Logging Out

```
GO[ODBYE]
```

Causes LSD to close all files and terminate. A message is sent to the user to inform him of the completion of this termination.

3.3.7 Other Utility Software

External GLASP Utilities

This quarter, several programs were written to facilitate loading and storing the GLASP system and its local library.

GLOAD: The GLASP system can now be easily loaded from a GLOAD tape, which has the program segments and the system programs, each with a program number. The load program, GLOAD, loads the 32 program segments from tape onto disk, then loads the core resident portions of GLASP. As an option, the local disk library can either be freshly initiated or restored as of the last GEXIT (see below) from a GLASP Library Tape. An additional option allows loading the PDP-8/I with ACID (the display controller) via the PDP-8/PDP-8/I data channel.

GEXIT: The local disk library may be saved on the GLASP Library Tape by bringing up the dectape system and calling GEXIT which saves the data and pointers for a subsequent GLOAD.

WLOD: The GLOAD tape may be updated with new assemblies from the PDP-8 disk monitor system by calling WLOD, specifying a system disk as input, and the GLASP program segment number as output.

PEEPER: The GLOAD tape may be directly modified by using PEEPER from the dectape system. The user specifies a program segment number, and then he may change any instruction in that program directly on the GLOAD tape. An option allows modification of any specified general tape block.

Computek Test Routines

An alignment and test program for the Computek CRT graphics terminals was written. It runs in the PDP-8/I, and has 17 looping displays for visual alignment, and four static displays for specific voltage measurements. The items checked are: loop time constant, offset, overall gain--horizontal and vertical, overall gain, velocity sensor, slow loop offset, fast loop offset, fast loop DC gain, overall offset, slow loop DC gain, time constant, and scope.

3.4 Hardware

3.4.1 Computek Computer Graphics Terminal (C. Carter, H. Lopeman, R. Miller)

STATUS:

University of Illinois P. O. #215711 (First Complete Unit)

Most of the problems have been remedied and is now considered operational.

University of Illinois P. O. #228151 (Operating Spares)

The 611 has been returned, checked out and considered operational and under warranty.

University of Illinois P. O. #233131 (Second Complete Unit)

The 611 has been repaired and warranty has begun on this unit. The terminal is considered operational.

We recently received from Computek many of the promised materials by Mr. Bob Cvitkovich. They are:

1. One extender card.
2. New service manual.
3. Spin-tight tool.
4. Sylvania IC information (in service manual).
5. Spare parts list (in service manual).

The slash thorough the zero (Ø) modification has been done in house simply by adding a 10K 1/4W resistor at P4 time to intensify the beam.

SPARES:

Of the 23 cards mentioned in the last quarterly report to be built as replacement cards, 22 of these have been built and checked out. The card to be built is card 'C'.

CONSTRUCTION:

1. Three cables have been wired and two checked out which interfaces the two terminals with the PDP-8/I multiplexor.
2. Rack has been wired for checkout of the 23 above mentioned cards and may be used for further design changes while the two terminals are in use.
3. The multiplexor has been checked out and is operational.
4. The Joystick interface (card 'f') has been built and checked out using a 'Mouse' for the joystick.
5. Curve generation is now being undertaken and card F has been modified, card I has had the drivers added, card J (containing the 'fast loop') is in the process of being built, and card K has a few modifications that need to be added.

3.4.2 PDP-8 System Engineering Log Summary

TTY

1. TTY grinding, very noisy. (Repaired)

DISK

1. Problems with Track 4, Sector 0. (Fixed)
2. Clock track failing on disk. (Pulled disk down and repaired clock track amplifier.)
3. Lost disk logic power supply. (Repaired broken bus.)
4. Disk track 13 not swapping. (Switching to Track 0.)

INKTRONIC

1. Inktronic dead. Pump screen clogged. Ink level in tank too high. (Cleaned and repaired.)
2. Inktronic paper feed dynamic braking control burned out. (Repaired)

PDP-8

1. Switches on PDP-8 register intermittant. (Replaced all known bad ones.)
2. Top DEctape transport has slow response (tape motion) problem is intermittant.

PDP-8/I

1. Several indicators burned out. Some drivers gone. (Replaced bulbs and driver transistors.)

338

1. 338 problem in deflection drive circuits (X) bad board in A19. (Located and replaced.)

COMPUTEK

1. #1 terminal had joystick problems. (Repaired)

3.4.3 Disk (C. Hyde)

The new disk interface became fully operational during the first weeks of the quarter. At that time, all users of the PDP-8 disk system reformatted their swap tapes. One minor error was discovered and eliminated in the disk monitor software.

One hardware error was detected and corrected during this period.

3.4.4 Multiplexor

The multiplexor for the Computek terminals was completed and tested. Several minor construction errors were corrected, and the unit is now operational with up to three device slots currently available. A report which describes the operation of the multiplexor is being written.

3.4.5 Line Buffer

The design of the Computek line buffer is about 90% completed. Construction and testing should begin as soon as a third terminal is available.

3.4.6 PDP-8/I, Stereomatrix Interface (I. Cunningham)

The design of the interface between the PDP-8/I and the stereomatrix display has been completed and all of the cards have been layed out. The back plane wiring has been started and one card is almost complete.

Transmission methods between the display and the PDP-8/I have been finalized. Control signals and picture coordinates are transmitted using SN55107 series line drivers and receivers. These are TTL compatible, fast, and insensitive to external noise. The cursor coordinates are transmitted by coaxial cable and converted to 10 bit words at the PDP-8/I. The coordinates exist internally in the display as analog values and placing the A/D's at the computer saves in cable costs.

The cursor for the stereomatrix is generated by the display logic. The coordinates are stored in a buffer in the PDP-8/I with each display frame and are available to the program at any time. Two interrupt flags are associated with the cursor. The first flag CURSOR indicates that the computer should use the present position for its next operation. The second flag, INCIDENT, results when the cursor approaches "close" to a line on the screen that is to be identified. The logic that generates the

display picture pauses when the INCIDENT flag is set. This permits the program to read the display memory address which identifies the line being drawn. Then picture generation can be resumed with no loss of data.

The interface design permits the addition of a programmer defined function box with up to ten buttons plus interrupt.

The hardware should be completed in the next quarter and the development of the software begun.

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)

4.1 INTRODUCTION

High lights of the work this past quarter include the following:

(1) Interactive picture processing using the Show-and-Tell System is now operable, if not complete. The current status of the Show-and-Tell System, the Scan-display Device status and Inter-machine Link is described in section 2.

(2) Signal Detection Theory is now being applied successfully to the local characterization of pictures. Extensions of this work called "Interval Coverings" promises significant contributions to scene segmentation. This work is described in section 3.

(3) The first half of Iterative Array of the Pattern Articulation Unit of Illiac III is now operational. The lights (all 512 of them) are "on again all over the world." See section 4.5.3.

(4) Image processing has been applied to Pap smears with limited success. See section 4.5.3

We are beginning to take seriously the responsibilities for maintaining the PAX II picture processing language on the IBM 360/75. Those who would like to join in establishing the PAX II Users Group should contact Val G. Tareski, ILLIAC III Project, Department of Computer Science, University of Illinois, Urbana, Illinois 61801.

B. H. McCormick

4.2 PICTURE PROCESSING STRATEGIES

4.2.1 Show-and-Tell

4.2.1.1 System Objectives

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

S&T 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 in the form of a means of calling IBM 360 programs and subroutines from the PDP/8e console teletype, and a set of subroutines for use by IBM 360 programs in transmitting data and pictures back and forth between the PDP/8e and the IBM 360.

Due to the presently ill-defined requirements for picture-processing software, it was deemed wise to implement a minimal system at first, expecting that additions and changes would naturally occur as work on applications progresses. To this end, the system is fairly modular and includes provision for loading parts of the system into the very restricted (8K) PDP/8e core as they are required so as to reduce the need for tight or highly finished code.

4.2.1.2 System Elements

Show-and-Tell is currently composed of the following PDP/8e resident subsystems and IBM 360 resident subsystems:

PDP/8e resident subsystems:

- Translator: Converts S&T language typed by the operator into an interpretable list structure. Enough information is contained in this internal coding to reconstruct the source statements on demand for listing and editing.

[1] Read, John S. DCS Report No. 429, "Show-and-Tell System Specifications," University of Illinois, Urbana, Illinois, March, 1971.

FIGURE 1. HARDWARE CONFIGURATION

- . Interpreter: Operates on the aforementioned list structure and on the data storage lists.
- . Interpretable Code Area (ICA):
Storage for the list structure produced by the Translator.
- . Executable Code Area (ECA):
Storage for programs executed directly by the PDP/8e hardware; i.e. the output of the PAL assembler.
- . Interpretable Code Loader: Loads interpretable code into the ICA.
- . EC Loader: Loads executable code into the ECA.
- . Supervisor: Provides first-level interrupts handling and controls loading of System Command Processors.
- . System Command Processors: Carries out system commands.

IBM 360 resident subsystems:

- . PAXDRIVR: Accepts commands and subroutine calls from the PDP/8e, reformats argument strings for FORTRAN compatibility, executes subroutines.
- . S&T/360 subroutine package: FORTRAN-callable subroutines to permit the IBM 360 to perform exchange of data with the PDP/8e. Also included are some S&T compatible versions of PAX II subroutines.

4.2.1.3 Documentation and Status

Progress is reported below under the subsystem headings described in the previous section. The reader is referred to reference [1] for a more complete picture of the context of this quarter's work.

Translator: A new version of the Translator was written and debugged. The new version is more systematic than the first one, and the addition of new functions is expected to be much easier. As of April, 1971, the following are acceptable to the Translator:

Labels

GOTO

CALL

INCR

DECR

ASGN

IFGO

EXIT

SCANM

SHOW

LOOK

Coordinate references (direct and indirect)

Integer references (direct and indirect)

Picture references (direct and indirect)

Relations

Signed integers

and Text literal (of length ≤ 6 characters)

Interpreter:

The following instructions are now obeyed by the
Interpreter, with restrictions as noted:

GOTO

CALL (to subroutine PAX₈ only)

EXIT

SCANM

SHOW

LOOK

A built-in S&T subroutine, PAX₈, has been implemented to
handle the interface with the IBM 360 resident subsystems. Functions
currently supported are:

Send a subroutine call to the 360

Send a picture to the IBM 360 from Illiac III core

Receive a picture from the IBM 360 and place it in Illiac III core

Type an error message from the IBM 360

Receive and carry out a command from the IBM 360 to scan a
picture with Illiac III core

Receive and carry out a command from the IBM 360 to display a picture residing in Illiac III core.

Send a command to the IBM 360 to delete the IBM 360 resident task.

Interpretable Code Loader and Executable Code Loader:

Only a small amount of time was spent this quarter in considering procedures for paging code in and out of the PDP/8e's 8K memory. This facility has not yet become necessary, although it is clear that it soon will.

Supervisor: A table-driven Supervisor has been implemented which handles TTY interrupts and routes control characters [such as those which control cursor movement] to the proper routines. The Supervisor also interprets system commands and calls the proper processor. The current version assumes that all the required control character routines and command processors are in core.

System Command Processors: The command processors listed below were operational by the end of this quarter:

GO	-Execute the program in the Interpretable Code Area
KL	-Terminate the IBM 360 task comprised of the IBM 360 resident subsystems
LD	-This function is being handled temporarily by the DEC system program loading function
SV	-This function is being handled temporarily by the DEC system program saving function
WK	-Wait for IBM 360 task to begin
XG	-Translate and execute each statement as it is entered
XL	-Translate statements and store in Interpretable Code Area for later execution.

PAXDRIVR: By the end of the quarter, PAXDRIVR was able to call a PAX II subroutine (or any subroutine following OS/360 conventions), and terminate itself on command from the PDP/8e. Legal subroutine arguments are text strings, integers and references to plane or stacks of planes.

S&T/360 Subroutine package: All of the subroutines listed in section 4.3.2 of reference [1] are implemented. Also, some new subroutines were added this quarter. Fuller documentation on the latter will be issued in the near future.

New Subroutines:

DISPLY	-Send a picture to the Illiac III and display it
SAVPIC	-Write picture on IBM 360 disk or tape
SHOW	-Display current contents of Illiac III core on monitor
XRANPI	-Generate a pseudo-random binary picture
XPRINT	-Print picture on IBM 360 printer
XPUNCH	-Punch picture on IBM 360 card punch.

John S. Read

4.2.2 Scene Segmentation

We have found a useful conceptualization of scene segmentation strategies as an extension of classical clustering techniques.

The levels of generalization allowed in this conceptualization may be described by three classes: classical unirelational clusterings, multirelational clustering, and relational inference.

Classical clustering techniques operate on a fixed graph whose elements are related by a symmetric similarity measure. Many current scene segmentation strategies are in this class.

Multirelational clustering allows different relations (e.g. colors) on a graph. The procedures can be visualized and implemented as the iterative application of basic graph transformations.

Relational inference allows us to form composite elements over a graph which has unspecified relations. The inference of unknown relations makes it plausible to generalize and find simplifying relational coverings.

Work is proceeding to express our previous results in this unifying framework.

John C. Schwebel

4.2.3 Scanner/Monitor Evaluation

PDP/8 machine language and FORTRAN programs are currently being written to calculate the performance parameters. Also, a revision has been made to standard film frames #1 and #2 to better compromise between efficiency and accuracy.

New Standard Film Frames

Walter Donovan

4.2.4 Scan-Display Devices

Final alignment was completed on the automatic video microscope. All drive motor problems have been solved--the motors are under manual control until the PDP-8e arrives to supply program 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.

