C00-1469-0196

C00-2118-0025

MASTER

QUARTERLY TECHNICAL PROGRESS REPORT

July, August, September 1971

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

July, August, September 1971

UIUCDCS-71-3

NOTICE

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

Rey

TABLE OF CONTENTS

	Page
0. DISTRIBUTION OF DEPARTMENT OF COMPUTER SCIENCE REPORTS	1
1. CIRCUIT RESEARCH.	2
1.1 Bundle Processing.	3
1.1.1 Bundle Repeater and Restorer.	3
1.1.2 SABUMA.	3
1.2 APE.	5
1.2.1 Progress in General	5
1.2.2 The 32-Bit Control Code Generator	5
1.3 PENTECOST.	10
1.3.1 Monitor Modification.	10
1.3.2 Screen Voltage.	10
1.3.3 Focus Voltage	10
1.3.4 Horizontal Deflection	11
1.3.5 Vertical Deflection.	11
1.3.6 Video Circuit	11
1.3.7 PENTECOST camera.	11
1.4 Ergodic.	14
1.4.1 Outline	14
1.5 Telemaze	15
1.5.1 Introduction.	15
1.5.2 X,Y Coordinate Control Logic.	16
2. HARDWARE SYSTEMS RESEARCH	20
2.1 OLFT	21
2.1.1 Operation of Intermediate Cooled System.	21
2.1.2 Optical System Design	21
2.2 Tricolor Cartograph.	22
2.2.1 Circuits.	22
2.3 BLAST.	29
2.3.1 Screen Signal	29
2.3.2 Dynamic Scan Correction	29
2.4 Sematrix.	31
2.4.1 Summary of Project.	31
2.4.2 Project Status.	32
2.5 LINDA.	32
2.5.1 Summary of the Project.	32
2.5.2 Project Status.	33
2.5.3 Future Work	33
2.6 Stereomatrix	37
2.6.1 Transformer	37
2.6.2 Coefficient Generator	37
2.6.3 The Display	38

	Page
2.7	Scantrix 38
2.7.1	4-Bit A/D Converter 38
2.8	FROG 40
2.8.1	Project Outline 40
2.8.2	Basic Elements of the Adaptive Filter. 43
2.8.3	Design Progress 46
3.	SOFTWARE SYSTEMS RESEARCH 49
3.1	Numerical Processes. 50
3.1.1	Ordinary Differential Equations 50
3.1.2	Sparse Matrix Inversion 55
3.1.3	The Steady-State Package. 57
3.1.4	Plot Package. 61
3.1.5	Numerical Package 66
3.1.5.1	Elimination. 66
3.1.5.2	SETUP. 66
3.1.5.3	SPARSE 67
3.1.5.4	MATSET 69
3.1.5.5	COMPIL 70
3.2	Non-Numerical Packages 71
3.2.1	Item Analys 71
3.3	Graphical Rempte Access Support System (GRASS). 73
3.3.1	Disk Monitor System 73
3.3.2	Display Terminals 73
3.3.3	Information Retrieval 74
3.3.4	Monitors. 74
3.4	Computer Maintenance and Construction. 79
3.4.1	Graphics-8 Hardware 79
3.4.2	Equipment Maintenance Log Summary 81
3.4.3	Stereomatrix Interface. 82
4.	IMAGE PROCESSING AND PATTERN RECOGNITION RESEARCH: ILLIAC III. 84
4.1	INTERACTIVE PICTURE PROCESSING 85
4.1.1	Show-and-Tell 85
4.1.2	Atlas Extension of Show-and-Tell. 85
4.1.3	Terminal Communications 90
4.1.4	Scan/Display Devices. 90
4.2	PARALLEL PROCESSING. 93
4.2.1	Extensions of Signal Detection Theory 93
4.2.2	Interval Covering 93
4.2.3	PAX Language Support. 94
4.3	STRUCTURAL INFERENCE 95
4.3.1	Scene Segmentation by Clustering. 95
4.3.2	SOL Language Support. 95