The transport section of the flying spot automatic microscope has been moved to the outboard section of the 46 mm scanner frame. It is being readied to operate under the current 46 mm control (alternately with the current film scanner) in order to get the microscope scanner operating as soon as possible.

A second scanner control unit is being built-up in the 70 mm scanner frame. When it is operational, either the new controller or the current 46 mm unit will be altered for television digitizing.

One 46 mm film transport is being modified to take 35 mm sprocketed film, as many of the image samples being investigated are in this form. Losses in quality are incurred when the 35 mm film is reformatted to 46 mm.

A preliminary set of reference images for the scanners has been outlined by Donovan. These are being argued out and a standard set should be agreed upon and manufactured by the end of next quarter. There will be a set for each format (including high resolution television) which will allow systematic checks on resolution, distortion, and gray-scale parameters for each input device. Hopefully, these will eliminate qualitative bickering among users and engineers.

R. Amendola

4.2.5 SMV Controller

Four additional sections of the SMV manual are complete. Relevant logic diagrams and flow charts are complete to this point.

Work on construction of the second SMV Controller and the video-digitization circuitry was slowed this quarter in favor of putting emphasis on adding a microscope scanner to the existing controller. This was done to facilitate earlier testing of the rapidly-developing cytological image-processing work. [See Section 4.5].

Rough wiring to the microscope scanner is almost complete at this time (March) and will be completed early next month.

John Read
J. V. Wenta

4.2.6 Intermachine Link

While awaiting arrival of the PDP/8e work has continued on the definition of various interfaces. Among those nearing completion are:

- a) Interface to 360 via 2701 PDA: This is essentially an update of a similar interface constructed by AEC contract 1469 (Professor Gear). See DCS Report 372. This device requires a KD8/E single-cycle data break interface.
- b) Maintenance Processor/Exchange Net Interface: This is a flexible, high-speed interface to the Exchange Net which is capable of accessing any units in a variety of modes. Operation requires a KD8/E single-cycle data break interface.
- c) Real Time Clock: This device is a crystal-controlled clock suitable for use as an interval timer. The clock features a program selectable time base.
- d) Bus Drivers and Device Selectors: These are essentially copies of devices designed by contract 1469 to facilitate interfacing for programmed I/O devices. See DCS File No. 842.

Approximately 12 cards have been designed, wire-listed and are ready for wiring.

Richard T. Borovec

4.2.7 Low-Speed Terminal Network

Work in this area has been primarily directed toward resurrecting the LINC tape transports on hand. A transport has been operated under manual control with the most apparent problem being excessive noise in the rear bearings of the transport motors.

Some tests have been devised to check the condition of the transport heads and of the associated read/write electronics. In this regard, Charles E. Molnar of Washington University, St. Louis, was kind enough to mark several tapes on a PDP-12 adjusted for an absolute minimum of head skew. These tapes will allow proper adjustment of our magnetics.

Richard T. Borovec

4.3 PARALLEL PROCESSING STRATEGIES

4.3.1 Synthesis of Interval Coverings for Pattern Recognition:

In a discrete vector space E are given two subsets E^1 and E^0 . The subset E^1 consists of vectors (events) measured from a distinguished class of signals (e.g. texture of a picture segment, border between two picture segments, etc.) and the subset E^0 of vectors from the background. In general the intersection $E^\phi = E^1 \cap E^0$ is a non-empty set; and in this case we order the elements of E^ϕ with regard to ratio of frequencies of its occurrence in E^1 and E^0 , respectively.

The question considered is how to find an ordered set of filters in the form of multidimensional intervals for recognizing events from the signal class such that the individual filters correspond to the consecutive points on the optimal receiver-operating characteristic defined as in statistical decision theory. This class of filters can be viewed as an ordered covering of the subset E^1 . A synthesis procedure for constructing a quasi-optimal ordered covering, using the method of disjoint stars, is being prepared as a report and for conference talks.

B. H. McCormick
R. S. Michalski

4.3.2 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 '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.

We select a 2×2 window with 4 gray levels which defines 256 patterns. Each different pattern we call an event.

Given two textures T_1 and T_0 , we scan these areas to extract 'events' and divide them into three disjoint sets F_1 , $FMIX$, and F_0 . F_1 and F_0 consist of events which have occurred exclusively in T_1 or T_0 , whereas $FMIX$ consists of events which have occurred in both of these areas ordered according to the likelihood ratio ($\#$ of occurrences in T_1 / $\#$ of occurrences in T_0).

By selecting a suitable λ we can extract a set E of events which are more likely to occur in T_1 as compared with T_0 .

Extraction of different textural regions in a given scene:

Given n proto-samples of textures ($t_1, t_2, \dots, t_n$) which might occur in a scene, we will extract n sets of events $E_1, E_2, \dots, E_n$ using the R-O-C technique by successively comparing t_i with union of other samples ($\cup t_j$) $j \neq i$ for all $i = 1, \dots, n$.

After obtaining E_i , we extract events from different spatial regions in a given scene of analysis and mark those regions accordingly depending on to which particular set E_i the events belong.

Satisfactory results were obtained when this technique was applied to mark the nucleus, cytoplasm and background in a picture of brain cells.

S.N. Jayaramamurthy

4.3.3 Global Image Straightening

Fortran IV coding of the preliminary version of the curvilinear transformation algorithm was continued, but it has not yet been completed since other tasks dominated.

V. Tareski

4.3.4 Pattern Articulation Unit

4.3.4.1 Iterative Array

The principal work on the IA this past quarter has been to bring section 3 (one-half of the IA) to the level of section 1 (the currently operating half of the IA). All wiring has been completed except for the control/border wiring and the IA/TM interconnections.

4.3.4.2 Control

The design of the Phase 0 has been completed. Also the instruction TOPOLOGY has been implemented.

The GATEIA instruction has been operational during the past quarter. The results of tests indicate that a PLANE WRITE signal of about 500 ns. is sufficient to insure proper operation and therefore, a GOP (Gate Operation) time of 1 μ sec. is reasonable.

Since all the "dirty details of implementation" (i.e. fan-in and fan-out logic, etc.) have been completed, design will continue on an instruction by instruction basis. Approximately 15 instructions are flow charted awaiting implementation.

R. T. Borovec

4.3.5 PAX II Language Support

The following additions and changes have been made to the IBM 360 version of the PAX II Picture Processing System: two new FORTRAN subroutines, DCONV and LBLNUM, and one new assembly language routine, DATE, have been added. IBM 360 versions of the

following UNIVAC 1108 PAX Subroutines have been added: ABSUBT, CHECKW, FINWD, GLHS, GRIDW, LABEL, LINE, LSTCOM, MULT, MULTC, NAND, NANDS, NOR, NORS, PTHS, ROT90, SETSIZ, SUBSIN, and XREFL. Errors in FORTRAN subroutine PRINT and assembly language routine FP were corrected. Subroutines READZ, WRITEP, and WRITEZ were changed to utilize routine KPBS. Assembly language routines KGC and KPC were rewritten to reduce their execution times.

One more "PAX Memo" has been sent to University of Illinois PAX users to describe the latest version of the PAX II Subroutine Library, PAXSLIB. A full description of IBM 360 PAX memory requirements was included, as was documentation describing the new and added routines.

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.

V. Tareski

4.4 GRAPH TRANSFORMATION STRATEGIES

4.4.1 Graph Transformational Languages

We have specified a Root Graph Language, RGL, in order to be able to simply and conveniently express operations on graph networks which allow an arbitrary number of associated variables 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 network. Many scene segmentation algorithms can be expressed most simply by operations on a graph network. We have developed some theory for the next higher level of picture processing operations represented as graph network transformations.

Thus, the graph 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 was specified with the goal of embedding it in PL/I.

J. Schwebel

4.4.2 Taxicrinic Processor

Work has continued on Volume II of the TP Manual. It should be ready to publish early in the next quarter.

The card design and wiring of the integrated circuit portions of the TP control logic and register sections has progressed well. At the present time, about 102 final drawings have been completed representing about 70 IC boards. This does not include the inter-board connections. Of the boards whose drawings have been completed, about 57 have returned from the wiring contractor and the rest are in the process of being wired. Physical checkout of the boards has just been started.

During this quarter the first two volumes of the new edition of the Illiac III Programming Manual were completed. These two volumes have included the description of the computer system and the instruction repertoire. Volumes III and IV, which are currently in progress will include the input/output operations and the supervisor organization.

The logical simulation of the Taxicrinic Processor continued during the first part of this quarter. However, the size and complexity of the simulator has become such that it greatly taxes the capacity of the Digital Computer Laboratory's IBM 360 system as well as the single programmer working on it. As a result, work on the simulator was halted after about 1/3 of the basic machine control logic had been simulated. The sequences which have been programmed operate successfully. However, due to the various system problems which come up whenever a single program needs an exceptional proportion of the total system resources, it was felt that the time spent on the simulator could be more usefully spent in other phases of the taxicrinic processor design.

B. J. Nordmann, Jr.

4.5 APPLICATIONS

4.5.1 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, late 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 (see Section 4.5.2 below). 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, leukocytes, urine crystals, a few kinds of bacteria in different forms, chromosomes, etc., courtesy of Dr. Ben Williams.

(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/Pd alloy is a compromise to solve the problem. For a further improvement, we might try to treat the sample with OsO₄ vapor* to insure good image quality.

J. Chen

* Gerhard E. Pfefferkorn, "Specimen Preparation Techniques," Proceedings of 3rd. Annual Scanning Electron Microscope Symposium, 1970, p.91

4.5.2 Cytospectrometer

The main effort this quarter has been the design, construction and checkout of the droplet generator digital control. A brief description of the control follows.

A pulse source, currently an audio oscillator, is used as a master clock. The clock drives a switch-programmable frequency divider chain. The divider consists of a twelve-bit counter and associated reset logic.

The output pulse of this divider is used to drive both a high-voltage charge ring driver and a stroboscope driver. This pulse also drives a switch-programmable frequency divider chain consisting of an eight-bit counter.

The output pulse of this divider is used to drive the high voltage deflection plate drivers.

Independent adjustment of pulse width and delay are provided for the charge ring, strobe, and deflection plates.

The drivers for the charge ring and deflection plates are simple, single transistor stages. Output voltages may be set between 0 and ± 350 volts. The outputs are current-limited to reduce the possibility of serious electrical shocks.

The bimorph driver is also a single transistor device. Since the bimorph exhibits very large capacitance at high frequency, operation of this prototype driver is limited to about 40 KHz. The bimorph driving signal may be varied from 0 to 60 volts.

The entire control, less power supplies, is contained in a 10 3/4" rack panel. The twenty-five integrated circuits are mounted on an 8 x 10 inch wire wrap board to facilitate easy modifications.

R. T. Borovec

4.5.3 Pap Smears

To make automated or semi-automated measurements on large numbers of cells presented in the standard Pap-smear format, it is necessary to have an efficient searching procedure which can reliably locate cells and cell components in the presence of pictorial noise caused by cell clumping and overlapping.

This quarter, a Pap-smear searching program using Illiac III-type instructions was designed and partly implemented. The design and some preliminary results were presented at a biomathematics symposium in March. [see Section 4.7 this report]

The search procedures are based on the Illiac III's plane-parallel operations and were programmed and tested using the PAX II language interactively in conjunction with the Show-and-Tell programming system [see Section 4.2.1, this report].

Figure 1 is a microscope field showing a portion of a Pap smear magnified by a factor of 63 (on the 35-mm film). The currently-implemented routines operate on a coarse-resolution scan, as in figure 2, [128 x 128 pixels, 16 gray levels covering a field of approximately 225 μ x 350 μ on the specimen]. The programs label points in the picture as cell boundary, cell nucleus, leukocytes, background and clump. The label for each point is generated with parallel, Pattern Articulation Unit-type operations and stored in a plane or "map" having a bit set on for each picture point having the label. A given point may have more than one label; i.e., may be "on" in more than one plane.

Examples demonstrating this labelling are in figures 3 and 4. Figure 3 shows lines drawn by the program around points identified as clump points, that is, places where a texture was detected which indicated that a leukocyte clump might be present, but where individual cells could not be resolved. In figure 4, the program has removed points labelled as likely to be leukocytes. The "holes" were filled with background gray-scale values. Note that these displays were generated primarily to

Figure 1. Microscope view of Pap smear

Figure 2. Digitization of figure 1 (coarse resolution)

Figure 3. Clump points labeled

Figure 4. Leukocytes identified and removed

permit tracing of the program's progress. These pictures are not used per se in the processing. Information required in this way is used to direct rescanning of small areas at higher resolution, thereby pruning the search tree relatively rapidly. When objects are found which are known to be ambiguous at the coarse resolution, they can be re-examined with specific procedures to resolve specific ambiguities.

John S. Read

4.5.4 Brain Mapping

The purpose of this investigation is to develop inference techniques and to test and apply them in the environment of neuro-anatomy of the brain.

It is intended to apply these techniques first to a small subset of objects in the brain. In connection with current research performed in quantitative neuroanatomy*, methods are being developed for the automatic recognition of cellular and subcellular objects visible in cresylecht violet stain and Weil stain as far as they can be resolved under a light microscope. An immediate application is to determine quantitative properties such as the number of neurons in their specific domains of the brain and the geometrical dimensions of their nuclei and nucleoli. A further application is the delineation of the boundaries of larger domains in the brain, e.g. nuclei, 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

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

partitioning a picture into regions, we associate with each region applicable attributes and their values as well as binary relations valid between pairs of regions.