	Page
4.4 APPLICATIONS	97
4.4.1 Cervical Smears	97
4.4.2 Brain Mapping	97
4.4.3 Cytospectrometer.	98
4.4.4 Remote Manipulation	98
4.5 COMPUTER SYSTEMS	100
4.5.1 IBAL Assembler.	100
4.5.2 Operating Systems	101
4.5.3 TM II	101
4.5.4 TP, AU, and IOP	102
4.6 DOCUMENTATION.	104
4.6.1 External Documents Issued	104
4.6.2 Logic Drawings Issued	104
4.6.3 Engineering Drafting Report	105
4.7 ADMINISTRATION	106
4.7.1 Personnel Report.	106
4.7.2 Computer Usage Log Summaries.	106
5. ILLIAC IV	107
5.1 Operating System	109
5.2 COCKROACH.	109
5.3 GLYPNIR.	110
5.4 ILLIAC IV Assembler (ASK).	110
5.5 Documentation.	111
5.6 Burroughs B6500 Facility at Illinois	111
5.7 Financial Report	113
6. SWITCHING THEORY AND LOGICAL DESIGN	115
7. SOUPAC.	119
8. MACHINE AND SOFTWARE ORGANIZATION STUDIES	121
8.1. Fortran Parallelism Detection.	121
8.2. Tree Machine Simulator	122
8.3. Weighted Node, Directed, Acyclic Graph Schedule by m-Machines	122
8.4. Simulation Processor	123
8.5 Memory Hierarchies	124
9. COMPUTER SYSTEMS ANALYSIS	125
9.1. Computer Network Modeling.	125
9.2. Center Throughput Analysis	125

	Page
10. THEORY OF DIGITAL COMPUTER ARITHMETIC.	127
10.1. Continued Fraction Arithmetic.	127
10.2. Continued Fraction Algorithm for the Square Root.	129
10.3. Evaluation of Some Elementary Functions in Radix 16	130
11. COMPUTER LANGUAGES FOR MATHEMATICS AND NUMERICAL ANALYSIS	131
11.1. COMPUTER LANGUAGES:.	131
11.2. NUMERICAL ANALYSIS	131
11.3. ALGORITHMS:.	132
12. NUMERICAL METHODS, COMPUTER ARITHMETIC AND ARTIFICIAL LANGUAGES	135
12.1 An Adaptive Algorithm for the Solution of Finite Difference Equations	135
12.2 ETS Operating System	136
12.3 Computer Aided Instruction	137
12.4 Publications	137
12.5 PDP-11 Hardware Additions.	137
12.6 General Programming.	138
12.7 SPOOLING	139
12.8 Assembly Language Instruction.	139
13. GENERAL DEPARTMENT INFORMATION	141
13.1 Personnel.	141
13.2 Bibliography	142
13.3 Colloquia.	144
13.4 Drafting	145
13.5 Shop's Production.	145

O. DISTRIBUTION OF DEPARTMENT OF COMPUTER SCIENCE REPORTS

In order to achieve greater efficiency and more effective dissemination of information, some revisions in the method of distributing reports from the Department of Computer Science are being put into effect with the opening of the 1972 fiscal year.

The Quarterly Technical Progress Report will be the primary medium by which information about the department is disseminated. Normally, a person requesting the reports of the department will be placed on the mailing list for this report. Individual mailing lists for all the reports generated in the department or in any group thereof will be held to a minimum. However, each report generated will be listed at the end of the section pertaining to each group. A combined list of all of these reports will appear toward the end of the Quarterly Technical Progress Report. An addressed request sheet is also provided at the end of this report by which a selected number of departmental reports of interest may be requested.

The series of file numbers generated by the department normally contains material useful internally. In the rare event that one of these might be useful outside of the department, a request should be submitted directly to the person concerned.

Approximately once per year an inquiry form will be provided in the Quarterly Technical Progress Report via which the department can ascertain whether or not the recipients of the report wish to remain on the mailing list.

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

The bundle repeater/restorer for the system that has been described by Bernard Tse is now being built, and the system should be complete shortly. SABUMA (Safe Bundle Machine) is also nearly finished; its properties are summarized by Trevor Mudge. Yiu Wo's report on APE (Autonomous Processing Element) deals mainly with the control unit and, in particular, with a control code generator for transmitting setup information to the assembly of elements. Panigrahi describes the transmit/receive sections of PENTECOST, the color display that uses the two color Land scheme, a "filtered" black and white TV camera and a Penetron display tube. In addition, two new projects are outlined in this section. Ergodic, tackled by Jim Cutler, aims to build a stochastic processor which combines both spatial and temporal aspects of random sequence coding to add redundancy to a system. The second of these is Telemaze, which uses a "world model" approach to get around unacceptably long feedback delays in remote space control systems. Basic to this is the idea that an earthbound model of the environment of a remote object (planetary lander) need be updated relatively infrequently by the controller, thus eliminating the need for the object to send back environmental information at all times. Edward Pott is the mission controller.

M. Faiman (ed.)

1.1 Bundle Processing (Project No. 21)

1.1.1 Bundle Repeater and Restorer

Construction of the bundle repeater/restorer is underway. Supplies for +5V and -5V had been acquired. The logic circuit for the repeater/restorer stages will be laid out by the shop since 384 similar circuits will be required.

It is expected that the repeater/restorer system will be completed during the present quarter.

Bernard Tse

1.1.2 SABUMA

Summary of Project:

This system demonstrates the feasibility of failsafe information interchange and processing using a bundle representation. The technique may be summarized as follows. Numbers in the range $-50 \leq n \leq 50$ are represented by the sum of the potentials on a bundle of 100 wires. A wire is weighted -1, 0, or +1 according to whether it is at a negative potential, open circuit (broken), or at a positive potential. Thus, $n = 32$ would be represented by having 32 more wires at a positive potential than at a negative one.

The information interchange is made failsafe by using these bundles in pairs, to represent numbers by the quotient of the two numbers represented by the two bundles. The new number generated is thus in the range $-1 \leq N \leq 1$. If one bundle represents X and the other, Y, then

$$N = \frac{X}{Y}$$

If the probability of a wire going open circuit, i.e. breaking, is P, then by the law of large numbers we have on average

$$N^1 = \frac{P.X}{P.Y} = N$$

i.e. failsafe action.

Furthermore assume we wish to perform arithmetic with two numbers represented in this bundle technique, say $A = \frac{X_1}{Y_1}$; $B = \frac{X_2}{Y_2}$; in general we need

(i) $A + B$

(ii) $A - B$

(iii) $A * B$

(iv) A/B

These are formed as follows:

(i)
$$\frac{X_1 Y_2 + Y_1 X_2}{Y_1 Y_2}$$

(ii)
$$\frac{X_1 Y_2 - Y_1 X_2}{Y_1 Y_2}$$

(iii)
$$\frac{X_1 X_2}{Y_1 Y_2}$$

(iv)
$$\frac{X_1 Y_2}{Y_1 X_2}$$

Addition between two bundles is trivial - form the new sum bundle from half of each of the addend & augend bundles. Subtraction requires the value of a bundle to be complemented i.e. positive potential must be made negative and vice versa. This is performed electronically. Finally multiplication must be achieved; this is done using a 3-valued logic gate having the truth table shown.

	+	0	-
+	+	0	-
0	0	0	0
-	-	0	+

TRUTH TABLE

FOR MULTIPLIER.

Project Status:

Construction is nearly complete. The remaining task is to install the displays. Subsequently to this the hardware must be debugged.

Trevor Mudge

1.2 APE (Project No. 25)

1.2.1 Progress in General

During the past quarter, much effort was put into the design and construction of the control unit. Design work on most of the circuits has been completed. Orders on most of the required active circuit elements, components, power supplies, cabinet and mounting hardware have been made. Printed circuit cards for the control unit are now being produced. In the following section a control code generator of the control unit is described in detail.

1.2.2. The 32-Bit Control Code Generator

An important function of the control unit is to send out information to each APE as to what operation it is to perform on its input data, as well as from what channels its inputs are tuned to acquire data. This information is coded into a binary pulse width modulated signal which is later used to modulate an RF signal for transmitting through a whip antenna. The function of the 32-bit control code generator is to produce these pulse width modulated control signals. The block diagram of this generator is given in Figure 1. It consists of a pulse sequence generator, a 5-bit counter, a 32-bit multiplexer, and a monostable multivibrator. The pulse sequence generator produces a sequence of 32 pulses when a starting signal is received. The 5-bit counter and the 32-bit multiplexer are connected to form a parallel to serial converter. The 32-bit control words are fed to the data inputs of the multiplexer. Then the 32 inputs of the multiplexer are gated sequentially to the output. The pulse sequence from the output of the multiplexer is used to modulate the timing network of the monostable multivibrator by switching an auxiliary timing resistor in and out of the timing network. The output

Figure 1. Block Diagram of the 32-bit Control Code Generator.

from the monostable multivibrator produces, therefore, a binary pulse width modulated signal. Figure 2 shows the actual circuit of the 32-bit control code generator. Two SN74150Ns are employed to form the 32-bit multiplexer. The diode D_1 is used to switch the $4.7k\Omega$ auxiliary timing resistor R_T in and out of the timing network. The ready indicator circuit is to turn on an indicator light when the 32-bit control code generator has completed sending out a full control word and is ready for loading with another word for another APE channel. Figure 3 shows a control word and the corresponding binary pulse width modulated control signal. Error detecting and correcting codes could be used for the control word if power is available for the more complex decoder in the APE. At the present level of 100mW available power to an APE, these desired coding schemes are not incorporated into the machine.