Region identifiers, attribute values, and binary relations constitute the input to the inference phase. Two strategies are being considered:

1. 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 that 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 such that the structure of the picture is described by a derivation tree whose nodes are associated with a region identified.
2. To generate appropriate grammars using covering and clustering techniques.

It is intended to make use of all clustering techniques that are available. The great diversity of objects constituting a brain (gross anatomy, histology, **neural** microanatomy, etc.) allows us to select specific methods of varying degrees of sophistication. The following techniques are being considered:

1. Graph theoretical clustering methods (see Section 4.4.1)
2. Covering techniques that are being developed in the framework of signal detection theory (see Section 4.3.1)
3. Probabilistic clustering
4. Grammatical inference

Peter Raulefs

4.6 COMPUTER SYSTEMS SUPPORT

4.6.1 IBAL Assembler

IBAL (Illiac III Basic Assembly Language) grammar was reviewed. CF productions were partly incomplete. Grammar for integer expressions and boolean expressions were modified to a PL/I-like structure. Segmentation and synchronization, which are important in file management, memory allocation and parallel processing, are still under consideration.

The work to write the parsing phase of the assembler, using Alan James Beal's "Translation Writing System," has started. The grammar has to be rewritten in TWINKLE which is an English-like-meta-language for writing grammars.

A revised IBAL Manual will be issued early in the next quarter.

A. Masumi

4.6.2 Operating System

The overall framework of the Illiac III Operating System, specifically the part which will be implemented in hardware, is almost completed. The necessity of knowing more about the environment in which this OS will be functioning has made us work in parallel on the assembler system and on the file management system which will be an integral part of Illiac III OS.

The consideration of IBAL as a general assembly system is leading to some programming convention for our system, which in turn will specify environmental requirements on the operating system.

A. Masumi

4.6.3 Arithmetic Units

Control logic drawings for the instructions: Add, Subtract, Compare Algebraic and Multiply were completed and proofed. Layout on PCB's is still being done and should be completed in the coming

quarter. Drawings for Division instruction were completed and are being proofed. Cross-referencing between AU processing hardware drawings is being continued.

L. N. Goyal

4.6.4 I/O Processor

The System Organization of the I/O Processor was delineated in Reference Manual, Vol. I^{*} issued this quarter. Vol. III which discusses in considerable detail the Command Repertoire of this processor, should be available for printing in the next quarter.

J. V. Wenta

4.6.5 Channel Interface Units

The wire count for the integrated circuit cards was completed.

Duplicate wire lists for these cards have been completed and corrected and made available to the vendor. Vendor has completed the wiring of one complete set of IC cards and promised delivery on April 2, 1971.

Listing of flow charts and completed logic diagrams for CIU manual is complete.

J. V. Wenta

4.6.6 Device Controller

There have been no new developments on this item during this quarter.

Val Tareski

4.6.7 Diagnostic Procedures:

The study undertaken on the subject "Control Point Strategy and Its Automatic Diagnosis" (Master Thesis to be published) is now in its final phase. During the last months, two points have been

^{*}McCormick, B.H., Nordmann, B.J., "ILLIAC III REFERENCE MANUAL, VOL. I: The Computer System," DCS Report No. 433, University of Illinois, Urbana, Illinois, February 17, 1971.

given particular attention: the development of a generator of fault detection tests for the control cards and the design of the Sequence Tester.

At the present time, the generator has been completed. A program called "CPTEST" has been written in FORTRAN which implements two "merging algorithms" developed in the above thesis. The idea is:

- (1) to functionally divide the card into subnetworks
- (2) to generate tests for these subnetworks
- (3) to merge the obtained tests.

The structure of the Sequence Tester, interface between the card and the PDP/8e which monitors the testing process, is developed in the above thesis. Its design (packaging, wiring lists, etc.) is 80% completed and its construction has begun.

C. Rey

4.7 DOCUMENTATION

4.7.1 External Documents Issued

Report No. 433 ILLIAC III REFERENCE MANUAL,
VOL I: The Computer System,
edited by B. H. McCormick and
B. J. Nordmann, Jr. February 17, 1971

Report No. 434 ILLIAC III REFERENCE MANUAL,
VOL II: Instruction Repertoire,
edited by B. H. McCormick and
B. J. Nordmann, Jr., February 26, 1971

Outside Talks

Read, John S., "Image Segmentation Techniques for Automated Cervical Smear Processing," 9th Annual Symposium on Mathematics and Computer Science in the Life Sciences, Anderson Hospital and Tumor Institute, Houston, Texas, March 22-24, 1971.

Seminars: Sight Sensory Systems

"The Synthesis of Interval Coverings for Picture Filtering," I, II, III, IV, Professor R. S. Michalski, Department of Computer Science, University of Illinois, January 14, 21, February 17, 24, 1971

"Quantitative Neuroanatomy," Professor Frank Fry, Department of Electrical Engineering, University of Illinois, March 3, 1971

"Investigations in Artificial Non-Symbolic Cognition," Professor Sylvan Ray, Department of Computer Science, University of Illinois, March 10, 1971

"Image Segmentation Techniques for Automated Cervical Smear Processing," John S. Read, Department of Computer Science, University of Illinois, March 17, 1971

"Scene Segmentation with a Graph Transformational Model," John C. Schwebel, Department of Computer Science, University of Illinois, March 24, 1971

"The Particle Transport System: Its features and foibles," Robert C. Amendola, Department of Computer Science, University of Illinois, March 31, 1971

4.7.2 Logic Drawings Issued

37 TP and PAU control logic and 39 TP processing hardware logic drawings have been drawn and issued during the last quarter.

SMV control logic drawings have been updated to current design.

All documentation of control logic is proceeding in parallel with the design.

Set of (23) Sequence Tester logic drawings has been submitted to drafting and is presently being processed.

4.7.3 Engineering Drafting Report

During the last quarter a total of 329 drawings, including drawing changes, layouts, flow charts, thesis, report drawings and drawings related to Opto/Mechanical design of Illiac III have been processed by AT(11-1)-2118 drafting section.

S. Zundo

4.8 ADMINISTRATION

4.8.1 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

Research Assistants

Jerry Chen

Walter Donovan

Lakshmi N. Goyal

Richard P. Harms

S. M. Jayaramamurthy

Ahmad Masumi

Bernard J. Nordmann

Christian Rey

John C. Schwebel

Val G. Tareski

Secretarial

Mrs. Roberta Andre'

4.8.2 Computer Usage Log Summaries

Scanner-Monitor-Video:

During the past quarter, the S-M-V was under power approximately 175 hours with the following division of usage:

Operational	128 hours
Preventive Maintenance and Testing	24 hours
Corrective Maintenance	23 hours

IBM 360/75

Total first quarter expenditure was \$1,713.93

John S. Read

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

5.1 Computerized Mathematics

We continued our research on the naming rule as an additional rule of inference with resolution. The naming rule corresponds to the axiom $C(x, y_1, \dots y_m) \leftrightarrow \epsilon(d(y_1, \dots y_m), x)$ (or $C(z, x, y_1, \dots y_m) \leftrightarrow \neg \epsilon(z, d(y_1, \dots y_m), x)$ where C is a negative clause with all ternary relation symbols). These axioms, intuitively, give names to newly constructed predicates or functions. The restrictions on C in the second case are meant to insure that the new object really is a function. The naming rule says that, given the clause C , one can deduce $C(x, d(y_1, \dots y_m))$ or $\neg \epsilon(z, d(y_1, \dots y_m), x)$. Conversely, the unnaming rule states that, from $\epsilon(t, d(u_1, \dots u_m))$ or $\neg \epsilon(s, d(u_1, \dots u_m), t)$ one can conclude the appropriate instance of C . The need for naming arises from the requirement for Henkin validity--that the interpretation of any formula be in the domain of interpretation. Since a clause can be interpreted as being a predicate or function, we must have a way of guaranteeing that predicate or function is in the domain. This is accomplished by giving that object a name, i.e. $d(y_1, \dots y_m)$. During this quarter we were able to show that if only one instance of such an axiom was needed for inconsistency, resolution and the naming/unnaming rule is sufficient to generate the empty clause, i.e. together they are complete. More generally, we showed that using resolution alone, one can generate a set of clauses I such that each clause in the set of clause parts of axioms, say $C_1, C_2, \dots C_k$, needed for inconsistency is subsumed by some clause in I . Of course,

the set of axioms needed is not known beforehand. However, the above fact asserts that the clause parts of these axioms can essentially be generated by resolution. Thus, in considering possibilities for axioms to add after resolution has failed, one need not consider at one time material from more than one clause.

These results are being collected into a Ph.D. thesis to be presented to the Graduate College for this graduation this June.

(L. J. Henschen)

5.2 Problems In Computational Geometry

An analytical intersection detection procedure for polyhedral objects by means of "even parity modes" has been studied, and its computation results will be shown. As an application of the procedure developed, so called "sofa" problems have been solved and the simulation results for various criteria will be presented.

The concept of a "dynamic" sofa problem will be introduced and an approach for the solution of this problem will be presented.

(K. Maruyama)

5.3 PDP-11

We now have a batch monitor running on the PDP-11. We, also, have interfaced a card reader and line printer to the system. Digital Equipment Corporation has loaned up a high-speed paper tape reader for use until our own Dec tapes are delivered.

We have a basic time-sharing monitor working and hope to have an editor, assembler, etc. for use with it soon.

(D. W. Oxley)

5.4 Factorization Methods Used in the Solution of Partial Differential Equations

We continued to study factorization methods with particular interest given to the symmetric factorization suggested by Stone [1]. The differential equations is approximated by a difference equation at each point of a grid and a system of simultaneous linear equations results.

In matrix notation the system $Ax = q$ is solved using the iteration

$$(A + B)X_{N+1} = (A + B)X_N - \tau_N(AX_N - q) \quad (1)$$

where B is chosen to make each step of (1) easily computable and the procedure rapidly convergent. We have been concerned with the selection of τ_N which is dependent on the eigenvalues of $(A + B)^{-1}A$.

An algorithm was developed and tested to simultaneously calculate the extreme eigenvalues of $(A + B)^{-1}A$ and the iterate X_N using (1). As the approximation to the eigenvalue converges, a better sequence of τ_N 's is selected increasing the rate of convergence of X_N to X .

The algorithm first calculates y_N , an approximate eigenvector of $(A + B)^{-1}A$. Then the approximate eigenvalue

$$u_n = \frac{(Ay_N, y_N)}{((A+B)y_N, y_N)} \quad (2)$$

is calculated. If y is the actual eigenvector, λ the eigenvalue and

$y_N = cy + \sum_{i=2}^K \epsilon_i w_i$, where w_i $i = 2, \dots, K$ is an eigenvector of $(A + B)^{-1}A$

orthonormal with respect to $A + B$ and λ_i $i = 2, \dots, K$ corresponding

$$\text{eigenvalues, then } u_n = \frac{\lambda + \sum_{i=2}^K \epsilon_i^2 \lambda_i / c^2}{1 + \sum_{i=2}^K \epsilon_i^2 / c^2}.$$

Tests have shown the algorithm works as well as using (1) with a fixed "best" value of τ_N .

For the symmetric factorization suggested by Stone in which B depends on a parameter α , it was shown that for $\alpha = 1$ and A derived from Laplace's equation, $\lambda_{\min}((A + B)^{-1}A) > 1/2$ and approaches $1/2$ as the gridsize h goes to zero.

For $A = (a_{ij})$ with constant diagonals $a_{i,i+n} = B < 0$, $a_{i,i+1} = D < 0$ and $\alpha = 1$.

$$\lambda_{\min}((A + B)^{-1}A) \leq \frac{1}{1 + \frac{2\sqrt{BD}}{-(B+D)}} \text{ as } h \rightarrow 0.$$

Furthermore if $a_{i,i+1}$, $a_{i,i+n}$ are constant on the matrices A_m , $m = 1, \dots, M$ I.E. $a_{i,i+1} = D_m$, $a_{i,i+n} = B_m$ in A_m then

$$\lambda_{\min}((A + B)^{-1}A) \leq \min_{M=1, \dots, M} \frac{1}{1 + \frac{2\sqrt{B_m D_m}}{-(B_m + D_m)}} \text{ as } h \rightarrow 0.$$

It is conjectured that $\lambda_{\max}((A + B)^{-1}A)$ grows without bound as $h \rightarrow 0$. If X is the vector $(X_{1,1}, X_{2,1}, \dots, X_{n,1}, X_{1,2}, \dots, X_{n,2}, \dots, X_{n,n})^T$

where $X_{i,j} = \cos^{20}(3(i-j)h\pi/2) e^{-(1+|ih-1/2|+|2h-1/2|)^2(2-\cos(3(i-j)h\pi/2))}$

$$(n+1)h = 1, \text{ then for } h = \frac{1}{30}, R_h = \frac{(AX, X)}{(A+B)X, X} \approx 4$$

$$\text{for } h = \frac{1}{100}, R_h \approx 10$$

$$\text{and for } h = \frac{1}{250}, R_h \approx 18$$

$R_h \leq \lambda_{\text{MAX}}((A+B)^{-1}A)$ thus it appears that as $h \rightarrow 0$ λ_{MAX} grows considerably and the convergence of (1) is slow. Work is in progress to get some theoretical results on the growth of λ_{MAX} .

(M. Diamond)

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

NOTE: The following work was performed during the period July 1 to October 1, 1970 and is reported on here because it was inadvertently omitted from the earlier Quarterly Progress Report.

5.5 Graph Algorithm Research

During this quarter, a new algorithm for finding all the trees in a graph was programmed in GASP and debugged. When tested on the complete graph on nine nodes, the program took under two minutes (on an IBM 360-91), averaging less than 250 nanoseconds per tree.

The following table compares the computation costs of the various algorithms for finding all trees. "n" is the number of nodes, "b" the number of branches and "t" the number of trees in the given graph. For the complete tree, $t = n^{(n-2)}$. Details will appear in a future departmental report.

<u>Author and Year</u>	<u>Type</u>	<u>Growth Rate of Cost</u>	<u>Complete Graph</u>
MacWilliams 58	exhaustion	$= \binom{b}{n-1}$	$\leq \frac{n^{2n}}{2^n(n-1)!}$
Zobrist 64	exhaustion	$= n^{n-1}$	$= n^{n-1}$
Mayeda 65	tree sections	$\geq t$	$\geq n^{n-2}$
Kamae 67	hamiltonian path	$> t$	$> n^{n-2}$
Ballert 62	cartesian product	$\leq t \left(\frac{2b}{n}\right)^{n-1}$	$\leq n^{n-2}(n+1)^{n-1}$
Mason 57	cartesian product	$\leq n \left(\frac{2b}{n}\right)^{n-1}$	$\leq (n+1)^{n-1}$
Feussner 02	$T = T(b) \quad T(b)$	$\geq t$	$\geq n^{n-2}$
Percival 53	factoring	$\leq (n-1)!$	$= (n-1)!$
Chase 70	factoring	$< (.8)^n(n-1)!$	$< (.8)^n(n-1)!$

(S. Chase)

5.6 Stability Charts of Stiffly Stable Methods for Digital Simulation

The stiff differential equations arise in many computer aided design techniques particularly network analysis and simulation. The real-time element in simulation makes the computer operate as in a real word environment where, for example, one second of computer time is one second of actual time. This real-time requirement is particularly important in the training function of simulators, but it is also highly desirable for simulation in general. The real-time digital simulation of continuous systems received its greatest initial impetus in the 1950's when Gray [1] developed the method for constructing the stability charts of numerical integration methods of differential equations.

Gear [2] has developed stiffly stable methods of order as high as six and showed their application to the stiff differential equations. Jain and Srivastara [3, 4] have obtained stiffly stable methods of order K as high as 11 for suitable parameters.

Here we have constructed stability charts for stiffly stable methods. The stability charts enable us to choose an optimal integration step and thus result in saving considerable computer time.

(M. K. Jain)

-
- [1] Gray, J., Jr., "Numerical Methods in Digital Time Simulation," Moore School of Engineering, University of Pennsylvania, Philadelphia, Pennsylvania, (Ph.D. Thesis), June, 1953.
 - [2] Gear, C. W., "Numerical Integration of Stiff Ordinary Differential Equations," Department of Computer Science, University of Illinois, Urbana, Illinois, Report No. 221, January, 1967.
 - [3] Jain, M. K., and V. K. Srivastava, "High Order Stiffly Stable Methods for Ordinary Differential Equations," Department of Computer Science, University of Illinois, Urbana, Illinois, Report No. 394, April, 1970.
 - [4] Jain, M. K., and V. K. Srivastara, "Optimal Stiffly Stable Methods for Ordinary Differential Equations," Department of Computer Science, University of Illinois, Report No. 402, June, 1970.

6. SWITCHING THEORY AND LOGICAL DESIGN

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

After completing the comparison of the implicit enumeration method (explicit use of inequalities) with the branch-and-bound method (no use of inequalities) and the improvement of the branch-and-bound method, T. Nagakawa left. Y. Kambayaski joined the group. We started to look at the logical design problem with integrated circuit.

S. MUROGA

A program was devised and debugged which will draw (by Calcomp) networks of NOR, OR, AND, WIRED-OR, and WIRED-AND gates when given the connections among gates and external variables. The intent of this program is to prevent time-consuming reconstructions of networks from the connection patterns which are the present form of output from our implicit enumeration and branch-and-bound programs.

(J. Culliney)

Properties of logic families of integrated circuits are studied in order to investigate (1) conditions for wired logic and (2) possibility of mixing different types of logic gates. For example, there exists a NOR gate with wired AND capability (RTL), a NOR gate with wired OR capability (ECL), a NAND gate with wired AND capability (DTL, TTL) and an AND gate with wired OR capability (CTL). Synthesis procedures using logic gates with wired logic capability

are not treated before. Several properties of such gates were obtained, which will be used in synthesis methods.

(Y. Kambayashi)

NOR networks with the minimum number of interconnections for all 3-variable functions and some 4-variable functions have been obtained with the program ILLOD(NOR-B). Two out of seventy seven 3-variable functions and five out of fifteen 4-variable functions have the minimum interconnection networks which differ from the networks with the minimum number of gates.

The program ILLOD(NOR-B) is augmented with an optional use of the magnetic tape to store the solutions obtained by this program.

(H.C. Lai)

Some theoretical studies were made on the logic networks which consist of MOS complex cells. A few properties of these networks were obtained. An algorithm was developed for the synthesis of multi-level networks with a minimum number of cells.

(T.K. Liu)

7. SOUPAC SECTION

(Statistically Oriented Users Programming and Consulting)

During the period January through March, 1971, the SOUPAC consultants began the third series of SOUPAC seminars. The weekly seminars have proven a valuable means of communicating the scope and power of SOUPAC as well as giving valuable insights concerning its use. In addition, weekly staff meetings are being held to acquaint members of the SOUPAC staff with the theory and use of some of the newer programs. Some consideration has been given to opening these meetings to the public.

The fourth edition of the SOUPAC Manual was issued in March. This edition contained, for the first time, the programs FIT-Chi Square Goodness of Fit, Kolmogorov-Smirnov D Statistic, Paired Comparisons, Three Mode Factor Analysis, a completely rewritten Transformations, and several revised or additional Matrix operations.

The SOUPAC consultants personally conducted class sessions in SOUPAC for two classes, Advertising 381 and Economics 477. Writeups and manuals were provided for several others.

A program feature called ALLOC8 (allocate) is being incorporated into all programs which are not currently dynamically allocatable with roll-in and roll-out of data. Dynamic allocation with roll-in and roll-out enables a program to be run in a arbitrary region size with arbitrarily large data. This capability generates overhead in execution of a program (notably machine time and I/O) and also requires extensive reprogramming. ALLOC8 enables a program to be run in arbitrary region size, but only on data which can be accomodated by that region size. ALLOC8, by virtue of its simplicity, cuts down on overhead and reprogramming, but maintains the advantage of running in arbitrary region. ALLOC8, in the case of a problem

size which is too large for the requested region will not execute but will provide a message indicating how much more region would be required to run the problem. The vast majority of research problems can be fit into a region size less than or equal to the maximum made available to users of our machine. Thus ALLOC8 provides for optimal matching of data to region. Only problems of a magnitude such that they cannot be contained in maximum available region would suffer from the fact that roll-in and roll-out of data is not provided by ALLOC8.

During this last quarter further study of the SPSS statistical package was done. The conclusion of the SOUPAC group was that it was one of the better small statistical packages available.

Kern W. Dickman

8. MATRIX MULTIPLICATION OPERATION MINIMIZATION

Given a product of a sequence of conformable matrices and vectors, a reduction of the number of operations is possible by proper rearrangement of the sequence. That is, since all such multiplications are associative and some are commutative, the time required is highly dependent on the order in which the multiplications are performed. An example is $C \times R \times A$ where C , R , and A stand for a column vector, a row vector, and a matrix of dimension n respectively. Then the computation $(C \times R) \times A$ requires $n^3 + n^2$ multiplications whereas an equivalent computation $C \times (R \times A)$ requires only $2n^2$ multiplications. A simple algorithm which associates pairs of arrays in a way that always gives the minimum number of operations has been developed. It was also shown that the same algorithm could be used to evaluate matrix expressions on a parallel machine with the minimum number of operations as well.

D. Kuck and Y. Muraoka

9. PARALLELISM EXTRACTION

Introduction

A program analyzer which takes a program written in a conventional language, e.g. FORTRAN, and analyzes for a parallel machine (e.g. STAR, Illiac IV) is being programmed in PL/1. The analyzer not only measures potential parallelism in a program but also modifies it if necessary to extract more parallelism. In the past quarter emphasis was put on the analysis of an assignment statement and a DO loop. Parallel computation of an assignment statement has been studied by many people. An important feature being overlooked is the possibility of reducing computation time by distribution, e.g. compare parallel computation of two equivalent expressions $\sum_{i=0}^n a_i x^i$ and $a_0 + x(a_1 + x(a_2 + \dots(a_{n-1} + a_n x)\dots))$. An algorithm has been developed and implemented to achieve this end.

The DO loop analyzer has also been coded. Given, e.g. a matrix addition coded in FORTRAN, the analyzer indicates that it can be computed in one step on a parallel machine, that is, element-wise additions are independent. If it is appropriate, the analyzer tries to, e.g. partition a DO loop to achieve higher parallelism. At this point, the analyzer consists of three major parts, Masterprogram, DO Loop Subprogram and Assignment Statement Subprogram. The details of each part are given subsequently.

9.1 Masterprogram (C. Cartegini)

The input information in the form of FORTRAN code is mapped into a suitable form to be used by the subprograms processing the tree height of

arithmetic expressions within their own context. The purpose of the Master-program is initially to delimit each context and submit the related information in the form of parameters to the subprograms.

To do this, the stream of FORTRAN statements is partitioned into blocks characterized by the absence of transfer-control statements. These blocks are identified either by arithmetic assignment statements or DO-group statements. Non-executable FORTRAN statements are handled separately. The blocks as well as selective table-information are submitted to the subprograms.

The returned information in the form of unique or optional values is used as a weight associated to the nodes of a precedence graph identifying the original code and used in drawing the predictable behavior of the analyzed FORTRAN program as far as the rule of making no assumption in the actual value of input variables is not violated.

9.2 Assignment Statement Subprogram (J. Han)

This section describes the work of implementation of the tree height reduction algorithm discussed by Muraoka [3]. It describes two important properties, holes and spaces, within an arithmetic expression. By using distribution of multiplication over addition to fill these holes and spaces the tree height for an arithmetic expression can be reduced. The algorithm developed is based on some assumptions about a particular machine organization. Originally, this algorithm assumes, for simplicity, the operations of multiplication and addition have no difference in execution time between them. In the development of the implementation, this algorithm has been expanded to cover divisions and subtractions on the assumption that all kinds of operations take almost the same execution time.

So far the implementation of the tree height reduction algorithm has been programmed in PL/1 and tested. The first step of this implementation algorithm is to standardize an arithmetic expression. By this we mean to delete redundant parentheses for easy recognition of parallelism within it. Then we scan the standardized arithmetic expression to generate temporary results when one is found. Once the temporary result is to be generated, examination for the existence of holes and spaces is made to determine whether distribution of multiplication or division over additions and/or subtractions should be done. If one exists we need to rearrange the syntactic tree for the combination of the terms in their distributed form. If no distribution exists we simply build a syntactic tree for it. This algorithm uses a multipass-scanning mechanism. The number of passes required is not the tree height of an arithmetic expression, but depends on the depth of parenthesis nesting of an expression.

9.3 Do Loop Subprogram (S. Chen)

The sequential and iterated operations of DO or similar statements appear most frequently in many programming languages and statistics have shown that a lot of parallelism is lying within these operations specified by the usual application programmer. These can be done in parallel on a multiprocessor system and this results in decreasing the computing time.

Some types of parallelism existing in a restricted class of DO statements have been found and proposed by Y. Muraoka [3], based on the ease of compiling for future multiprocessor machines. Algorithms for detecting these types of parallelism have been sought and implemented in PL/1 programs.

The more general a nested DO loop is, the more complicated it becomes to discover parallelism, partially because this property primarily depends on the value of the index expression associated with each array variable, which might in turn depend on a value derived at execution time. Under certain assumptions, a general algorithm has been established and implemented in PL/I programs to test the parallelism in nested loops, by using the algorithms for simple cases as building blocks. By this general algorithm, different kinds of parallel operations can be formed in terms of the number of processors in use. Some experiments are going to be performed on a set of randomly selected FORTRAN programs and the result will be compared with the original sequential operations of single processors. Furthermore, the effect of these assumptions upon the result will be evaluated.

Conclusion

In the next quarter the above three parts will be put together and experiments on real application programs will be done. Besides, refinement of the analyzer is also planned. Elaboration includes: (1) the analysis of a program structure, e.g. IF statements, (2) the measurement of a program, e.g. the frequency count of operators.

9.4 Tree-Height Reduction With Weighted Operands And Graph Schedules (P. Kraska)

Given an arbitrary arithmetic expression of scalar operands, there are algorithms (Muraoka [3], Baer [1]) which parse these expressions into trees such that maximum parallelism is exposed (i.e. tree-height is minimized). Thus, presumably, a computing system of parallel processors could evaluate the expression in minimal time. However, these algorithms do not account for the fact that the binary operators multiply, divide, and add (subtract) require different amounts of time for execution. During this quarter we have developed algorithms which minimize tree-height with time-weighted nodes by appropriate parsing of the expressions. Furthermore, the algorithms permit analysis to determine when distribution of multiplication and division across summation will reduce tree-height.