Yiu Wo

NUMBERS IN CIRCLE INDICATE THE PIN NUMBER OF IC

Figure 2. Circuit of 32-bit Control Code Generator.

Figure 3. Control Word and Corresponding Pulse Width Modulated Signal.

1.3 PENTECOST (Project No. 31)

1.3.1 Monitor Modification

The black and white Conrac monitor is being modified to permit installation of the Penetron tube. This is an RCA development type dual color phosphor screen, 17"-rectangular, 70° magnetic deflection. The screen of the Penetron consists of a white transparent layer of phosphor over which a red layer of phosphor is deposited. At a screen voltage of 11KV the electrons excite the red layer only. At 16KV the electrons mostly penetrate through the red layer and excite the white layer of phosphor producing a white approximately equivalent to CIE illuminant "C". This display tube employs a single high-voltage electrostatic focus electron gun and the focus voltage has to be switched from 1450-2100 to 1850-2700. Likewise the vertical sweep amplitude and horizontal sweep amplitude have to be switched by a factor of $\sqrt{16/11} \approx 1.2$. The brightness level of white phosphor is twice that of red phosphor. So with the change of screen voltage, the video gain of the amplifier and the cathode to grid bias voltage have to be switched.

1.3.2 Screen Voltage

It is planned that the Conrac monitor screen voltage, which is regulated at about 19KV, will be switched to provide the screen voltage for the Penetron. A high voltage switch, fast enough to change the screen voltage during the vertical retrace time, is being built.

1.3.3 Focus Voltage

The focus voltage supply of about 4KV will be obtained from the 19KV monitor high voltage by a resistance divider network. This voltage also will be switched by a high voltage switch. In the Conrac monitor the screen voltage is about 400V, to which a dynamic focus correction voltage will be added.

1.3.4 Horizontal Deflection

Since we are using the high voltage output derived from the horizontal sweep circuit, we cannot employ any potentiometer switch network at the input or output of the horizontal output tube, because this would not only change the horizontal sweep amplitude but also would vary the high voltage output. So we will place a variable-core coil, similar to the width coil, across the yoke coil. This coil will be switched in and out of the yoke circuit to switch current amplitude in the horizontal yoke.

1.3.5 Vertical Deflection

The gain of the vertical sweep output tube can be changed by switching a resistance input network.

1.3.6 Video Circuit

A switched resistance network will be used to vary the output signal from the contrast control. The grid to cathode bias voltage will be switched by using a similar resistance network at the brightness control.

A block diagram of the Pentecost receiver is shown in Figure 1.

1.3.7 PENTECOST Camera

A red-extended Plumbicon tube (Amperex 16 x QRIG) has been received. This will be installed in GE T-26 camera with a few minor modifications. A filter wheel consisting of alternate red and green filters will be driven by a synchronous motor. Elaborate motor drive circuitry will be needed to synchronize the wheel with the vertical fields. A photosensor detects the position of the wheel and compares this with the vertical sync and makes a correction so as to drive the motor synchronously. The field indicator sync is obtained at this point. A block diagram of the camera system is shown in Figure 2.

Figure 1. PENTECOST Receiver

Figure 2. PENTECOST Color Camera Camera with Filter Wheel

The possibility of using a voltage controlled electronic filter instead of the filter wheel is being looked into.

G. Panigrahi

1.4 Ergodic (Project No. 39)

1.4.1 Outline

This project stems from two previous and completed projects done by this group: BUM and RASCEL. Both of these were concerned with representing numbers by a random process and doing arithmetic operations on these numbers. In RASCEL numbers were represented by a random time sequence of binary voltage levels ("1" and "0") along a wire. Thus, it may be said that this scheme has a time average of a wire being at "1" or

$$\text{time average} = \frac{\text{amount of time that wire is at "1"}}{\text{total observation time}}$$

Now consider a bundle of wires in which every wire has a random sequence of binary voltage levels. If the time average of any wire is the same as the space average of the bundle, then we call this an ergodic bundle.