Tree-height minimization of arithmetic expressions of matrix operands, where the matrices are non-square, is a much more difficult problem since matrix multiply is not abelian and each matrix multiply node has a weight of $W_m = W_a \log_2 d_i$, where d_i is the conformable dimension of (M_i, M_{i+1}) , W_m and W_a are the time-weights of the scalar operators multiply and add, respectively. However, some progress has been made to the solution of this problem and it is conjectured that we can inductively prove that the tree-height of a product of n matrices may be minimized.

Tree-height reduction sometimes increases the number of nodes; thus any finite parallel processing system may be burdened with more work while evaluating the expressions. Evaluation of common subexpressions only once keeps the number of nodes to a minimum, but this introduces another problem; while algorithms exist which schedule m processors on a tree

(t. C. Hu [2]), there are currently no published algorithms which schedule m processors ($m > 2$) on a graph. However, we have had some success in this area. By analyzing an ordered connectivity matrix of a reduced directed graph it appears that cut sets of order m (i.e. m nodes which are mutually independent) may be extracted. It is encouraging to note that a recent paper by Ramamoorthy [4] uses this same technique. The problem of scheduling m -processors where weighted nodes comprise the graph is still unsolved.

9.5 Simulation Processor (E. Davis)

The purpose of this research is to study digital computer simulation of discrete time systems with the goal being the design of a machine for simulation processing. To achieve a significant factor of improvement in processing speeds a multiprocessor organization is being examined.

An interesting aspect of the design is that a range of the real time being simulated can be involved in the processing being done concurrently. That is, at an instant of computer time, more than one instant of simulated real time can be in process. This implies all the real time interactions plus those caused by the processing overlap of events at distinct times must be considered.

The problem then is to resolve all the processing interactions in a way that tends to maximize the processing that can be done concurrently.

Simulation languages have been studied. They reveal potential for concurrent processing due to many transactions flowing through a single program, corresponding to many executions of the program.

Resolution of the run time precedence requirements is being designed into a control unit. Output of the unit is tasks for the processors that make up the multiprocessor configuration.

9.6 Memory Hierarchies (D. Gold)

Memory speed and memory cost are roughly inversely related. This indicates a significant potential savings for systems which have problem mixes requiring large amounts of memory.

Specifically, an ideal system (memory-cost-wise) will have only as much primary memory as is required by the basic operations (kernels) in its problem mix. The remainder of its memory being slower - and hence cheaper.

One of the problems in devising such a system is that of making the bulk-storage device look like a random-access device. These memories (for obvious cost considerations) are typically periodically addressable memories: disks, drums, recycling delay lines, etc. This has been accomplished for several large classes of problems by having a memory hierarchy consisting of primary memory, large periodically addressable memory, and relatively small bulk (random-access) memory. This bulk memory acts as a buffer to the other memories and early results indicate that it need be no larger than approximately one percent the storage capacity of the periodically addressable memory.

Another problem being investigated is that of precisely defining those classes of problems for which solutions have been found. Thus far, two- and three-dimensional mesh problems and most matrix operations have been found to be contained in these classes.

Included in these results is a method whereby any set of data objects (mesh points, partitions of matrices, etc.) may be successively randomly permuted between operations on them.

9.7 Microprogramming and File Processing (L. Hollaar)

During the last quarter, prototype 3 of the Burroughs D-machine was checked out and is now functioning correctly. However, since the machine lacks input/output facilities, the only programs run on the machine have been short diagnostic routines. Current plans are for this machine to act as an input/output controller for the new processor from Burroughs.

The card reader and the printer have been loaned to the PDP-11 project; assistance was provided for the installation and interfacing of these two devices.

Planning for the interface of the disk to the new processor has begun. As many disk functions as possible have been checked and the crystal clock used during writes have been received.

9.8 Text-Searching Project (W. Stellhorn)

A principal application planned for the D-machine is the investigation of algorithms for searching large volumes of textual material for specific information requested by a user. Such a system would differ from most existing information retrieval systems in that it would operate directly with original text rather than with keyword lists or abstracts. Of particular interest are techniques for rapid text searching and for assisting the user in eliminating material which satisfies his request but which is of no interest.

To this end, preliminary investigations are beginning in the general area of information retrieval. Also under study are potentially useful properties of English prose, such as word frequencies and the characteristics of intervals which typically separate occurrences of words of various classes.

A data base consisting of 67 technical articles in digital form for initial experimentation has been obtained from Lehigh University and is presently being reformatted.

9.9 D-Machine Micro-Programs (H. Yamada and W. Stellhorn)

D-machine micro-programs for interpreting the S-language have been completed and substantially debugged. No further development or testing of these codes is planned until after installation of the computer. These are described in a forthcoming M.S. thesis by Hirohide Yamada [5].

9.10 S-Language Assembler (E. Polley)

In order to write programs for the D-machine it is necessary to have the instructions in the form required for Yamada's interpreter [5]. To prepare instructions for the interpreter they will first be written in an assembly language, created for the project, then translated into the object code which can be accepted by the interpreter. This translation process will be done by an assembler which is currently being finished.

The assembly language has two types of instructions: word and string. Word instructions are similar to those found on most machines, are of a three address format and include operations such as addition, subtraction, shifting and branching. The arithmetic instructions all have two forms one of which checks for overflow.

The other type of instructions, the string instructions, do the character manipulation which is the heart of this project. These instructions search for characters, compare strings and move strings in core.

The assembler presently can handle the word instructions only. It is a 360 assembly language program and will punch its object code into cards which will be read by the reader attached to the D-machine and loaded for interpreting.

9.11 Debugging (M. Kaplan)

Since September of 1970, we have been debugging a set of statements to be added to the WATFIV compiler to aid users in debugging logical errors in their programs. These statements include three kinds of traces - a line trace to trace the flow of execution, a variable trace to trace the flow of specific variables during the execution of a program, and a subprogram trace which will trace the invocation of function and subroutine subprograms and also print out the values of their parameters upon entering and leaving the routine. These traces will be similar to those provided in the FORTRAN-G compilers' debugging package but will be more flexible. For example, it will be possible to control their use at execution time. The other statement in the package is, as far as we know, a new idea. The rationale for this statement is that people write and debug programs in blocks. What this statement attempts to do is to provide an easy way to debug these blocks of code. The statement is of the form:

```
DEBUG < statement # > INITIAL < list of assign. statements > AT  
STATEMENT < statement # > < list of logical exp. > AT STATEMENT  
< statement # > < list of logical exp. > GO TO < statement # >
```

for example:

```
DEBUG 10 INITIAL X = 10.4, Y(1) = 12*Z**2 AT STATEMENT 20  
X.EQ.20.5, I.LT.12 GO TO 50
```

The effect of this statement is to execute the block of code from the debug statement to statement number 10 with the initial conditions specified by the INITIAL clause. When it reaches statement number 20, it tests if X.EQ.20.0

and if I.LT.12 and prints out appropriate messages. The GO TO 50 clause causes the DEBUG package to check that when the program leaves the DEBUG block it goes to statement 50.

A statement similar to the PL-I ON statement is also under consideration.

For the past three semesters, CS 109 students have been using, modifying and writing about a partial implementation of the DEBUG statement written as a PL-I prepass to WATFIV. Tests with these students, although not conclusive, show a tendency for them to debug the logic of their programs faster if they use the statement than if not. Further tests will be conducted with larger samples of CS 101 students this summer.

At the present the implementation into the WATFIV compiler of all the statements described is proceeding with completion hoped for by June.

LIST OF REFERENCES

- [1] Baer, J. E., "Graph Models of Computations in Computer Systems," Ph.D. Dissertation, University of California, Los Angeles, Report No. 68-46 (October, 1968).
- [2] Hu, T. C., "Parallel Sequencing and Assembly Line Problems," Operations Research, 9 (November-December, 1961), pp. 841-848.
- [3] Muraoka, Y., "Parallelism Exposure and Exploitation in Programs," Ph.D. Dissertation, University of Illinois at Urbana-Champaign, Report No. 424 (February, 1971).
- [4] Ramamoorthy, C. V., and Chang, L. C., "System Segmentation for the Parallel Diagnosis of Computers," IEEE Transactions on Computers, C-20 (March, 1971), pp. 261-270.
- [5] Yamada, H., "Emulation of Disc File Processor," M.S. Dissertation, University of Illinois at Urbana-Champaign, Report No. 436 (June, 1971).

10. COMPUTER SYSTEMS ANALYSIS

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

During this report period the National Science Foundation initiated Grant No. US NSF GJ 28289 for research entitled "Computer Systems Analysis."

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 is being investigated.

10.1 Computer Network Modeling

We have begun this research in Computer Systems Analysis by formulating a mathematical model for a geographically distributed network of computers. 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 queue disciplines and we are currently investigating nonpreemptive priority disciplines. Our future endeavor will include modeling preemptive priority and dynamic priority queues.

10.2 Center Throughput Analysis

We have begun research in this area by focusing our attention on 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. Based on the essential characteristics of the various computers within the center, we shall employ our queueing theory model to facilitate studying the effects of priority assignment and job dispatching.

Further effort in this area will be deferred until the queueing theory model is defined.

(E. K. Bowdon)

11. ILLIAC IV

(This work was supported in part by the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, 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

Automation Technology, Inc. reports that PE tests and Card tests proceeded well during the quarter. Diagnostic efforts will be concentrated on "diagnosing" conversion problems through a major portion of the next quarter.

During this quarter the ILLIAC IV Project has received a Burroughs B6500/B6700 Computer at the University of Illinois. This machine has been obtained primarily to support the programmers developing the ILLIAC IV operating system. It is also available to applications programmers and to prospective users of the ILLIAC IV computer.

The software effort was directed toward integrating the Operating System with the ILLIAC IV simulator on the B6500, converting the Cockroach-to-Glypnir translator to the B6500, supporting the assembler (ASK) on the B6500, and simulating ILLIAC IV special functions and algorithms written in ASK for a mathematical subroutine library. The Software Reference Manual is nearly complete.

Several major software efforts for the PDP-11 system were continued or completed. Hardware delivery began on the various segments of the ARPA network terminal system. Negotiations were completed and agreement reached with Digital Equipment Corporation on a cooperative research arrangement for joint development of the ARPA network port facility.

Application efforts in Numerical Analysis continue in solving partial differential equations, matrix inversion and linear algebraic equations, eigenvalues and eigenvectors, estimation and filtering, and identification of nonlinear differential equations.

Three seminars were offered during the quarter.

Project expenditures and commitments through March, 1971:

Burroughs Corporation	\$26,723,000.00
University of Illinois	6,814,273.83

HARDWARE

11.1 Off-Line Diagnostics

11.1.1 PE Test

The analysis of the PE logic and the development of tests to be used to test the PE off-line through use of the PE Exerciser (PEX) is progressing well. The Path Tests and Combinational Logic Tests are completed and in use. The analysis necessary for development of tests of the Control Logic is nearing completion and some test sequences have been defined. This area is expected to be complete by late summer.

11.1.2 Card Test

The development of tests to isolate stuck-type faults in active devices is on schedule except as noted below. All tests for PE Cards are complete, have been verified, and are in use. Of the CU Cards, the Type 1 boards are scheduled to be completed by mid-summer. The type 1 boards have been divided into two classes, 31 Priority Boards and 40 Non-Priority Boards. Twenty-eight Priority Boards and 21 Non-Priority Boards have their tests generated; however, these tests are not verified, nor are they in use, since there is no CU Card Tester operational.

11.2 Program Conversion

Conversion of the program used to develop the diagnostics mentioned above was begun during the quarter. Because of the importance of conversion to be able to carry on useful test generation, some other scheduled work is being postponed. The level of effort on the PE Control Logic Tests is temporarily reduced to analysis only and no board tests are being generated. The majority of diagnostic efforts will be concentrated on "diagnosing" conversion problems through a major portion of the next quarter.

11.3 ILLIAC IV Maintenance

ATI maintenance engineers are actively participating in the on-line debugging of ILLIAC IV. Among the areas of active participation have been: PE Board Test, PEM/MLU Test, PE Test, CU Test and I/O Test. In addition, ATI has identified the long lead spare parts and estimated the required number of each. Under authorization of the University of Illinois, ATI is obtaining competitive prices, where possible.

11.4 Financial

At this point in the contract, ATI is approximately ten percent (10%) below the budgeted cost estimates.

SOFTWARE

11.5 Operating System

11.5.1 Operating System I

The operating system is at present being integrated with the ILLIAC IV simulator (SSK) on the B6500, and has reached the stage where simple ILLIAC IV programs have been run under the control of the ILLIAC IV operating system. Instrumentation and error diagnostics are now being placed in some of the modules to allow the performance of the operating system to be measured in detail. An Operating System Maintenance Manual has been written.

11.6 Compilers and Translators

11.6.1 Glypnir

Glypnir, Version II, has been undergoing further consolidation. I/O for ILLIAC IV (as opposed to Simulator I/O) and Macro Facilities in the form of a pre-processor have been provided. The Glypnir compiler has also undergone a basic measurement investigation with a view to speeding it up. The possibility of providing a facility for separately compiling subroutines has been and is still being studied. A Glypnir Compiler Maintenance Manual has been written.

11.6.2 Cockroach

During the quarter the Cockroach-to-Glypnir translator was completed for a subset of the specified language on the B6500 and converted to the B6500. Subroutines and functions will be completed in the beginning of the second quarter. User documentation of the available features was also provided. The addition of an hourly employee to the staff hastened the completion and provided for more extensive debugging.

Cockroach is now available to users, and a certain limited amount of user support also is available.

11.7 Assembler

The Assembler (ASK) is now supported on the B6500. It compiles at about 1200 cards per minute. A plan for increasing its compiling speed to 2500 cards per minute has been formulated and is being implemented.

11.8 Interactive Communications and Graphics

11.8.1 Interactive Communications

During the reporting period several major software efforts for the PDP-11 system were continued or completed.

1. A high-level language compiler, PEESPOL, was completed. Version II is running on the B6500.
2. Version I of the ARPA Network Terminal System, ANTIS, was completed. Basic portions of the system were checked out and actual runs made on the PDP-11.
3. Design of the link-up between Paoli and the University of Illinois, via the ARPA network, using two ANTIS systems was completed and initiated.
4. Final specifications were agreed upon by Burroughs for the construction of an interface between the B6500/B6700 system and the ARPA network IMP.

During this reporting period, hardware delivery began on the various segments of the ARPA network terminal system. The basic PDP-11 processor, the 16K words of memory, the real-time block, the high-speed paper tape, ASR35 operator's teletype, four 2400 Baud line interfaces with adapters, one dataset control and interface for a Bell 103A dataset, plus two general-purpose interfaces to be used to connect the Gould Electrostatic Plotter to the system and to connect the Computek storage scope system were received. The remainder of the items ordered should be delivered within the next reporting period.

Two interfaces between the ARPA network IMP and the PDP-11 were received. Both interfaces were debugged, checked out, and installed in the University of Illinois PDP-11 and the Paoli PDP-11. Success in getting on the network now hinges on the completion of system software development.

Negotiations were completed and agreement reached with Digital Equipment Corporation on a cooperative research arrangement for joint development of the ARPA network prot facility.

11.8.2 Graphics

No work was done in the graphics area during this reporting period because of total lack of personnel. An effort was begun, toward the end of the period, to acquire a full-time professional in graphics who is expected to join the Center during the next period.

11.9 Mathematical Subroutines Special Function Library

This quarter's work has continued on the ILLIAC IV Special Functions and Algorithm subroutine library. The following set of routines written in ASK have been successfully simulated by the current version of the simulator and the results are good for 15 significant digits:

- 64-bit mode: •sine and cosine
- tangent and cotangent
- natural logarithm
- square root
- arctangent (1 argument)
- arctangent (2 arguments)
- hyperbolic sine and cosine
- hyperbolic tangent
- gamma range(0,1)
- gamma range (0, ∞)
- 32-bit mode: •sine and cosine
- tangent and cotangent

- exponential
- natural logarithm
- square root

Algorithmic library:

- addition/subtraction of matrices stored straight or skewed
- multiplication of matrices stored straight
- transpose of a matrix stored straight

At present work is being completed for:

- 64-bit mode:
- gamma (X) $0 < X \leq 57$
 $57 < X < \infty$ and
 $X < 0$
 - error function: erf (X) and
LN gamma (X).

For matrix operations the set is being expanded to rectangular matrices stored straight and skewed. The same set will also be made available in Glypnir. These routines will continue to be refined. Maintenance, distribution and programming assistance in using the mathematical routines is available.

11.10 B5500/B6500 Operations

During this quarter the ILLIAC IV Project has taken delivery of a Burroughs B6500 computer at the University of Illinois. This machine has been rented primarily to support the programmers developing the ILLIAC IV operating system. It is also available to applications programmers from the Center for Advanced Computation and future users of ILLIAC IV. The machine was installed in the basement of the Coordinated Science Laboratory and became operational around March 1.

11.11 Software Documentation

The Software Reference Manual is now at 900 pages, largely completed. Its one deficiency is an Assembler users manual, which is at present about half completed.

APPLICATIONS

11.12 Numerical Analysis

11.12.1 Partial Differential Equations

11.12.1.1 Numerical Solution of Problems in Hydrodynamics

The results of numerical experiments with the Brailovskaya, Dufort-Frankel, Cheng-Allen, Crank-Nicholson and Lax-Wendroff finite difference schemes on the Burgess equation have been presented in a formal report which is now in the process of being printed [1].

A code is being written and debugged for the two dimensional subsonic and transonic flow around a circular cylinder, as the next step in exploiting fully the techniques of parallel processing in the numerical computations of three-dimensional fluid and gas flow problems. In a polar co-ordinate system the governing equations in non-dimensional form are:

$$\text{Continuity} \quad \frac{D\rho}{Dt} + \rho \nabla \cdot \vec{V} = 0;$$

$$\text{Momentum} \quad \rho \frac{D\vec{V}}{Dt} + \nabla p = \sqrt{\gamma} \frac{M_\infty}{Re} \left(\frac{4}{3} \nabla \nabla \cdot \vec{V} - \nabla \times \nabla \times \vec{V} \right);$$

$$\text{Energy} \quad \rho \frac{DS}{Dt} = \sqrt{\gamma} (\gamma - 1) \frac{M_\infty}{Re} \Phi + \gamma \frac{\sqrt{\gamma} M_\infty}{Re \cdot Pr} \nabla^2 (p/\rho).$$

$\vec{V}$ = velocity, p = pressure, S = entropy, γ = ratio of specific heats, M_∞ = free stream Mach number, Φ = dissipation terms, Re = Reynolds number, Pr = Prandtl number.

The computational region comprising the flow field is divided into a suitable number of mesh points, which is a function of computer

storage capacity, and the dependent variables, namely the pressure p , radial and tangential components of the velocity, i.e. u , v , and temperature, T , are calculated at each mesh point. The equation of state for a perfect gas is assumed. Initially the gas is assumed to be flowing with a uniform velocity and the physical properties are assumed uniform at each mesh point.

To gain flexibility and save time on runs while the program is being debugged, the viscous terms have been suppressed in the calculation, so that the flow is treated as being inviscid. At time $t = 0$, the condition of zero radial velocity is imposed on the cylinder boundary, a suitable time step is selected, and the finite difference representation of the governing equations is used to calculate the values of the dependent variables at succeeding times. The finite difference scheme, chosen for these experiments is the two-step, second order accurate, Richtmyer variation of the Lax-Wendroff scheme [2], which has been previously tested with the Burgess equation [1].

11.12.1.2 Algorithm Development

During this period several numerical methods have been investigated in cooperation with AMOCO Oil Company to solve the elliptic type partial differential equations. The semi-iterative block Jacobi method and ADI method were adopted and both were implemented in Glypnir. The first method worked correctly and was made into a general type subroutine. An iterative step takes about 600 microseconds on ILLIAC IV in the case of 21×21 mesh points. The second method is still being implemented. The storage scheme for this method is more complicated than the first method.

11.12.2 Matrix Inversion and Solution of Linear Algebraic Equations

During this quarter, a program to solve linear equations and perform matrix inversion has been written in ASK, and is presently being debugged. It uses the Gaussian Elimination method and is designed to handle full matrices up to 630×630 for Gaussian elimination and up to 475×475 for inversion. These large matrices are not core

containable and are therefore broken down into blocks of 64 columns, each block being processed independently of the others as far as possible.

The algorithm works in two stages, first the matrix A is decomposed into the product of lower and upper triangular matrices, L U.

The second stage uses this decomposition to operate on a right hand side to solve $Ax = b$, on a unit matrix to form A^{-1} or on an arbitrary matrix B to form $A^{-1}B$. To reduce the round-off error, 'partial pivoting' is employed (see e.g. Wilkinson, J. H., The Algebraic Eigenvalue Problem), also the inner products of vectors are accumulated using double precision arithmetic.

Work is just beginning on the Conjugate Direction method which will be implemented in Glypnir. This should prove an efficient algorithm for solving $Ax = b$ when the matrix A is not given explicitly but a subroutine which computes Ax given the vector x is available.

11.12.3 Eigenvalues

Jacobi's Method for finding eigenvalues and eigenvectors of real symmetric matrices (including complex Hermitian matrices: Let $A = B + iC$ be a complex Hermitian matrix where B is real symmetric ($B = B^t$) and C skew-symmetric ($C = -C^t$), then the real symmetric $2n \times 2n$ matrix

$$A^1 = \begin{bmatrix} B & -C \\ C & B \end{bmatrix}$$

can be an input to the Jacobi algorithm) has been coded in ASK and satisfactorily tested on the B5500 ILLIAC IV simulator.

Eberlein's Method (Jacobi-like algorithm) for normalizing real non-symmetric matrices has also been satisfactorily tested.

The results of both algorithms agree with the literature up to 14 significant digits. An error of order 10^{-15} is due to conversion routines from internal machine representation of floating points to their external decimal representation.

To have a means of comparison, two well established algorithms by H. Rutishauser [3] and P. J. Eberlein [4] have been combined in one B6700 ALGOL program. It is in a running condition and exists under the file name (completed version) WINFRIED/BERNHARD/PROC. This file is a separately compiled procedure. The name of the procedure is JACEIG (N,A,TS,D,ROT,TMX).

In using this procedure the user has an option

- 1) He may write his own main-program for I/O and procedure call, but he then has to go through the B6700 BINDER for interrelation between his file and the above file or,
- 2) He can use the file BERNHARD/JACOBI/EIGEN which does the I/O and calculation for him.

For case (2) the following control cards and input-data are required:

```
8 EXECUTE BERNHARD/JACOBI/EIGEN
8 BCL EBOR
<integer 1>, <integer 2>,
matrix-elements in real.
8 END
```

where 8 is an invalid character like multipunch 1,2,3

<integer 1> = order of matrix

<integer 2> = TMX with

TMX = 0 no eigenvectors are calculated and printed

TMX > 0 the right eigenvectors are calculated
and printed

TMX < 0 the left eigenvectors are calculated
and printed.

The matrix for which the eigenproblem is to be solved can be either real symmetric or real non-symmetric. If real symmetric then only the eigenvalues are printed in decreasing order. If the eigenvectors are to be printed ($TMX \neq 0$), then this will leave the same order as the eigenvalues, i.e., to λ_j corresponds TS_j in the order of their appearance.

If real non-symmetric the total eigenvalue matrix is printed. If the eigenvalues are complex then the a_{ii} occupy the diagonal while the imaginary part occupies the off diagonal portions.

If $a_{ii} \pm i a_{ij}$ is the eigenvalue, the corresponding eigenvector $t_i \pm i(t_j)$ appears in column (row) i and column (row) j .

11.12.3.1 Eigenvalues and Eigenvectors of Symmetric Tridiagonal Matrices

11.12.3.1.1 QR Algorithm with Origin Shifts

The QR algorithm for finding eigenvalues of symmetric tridiagonal matrices has been written in XALGOL, debugged and tested for different-size matrices. A short description of the algorithm follows.

A is a symmetric tridiagonal matrix with diagonal elements C_i and subdiagonal elements b_i . Any symmetric tridiagonal matrix A_t , ($t = 1, 2, \dots$), may be expressed as

$$A_t = Q_t R_t \quad (1)$$

where Q_t is orthogonal and R_t is upper triangular of the form

$$\begin{bmatrix} p_1 & q_1 & r_1 & & & 0 \\ & p_2 & q_2 & r_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & r_{n-2} & q_{n-1} & \\ & & & & p_{n-1} & \\ 0 & & & & & p_n \end{bmatrix}$$

$$\text{Let } A_{t+1} = Q_t^T A_t Q_t \quad \text{then} \quad A_{t+1} = R_t Q_t. \quad (2)$$

For large t , A_t approaches a diagonal matrix, $\text{Diag} [\lambda_i]$, where λ_i 's are the eigenvalues of the matrix, A .

To accelerate convergence of this algorithm, we let

$$A'_t = A_t - K_t I$$

where K_t is the eigenvalue of the 2×2 matrix

$$\begin{bmatrix} c_n^{(t)} & b_{n-1}^{(t)} \\ b_{n-1}^{(t)} & c_n^{(t)} \end{bmatrix}$$

which is closer to $c_n^{(t)}$. Then we decompose $A'_t = Q_t R_t$ and find

$$A_{t+1} = R_t Q_t + K_t I.$$

By repeating these two steps, the eigenvalues of the matrix, A , are immediately available.

A document describing the algorithm with flow chart is being written.

11.12.3.1.2 Inverse Iteration to Find Eigenvectors

A program to calculate the eigenvectors of symmetric tridiagonal matrices with given eigenvalues has been written in XALGOL, debugged and tested. The algorithm is as follows:

A is a symmetric tridiagonal matrix with c_i as its diagonal elements and b_i as its subdiagonal elements. Its eigenvalues are given rather accurately and arranged in descending order.

Let vector $\vec{V}$ be taken as the initial vector. We solve

$$(A - \lambda I) \vec{X} = \vec{V} \quad (1)$$

followed by

$$(A - \lambda I) \vec{Y} = \vec{X} \quad (2)$$

then $\vec{Y}$ is the eigenvector corresponding to eigenvalue λ .

To make the algorithm more effective, we let

$$(A - \lambda I) = LU \quad (3)$$

where L is unit lower triangular and U is upper triangular, set $\vec{V} = L\vec{e}$ where $\vec{e}^t = (1, 1, \dots, 1)$, then

$$U \vec{X} = \vec{e}. \quad (4)$$

Hence $\vec{X}$ can be determined by back-substitution. The matrix U is determined by Gaussian elimination with partial pivoting, i.e. eliminate the variables in their natural order, but select the row having the maximal coefficient of X_i as the pivotal row in the i-th stage. Similarly, we can find the vector $\vec{Y}$.

A document describing the algorithm with flow chart and numerical examples is being written.

11.12.4 Polynomial Root Finder

Actual coding of a root finder has not been done in ASK. However ALGOL codings have been debugged on the B5500. Presently changes are being made in these programs to run on the B6500.

In general, the algorithm has been outlined to be:

(1) evaluate polynomial at various intervals to determine whether a change of sign takes place, (2) shorten these intervals as much as possible, (3) proceed on new intervals with Newton-Rapheson iterations. Also an ALGOL program using Sturm sequences to evaluate intervals is presently being debugged on the B6500.

11.12.5 Identification of Non-linear Differential Equations

Much is known about the numerical solutions of differential equations, however, almost nothing is known about the reverse problem, i.e., given some observed function of time, $x(t)$, can we find a differential equation of which $x(t)$ is a solution?

Such problems exist in economics, chemistry, medicine and many other fields, the equations involved often being nonlinear. In the past solution has been restricted to linear problems or problems involving the estimation of only a few parameters. This has been caused not so much by lack of solution techniques, but by inadequate computing power. ILLIAC IV can alleviate this problem through its high computing speed and the fact that the problem can utilize the array processor efficiently.

Since many different differential equations may have the same solution, it is necessary to restrict the problem to time invariant equations of the form:

$$F(x, x', x'', \dots, x^{(m)}, f_1, \dots, f_n) = 0 \quad (1)$$

where F is assumed to be known,

$$x^{(j)} \equiv \frac{d^j}{dt^j} x(t) \text{ and}$$

$$f_i \equiv f_i(x^{(j_i)}) \text{ are unknown functions}$$

of x or one of its derivatives. The problem is to develop an algorithm to determine the f_i .

$$\text{E.g.} \quad x'' + f(x) = 0.$$

If x were given as $x(t) = \sin t$, we could derive $f(x) = x$. Let $\hat{x}(t)$ be the observed function, defined on the interval $0 \leq t \leq T$.

The solution to (1) depends on the functions f_i and the initial conditions of x and its first $n - 1$ derivatives at $t = 0$.

I.e., $x = x(x_0, x'_0, x''_0, \dots, x_0^{(m-1)}, f_1, \dots, f_n)$
or more briefly $x = x(\underline{x}_0, \underline{f})$.

We define the error functional

$$E(\underline{x}_0, \underline{f}) = || \hat{x} - x(\underline{x}_0, \underline{f}) ||^2$$

where

$$|| y ||^2 = \int_0^T w(t) [y(t)]^2 dt$$

for some weighting function, w . E is a measure of the difference between observed function and the computed one for some $\underline{x}_0$ and $\underline{f}$.

The problem thus reduces to finding $\underline{x}_0$ and $\underline{f}$ such that $E(\underline{x}_0, \underline{f})$ is minimized.

For the numerical solution, $\hat{x}(t)$ is assumed to be observed at discrete, equal intervals and the integrals replaced by summations. It is also assumed that these observations are contaminated by random "noise" so that minimizing E constitutes a least squares solution to the problem.

The functions f_i are approximated over the interval on which they are to be identified, $[a, b]$, by a linear combination of fixed orthonormal functions

$$f(x) \approx \sum_{j=1}^n q_j \Phi_j(x)$$

where $\langle \Phi_i, \Phi_j \rangle_1 = \delta_{ij}$, the Kronecker delta function, the inner product $\langle g, h \rangle_1$ being given by $\langle g, h \rangle_1 = \int_a^b g(x) h(x) dx$. In this implementation the functions, Φ_j , will be splines.

The Solution Algorithm

For simplicity, we consider the problem

$$\frac{dx}{dt} = f(x)$$

Let $f(x) = \sum_{i=1}^n q_i \phi_i(x)$ and $x(0) = x_0 = q_0$. Define the solution of

$$x' = f(x)$$

$$x(0) = x_0$$

to be $x_f(t)$. The dependence on x_0 being implicit.

$$\text{Then } E(f) = E(q) = \|\hat{x} - x\|^2$$

$$= \int_0^T [\hat{x}(t) - x_f(t)]^2 dt.$$

When $E(q)$ is minimized, $\nabla q E = 0$

$$\text{where } (\nabla q E)_i = \frac{\partial}{\partial q_i} E. \quad i = 0, 1, \dots, n.$$

A vector q such that $\nabla q E = 0$ may be found iteratively using the Conjugate Gradient Method [1]:

$$q_{n+1} = q_n + c_n p_n$$

$$\text{where: } p_0 = -(\nabla q E)^{(0)},$$

$$p_{n+1} = -(\nabla q E)^{(n+1)} + b_n p_n, \quad (2)$$

$$(\nabla q E)^{(n+1)} = \nabla q E(q_{n+1}),$$

$$b_n = - \frac{\langle (\nabla q E)^{(n+1)}, (\nabla^2 q E)^{(n+1)} p_n \rangle_2}{\langle p_n, (\nabla^2 q E)^{(n+1)} p_n \rangle_2},$$

$$(\nabla^2 q E)^{(n+1)} = \nabla^2 q E(q_{n+1})$$

and $\nabla^2 q E$ is the matrix of second derivatives:

$$(\nabla^2 q E)_{ij} = \frac{\partial^2}{\partial q_i \partial q_j} E.$$

The superscripts indicate the iteration count. The inner product $\langle a, b \rangle_2$ being defined by

$$\langle a, b \rangle_2 = \sum_{i=0}^n a_i b_i.$$

At each iteration, c_n is chosen so as to minimize $E(g_n + c p_n)$ over all c .

The calculation of $\frac{\partial}{\partial q_i} E$.

$$\begin{aligned} \frac{\partial}{\partial q_i} E &= \frac{\partial}{\partial q_i} \int_0^T w(t) [\hat{x}(t) - x_f(t)]^2 dt \\ &= -2 \int_0^T w(t) [\hat{x}(t) - x_f(t)] \cdot \frac{\partial x_f}{\partial q_i}(t) dt \end{aligned} \quad (3)$$

if we abbreviate $\frac{\partial x_f}{\partial q_i}$ by δx_i , then δx_i satisfies the so-called "sensitivity equations"

$$\frac{d}{dt} \delta x_0 = \frac{\partial f}{\partial x}(x_f) \cdot \delta x_0, \quad \delta x_0(0) = 1$$

and $\frac{d}{dt} \delta x_i = \frac{\partial f}{\partial x}(x_f) \cdot \delta x_i + \phi_i(x_f), \quad \delta x_i = 0, \quad i = 1, 2, \dots, n.$

representing the sensitivity of the equation $x' = f(x)$ to changes in x_0 and f respectively.

If we define $\phi_0(x) \equiv 0$, these may be combined to:

$$\frac{d}{dt} \delta x_i = \frac{\partial f}{\partial x}(x_f) \cdot \delta x_i + \phi_i(x_f) \quad i = 0, 1, \dots, n. \quad (4)$$

To determine $\frac{\partial^2}{\partial q_i \partial q_j} E$ we note that,

$$\begin{aligned} \frac{\partial^2 E}{\partial q_i \partial q_j} &= \frac{\partial^2}{\partial q_i \partial q_j} \int_0^T w(t) [\hat{n}(t) - n_f(t)]^2 dt \\ &= -2 \int_0^T w(t) \frac{\partial}{\partial q_i} x_f(t) \frac{\partial}{\partial q_j} x_f(t) dt \\ &\quad - 2 \int_0^T w(t) [\hat{n}(t) - x_f(t)] \times \frac{\partial^2}{\partial q_i \partial q_j} (x_f)(t) dt \end{aligned}$$

$$\begin{aligned}
&= 2 \int_0^T w(t) \delta n_i \delta n_j dt \\
&- 2 \int_0^T w(t) [\hat{n}(t) - x_f(t)] \delta n_{ij}^2 dt
\end{aligned} \tag{5}$$

where we define $\delta n_{ij}^2 \equiv \frac{\partial^2}{\partial q_i \partial q_j} (x_f)(t)$.

By differentiating (t) with respect to q_j , $\delta^2 n_{ij}$ may be shown to satisfy the equation:

$$\begin{aligned}
\frac{d}{dt} \delta^2 n_{ij} &= \frac{\partial^2 f}{\partial n^2} \delta n_i \delta n_j \\
&+ \frac{\partial f}{\partial n} \delta^2 n_{ij} \\
&+ \frac{\partial}{\partial n} \phi_i \delta n_j + \frac{\partial}{\partial n} \phi_j \delta n_i
\end{aligned} \tag{6}$$

Since we require only $\nabla^2 q$ E explicitly, we need calculate only

the components $\Psi_i \equiv \sum_{j=0}^n \delta^2 n_{ij} p_j$.

Multiplying (6) by p_j and summing over j gives:

$$\begin{aligned}
\frac{d}{dt} \Psi_i &= \frac{\partial^2 f}{\partial n^2} \delta n_i \left(\sum \delta n_j p_j \right) \\
&+ \frac{\partial f}{\partial n} \Psi_i \\
&+ \frac{\partial}{\partial n} \phi_i \left(\sum \delta n_j p_j \right) \\
&+ \left(\frac{\partial}{\partial n} \sum \phi_j p_j \right) \delta n_i
\end{aligned} \tag{7}$$

where all summations are from $j = 0$ to n . This reduces the number of differential equations to be solved from $(n+1)^2$ to $n+1$.

Using (5) we obtain

$$\begin{aligned} [\nabla^2 q E p]_i &= 2 \int_0^T w(t) \delta n_i \times (\sum \delta n_j p_j) dt \\ &- 2 \int_0^T w(t) [\hat{n}(t) - x_F(t)] \Psi_i(t) dt . \end{aligned}$$

Preliminary experiments indicate that this algorithm is rapidly convergent, only 5-10 iterations being required when a good estimate of f is available.

Implementation on ILLIAC IV

A specialized compiler is needed to perform the following tasks:

1. Accept the form of the equation to be identified.
2. By using symbolic differential techniques, form the sensitivity equations.

11.12.6 Estimation and Filtering

11.12.6.1 Non-linear Least Squares Problem

Work is proceeding on the parameter estimation of non-linear least squares problems. The numerical algorithms to be investigated are:

1. modified Gauss method
2. gradient method
3. variable metric method
4. factorization method

The implementation of algorithm (1) has been successfully completed and the ASK code for (2) is proceeding for which more accurate results are expected. Work has also begun on the variable metric method.

11.12.6.2 Numerical Solution of the Non-linear Matrix Riccati Equation

An eigenvector solution of the matrix Riccati equation relating to optimal control theory has been studied during this quarter. For the

cases in which the matrix of coefficients resulting in linear dynamic systems can be transformed to a diagonal matrix, there is a straight forward method for constructing a similarity transformation whether or not the eigenvalues are distinct.

11.13 Linear Programming

This quarter has seen the switch to the B6500 computer installation. Regrettably, the transition involved a degradation of the B5500 assembler-simulator support for a prolonged period, without comparable facilities being available elsewhere. This seven week stretch was used to bring LP-system documentation into usable shape.

The simulation that could be performed was directed wholly toward the INVERT package. There are indications that simulation will not permit sufficient code "execution" to debug the inversion routines, but work is continuing.

B5500 SETUP routines have been run on the new computer, but require modifications and extensive updating to reflect the increasing load of SETUP that the ILLIAC IV will perform. Experience with our B6500 indicates that only the most minimal processing should remain on this easily overtaxed facility. The future will see consideration being given to transferring more functions, including those--such as sorting--which are ungainly and difficult to code for the ILLIAC.

The problem of removing the solution and associated information is now being resolved. Initially these SETDOWN procedures will be executed as a separate program as it is estimated that abnormal terminations will be frequent. Again, as with SETUP, the majority of the work--such as rescaling and transforming data to B6500 representation--will be executed on ILLIAC IV.

Documentation was reviewed, rewritten, and reorganized with the intent of improving the clarity of the system's interconnections and interpresumptions. Not surprisingly, discrepancies were found. Further, refinement will be necessary, but will be deferred until after the code has been run on the prototype ILLIAC.

11.14 Long Codes

An analysis of the noise vectors used for previous studies disclosed an alarming degree of correlation between the supposedly independent random vectors. Consequently, an improved pseudo-random number generator of proven period and potency was substituted for the previously used ad hoc generator. All identification algorithms tested were more effective when used with the less correlated noise vectors.

A Kalman filter algorithm is being implemented to allow testing of the parameters obtained from the identification algorithms without knowledge of the true values of the parameters. If the innovation sequence of a Kalman filter using the estimated parameters as a model and using the observed values as input is a white noise sequence, identification can be considered successful.

The most recently developed correlation approach for identification showed considerable promise before its use was suspended for recoding to run on the B6500. The equations for this scheme, like most other relations that occur in identification processes, include both the covariance matrix of the plant noise and the covariance matrix of the observation noise, which are usually unknown. Manipulation of the correlation algorithm expressions indicates that it will be possible to obtain not only estimates of the parameters of the system, but also estimates of the two covariance matrices. It may be necessary to require the correlation matrix of the observation noise to be in the form $\sigma^2 I$ (I = Identity matrix), but this restriction can probably be relaxed.

11.15 Signal Processing

A Glypnir subroutine has been written which computes the fast Fourier transform of any real vector of length N , where N is a power of two and $N \geq 64$.

A previously written Glypnir autocorrelation program has been put into the form of a subroutine.

Programs are being written for use on the B6500 computer which will be used to thoroughly test previously written ALGOL procedures and Glypnir subroutines.

Work is continuing with Amoco Production Company on implementing some signal processing and partial differential equations algorithms on the ILLIAC IV.

11.16 Education

11.16.1 ILLIAC IV Seminars

In addition to the graduate course on ILLIAC IV, three seminars were offered this past quarter: A one-day seminar was given on January 5, 1971. The outline follows:

Seminar on ILLIAC IV

<u>Background</u>	9:00 - 9:30 am
Buffer, Pipeline and Multiprocessor	
<u>Hardware Structure</u>	9:30 - 12:00 n
A General Description of Array	
A Sample Problem	
A More Detailed Description of Array	
A General Description of I/O System	
<u>Test/Repair Equipment and Diagnostics</u>	1:30 - 2:00 pm
Physical Characteristics	
<u>Software</u>	2:00 - 4:30 pm
Programming Languages -- ASK, GLYPNIR, FORTRAN	
Operating System	
Utilities	
<u>Applications</u>	4:30 - 5:00 pm

On January 11-15, a one-week seminar was given to Pan American Petroleum. On March 15-19, another one-week seminar was given to employees at Ames Research Center, Moffett Field, California, where ILLIAC IV is to reside. Additionally a one and one-half hour overview on ILLIAC IV was presented to about 300 Ames employees. The outline for the one-week seminar follows.

A Seminar on the ILLIAC IV System

I. Background -- Conventional and Unconventional Organizations

- A. Why ILLIAC IV?
 - 1. Conventional Organization
- B. How to speed up the Operation-Design Philosophies
 - 1. Overlap
 - a. Buffer
 - b. Pipeline
 - 2. Replication
 - a. General Multiprocessor -- Distribute, Memory, ALU, CU
 - i. Recentralize Memory
 - ii. Recentralize ALU
 - iii. Recentralize CU -- basis for ILLIAC IV
 - 3. Both
- C. ILLIAC IV is a Vector Processor

II. Hardware Structure

- A. Organization Chart
- B. ILLIAC IV Array -- General Description
 - 1. Control Unit (CU)
 - 2. Processing Element (PE)
 - 3. Data Paths
 - a. Control Unit Bus (CU Bus)
 - b. Common Data Bus (CDB)
 - c. Routing Network
 - d. Mode Bit Line
- C. An Illustrative Problem
 - 1. DO 10 I = 1, N
10 A(I) = B(I) + C(I)
 - a. N = 64
 - b. N < 64
 - c. N > 64
 - 2. DO 10 I = 2, 64
10 A(I) = B(I) + C(I-1)
 - a. Skew at Compile Time
 - b. Skew at Execution Time
 - 3. DO 10 I = 2, 64
A(I) = B(I) + A(I-1)
- D. ILLIAC IV Array -- A More Refined Description
 - 1. PE
 - a. RGD
 - 2. PU
 - 3. CU
 - a. ADVAST
 - b. FINST
 - c. MSU
 - d. TMU
 - e. ILA
- E. Another Illustrative Problem
 - 1. Laplace's Partial Differential Equation describing the Steady-State heat distribution on a slab
- F. Data Allocation

G. ILLIAC IV I/O System

1. I/O Subsystem
 - a. CDC
 - b. BIOM
 - c. IOS
2. Disk File System
3. B6500
 - a. CPU
 - b. Memory
 - c. Multiplexor
 - d. Peripherals
 - i. Remote Terminals
 - e. Laser Memory
 - f. ARPA Network

III. Configurations at CAC and Paoli

IV. Diagnostics and Test/Repair Equipment

- A. IDIAP
- B. PEX
- C. PEMX
- D. CUCT
- E. Some Physical Characteristics
 1. Slides

V. Programming Languages

- A. ASK
 1. Background, Review, Notation, Conventions
 2. Sample Problems
 - a. Summing an Array of Numbers
 - b. Finding the Maximum Value in an Array of Numbers
 - c. Matrix Multiplication
 - i. Skewed Storage
 - d. Temperature Distribution on a Slab
 - i. Case 1: one temperature value per PEM
 - ii. Case 2: eight temperature values per PEM
- B. GLYPNIR
- C. FORTRAN

VI. Operating System

- A. ICL
- B. Utilities

VII. Some Applications

ADMINISTRATION

11.17 Administration and Services

11.17.1 Financial Report

Actual expenditures and obligations for January-March 1971:

	<u>January</u>	<u>February</u>	<u>March</u>
Burroughs Corporation	\$312,000.00	(\$109,000.00)	N.A. *
University of Illinois	183,220.34	240,118.42	\$168,093.51

Expenditures and obligations to date through March 1971:

Burroughs	\$26,723,000.00
University	6,814,273.83

Budgeted expenditures - 3rd quarter, fiscal 1971.

	<u>January</u>	<u>February</u>	<u>March</u>
Burroughs	\$567,900.00	\$298,643.00	\$252,758.00
University	215,302.00	215,302.00	215,302.00

* Monthly status report not received from Burroughs as of 4/30/71, therefore figures for March are not available.

REFERENCES

- [1] Rajan, S. "Numerical Solution of the Partial Differential Equations of Gas Dynamics." Center for Advanced Computation, University of Illinois at Urbana-Champaign, (in preparation).
- [2] Richtmyer, R. D., and Morton, K. W. "Difference Methods for Initial Value Problems." Interscience Publishers, 1967. Pp. 354-365.
- [3] Rutishauser, H. "The Jacobi Method for Real Symmetric Matrices." Handbook Series Linear Algebra, Numerische Mathematik 9, 1-10, (1966).
- [4] Eberlein, P. J., and Boothroyd, J. "Solution to the Eigenproblem by a Norm Reducing Jacobi Type Method." Handbook Series Linear Algebra, Numerische Mathematik 11, 1-12, (1968).

THESES

- Moreno, V. "A Logic Test Generation System using a Parallel Simulator." Master's thesis. Urbana, Illinois: Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.
- Parker, J. L. "Logic per Track Information Retrieval System." Ph.D. thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

DOCUMENTS

- Hashimoto, A., and Stevens, J. "Path Cover of Acyclic Graphs." ILLIAC IV Document No. 239. ILLIAC IV Project, University of Illinois at Urbana-Champaign, (December 24, 1970).

Schuster, Stewart. "An Information Management and Analysis System for ILLIAC IV." CAC Document No. 1. Center for Advanced Computation, University of Illinois at Urbana-Champaign, (December 11, 1970).

Schuster, Stewart. "A Statistical System for ILLIAC IV." CAC Document No. 2. Center for Advanced Computation, University of Illinois at Urbana-Champaign, (December 11, 1970).

Stevens, James, Jr. "Matrix Multiplication Algorithms for ILLIAC IV." ILLIAC IV Document No. 231, Department of Computer Science File No. 855. ILLIAC IV Project, University of Illinois at Urbana-Champaign, (August 26, 1970).

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	5.00
Research Associates	0	0	---
Graduate Research Assistants	2	62	33.00
Graduate Teaching Assistants	0	28	13.11
Professional Personnel	21	0	21.00
Administrative and clerical	25	0	25.00
Nonacademic Personnel (Monthly)	54	0	54.00
Nonacademic Personnel (Weekly)	<u>0</u>	<u>91</u>	<u>37.50</u>
TOTAL	127	184	210.38

*This report does not include personnel employed on the ILLIAC IV Project.

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 first quarter, the following publications were issued by the Laboratory:

Report Numbers

- (1) Ibaraki, T., T. K. Liu, D. Djachan, and S. Muroga, "Synthesis of Optimal Networks with Nor and Nand Gates by Integer Programming," Report No. 427, January 1971.
- (2) "ILLIAC IV Quarterly Progress Report--October-December 1970," Report No. 431, January 15, 1971.
- (3) Koo, Ping L. and Daniel E. Atkins, "Arithmetic Unit of ILLIAC III: Simulation and Logical Design-Part II," Report No. 418, October 28, 1968; Revised and Edited by Lakshmi N. Goyal, November 10, 1970.
- (4) Maruyama, Kiyoshi, "Parallel Methods and Bounds of Evaluating Polynomials," Report No. 437, March 1971.
- (5) McCormick, B. H. and B. J. Nordmann, Jr., D. E. Atkins, R. T. Borovec, L. N. Goyal, L. M. Katoh, R. M. Lansford, J. C. Schwebel, and V. G. Tareski. "ILLIAC III Reference Manual Volume I: The Computer System, Report No. 433, February 17, 1971.
- (6) McCormick, B. H. and B. J. Nordmann, Jr., D. E. Atkins, R. T. Borovec, L. N. Goyal, L. M. Katoh, R. M. Lansford, J. C. Schwebel, and V. G. Tareski, "ILLIAC III Reference Manual Volume II: Instruction Repertoire," Report No. 434, February 26, 1971.
- (7) Read, John, "Show-and-Tell, An Interactive Programming System for Image Processing System Specifications," Report No. 429, February 18, 1971.

Theses

- (1) Adams, Harold Corwin II, "Dynamic Partitioning in the Array Language OL/2," (M.S.), Report No. 421, January 1971.
- (2) Chan, Vivian Wan-Man, "Overlay Supervisory Programs," (M.S.), Report No. 419, February 1971.
- (3) Latch, John Lockhart, "An Algorithm for Evaluating Array Expressions in OL/2," (M.S.), Report No. 422, January 1971.
- (4) Mark, Barbara Drahos, "Machine Independent Compilation of PL/1: Pass I Symbol Manipulation," (M.S.), Report No. 439, June 1971.

12.2 Bibliography (cont.)

- (5) Marshall, Larry Gene, "The Minicomputer: An Educational Tool," (M.S.), Report No. 432, February 1971.
- (6) Muraoka, Yoichi, "Parallelism Exposure and Exploitation in Programs," (Ph.D.), Report No. 424, February 1971.
- (7) Oberbeck, Peter Ernst Rudolf, "ORBIT--Online Reduced Bandwidth Information Transmission," (Ph.D.), Report No. 430, February 1971.
- (8) Reingold, Edward Martin, "On Some Optimal Algorithms," (Ph.D.), Report No. 428, January 1971.
- (9) Shiau, Lih-Er, "Design of Optimal One-Bit Adder Networks By Interger Linear Programming," (M.S.), Report No. 425, January 15, 1971.
- (10) Taranto, Richard Joseph, "Numerical Studies of Stone's Symmetric Factorization and the Iteration Parameters, α and ,", (M.S.), Report No. 423, January 1971.
- (11) Winterbauer, Albert Lee, "PLONE: A Load-and-Go Compiler For A Basic Subset of PL1," (M.S.), Report No. 426, January 15, 1971.
- (12) Yamada, Hirohide, "Emulation of Disc File Processor," (M.S.), Report No. 436, June 1971.

12.3 Colloquia

"On the Use of Functional Reasoning in Software Design," by Dr. Peter Freeman, Carnegie-Mellon University, Pittsburgh, Pennsylvania, February 8, 1971.

"The Anatomy of a Microprogrammed Communication Processor," by Professor James E. Vander Mey, Computer Science Department, The Pennsylvania State University, University Park, Pennsylvania, February 11, 1971.

"An Application of Automata Theory to the Isomorphism Problem for Planar Graphs," by Professor John Hopcroft, Computer Science Department, Stanford University, Stanford, California, February 15, 1971.

"Transducing Automata and Formal Languages," by Dr. Johannes J. Martin, 18835 Apache Drive, South Bend, Indiana, February 22, 1971.

"A Practical Model for Generating Code," by Mr. Thomas Wilcox, Department of Computer Science, Cornell University, Ithaca, New York, February 25, 1971.

"The Future of Instructional Computer Use," by Dr. Roger E. Levien, The Rand Corporation, 2100 M Street, N. W., Washington, D.C., March 8, 1971.

"Data Compression - Tradeoffs and Implications," by Mr. Ross Overbeek, Department of Computer Science, Pennsylvania State University, State College, Pennsylvania, March 11, 1971.

"Conversational Systems Programming," by Mr. Rudolph Krutar, Carnegie-Mellon University, Department of Computer Science, Schenley Park, Pittsburgh, Pennsylvania, March 15, 1971.

"Computing Polynomials with a Minimum Number of Operations," by Professor Albert R. Meyer, Project MAC, 545 Technology Square, Cambridge, Massachusetts, March 22, 1971.

"Learning to Use Contextual Patterns in Language Processing," by Mrs. Sara Jordan, Computer Science Department, University of Wisconsin, Madison, Wisconsin, March 25, 1971.

"Laser Displays and Related Research at GTE Laboratories," by Mr. Vernon J. Fowler, Bayside Research Center of GTE Laboratories Incorporated, Bayside, New York, March 29, 1971.

12.4 Drafting

During the first quarter, a total of 232 drawings were processed by the general departmental drafting section:

Large Drawings	67
Medium Drawings	16
Small Drawings	86
Layouts	9
Report Drawings	20
Changes	22
Miscellaneous	<u>12</u>
Total	232

(M. Goebel)

12.5 Shops' Production

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

	<u>AEC 2118</u>	<u>AEC 1469</u>	<u>Other</u>
Machine Shop	2	7	3
Electronics Shop	2	74	31
Chemical Shop	2	63	6
Layout Shop	1	56	9

(F. P. Serio)

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-1469-0182

2. TITLE

QUARTERLY PROGRESS REPORT - JANUARY-MARCH, 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 W. J. Poppelbaum
Principia Investigator
Hardware Research Group

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature

W. J. Poppelbaum

Date

March 31, 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.

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-1469-0182

2. TITLE

1st QUARTERLY REPORT 1971 (January, February, March)

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

Signature

Charles W. Gear

Date March 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.

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-2118-0007

2. TITLE

QUARTERLY PROGRESS REPORT - JANUARY-MARCH, 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

Bruce H. McCormick

Date

April 15, 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.