The goal of this project is to build a system using an "ergodic generator" that will consist of checkpoints; i.e., a unit that will add, subtract, multiply and divide the numbers represented by two ergodic bundles.

1.4.2 Project Status

Since this project has just started, progress has been at a minimum. Design work is now being done for a particular ergodic generator. The major concern will be the ergodic generator. Two techniques will be considered for the ergodic generator and each will be tested and evaluated.

Jim Cutler

1.5 Telemaze (Project No. 41)

1.5.1 Introduction

It is well known that feedback for on-line operation is only useful if the delay in the loop is short enough. A typical example where feedback is nearly useless is the control - from earth - of an exploratory vehicle on a remote planet: either we have to go so slowly that the vehicle hardly moves in "one feedback time", or we must risk destruction by unknown obstacles. It is now possible to imagine a system in which all that is known about the environment of the vehicle (i.e. an environmental "model") is stored in a computer on earth. Guidance is then obtained by steering an "image" of the vehicle through the model. Two cases arise: either the real vehicle (which follows with some delay the "image" in the model on earth) does not pick up any unexpected information - e.g. the position of an obstacle. Then we can happily continue through our "imaginary world" in the computer. Or, and this is the important case, an obstacle does occur. The vehicle then comes to a standstill, sends out a "hold it" signal to earth and updates the model in the computer. The driver will go back to the position before the "hold it" signal came in and continue his "excursion" through the updated model. The important point is that in general he will have a pretty good model right from the beginning (previous photographs, etc.) and that the update process will only happen relatively rarely. Thus, the vehicle can proceed at a reasonable speed, losing only a minimum amount of time due to feedback delays.

The goal of TELEMAZE is to design a simple system with an adjustable feedback delay, obtained by encoding the information on a magnetic tape loop, the latter running at a speed the operator can select. To come close

to the "vehicle problem" described above, we use a 16 x 16 array of squares, a "black" square corresponding to an obstacle, a "white" one to a free path. The arbitrarily chosen (and perhaps modifiable - while running!) "maze" of occupied and unoccupied squares has the "vehicle" projected by a flying spot scanner. This is a point obtained from the "image" (inscribed by an X, Y position tablet) after due delay. The transfer is via two buffers, as shown in Figure 1. Clearly it is possible for the Remote Buffer to be quite a bit behind the "image" positions stored in the Local Buffer.

1.5.2 X, Y Coordinate Control Logic

The basic function of the X, Y coordinate processor, Figure 2, is to take trajectory information from the input tablet and store it in a buffer until it can be transmitted to the remote vehicle. The tablet contains the logic necessary to resolve the path into discrete X and Y coordinates, by energizing one of 16 X, Y wires. This information is then encoded into a 4-bit binary code for each axis. The displacement logic will compare the previously processed word (8 bits) with any non-zero requests. This is necessary to ensure a continuous trajectory. Having successfully passed as a new point, a command is then given to write the new word into the large buffer. To accomplish this, the address of the last word entered into the buffer is found in the write address counter. It will now count up by one address (the buffer is implemented as a circular memory) and place the new word into the buffer. Reading this information back out of the large buffer is completed by the "Read" logic. After a word is read out of the buffer, the read buffer counter will count up by one address. Reading this information out will be slower than writing because of parallel to serial conversion

Figure 1. Telemaze Model Feedback System

Figure 2. Block Diagram of (X, Y) Coordinate Control Logic

needed for transmission. To allow for the difference in these time intervals, additional logic for timesharing the buffer has been added. If both read and write circuitry are active, control is passed alternately to each as long as they both request. If there are no new points from the tablet to be located in the buffer, the reading circuitry will have full control and vice versa.

Data points read out of the buffer are then processed and applied to magnetic tape. By controlling the tape speed, transmission delay time can be varied.

Edward Pott

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

Doug Sand's OLFT report deals with preliminary experiments in operating the crystal transducer at reduced temperature. Don Hanson has some new circuit diagrams for the Tricolor Cartograph. Larry Wallman discusses the problems of dynamic scan correction for BLAST, the stereo TV display project. Sematrix has the design of its block detection circuitry almost complete. Trevor Mudge also describes progress on the construction of the arm and the computer interface logic. Dick Blandford writes about the fiber optics and linear photocell array which LINDA uses to determine the identity of polygonal figures. Shiv Verma and Steve Whiteside are putting the finishing touches on Stereomatrix; their reports deal with the transformer, coefficient generator and display. The Scantrix project entails line-at-a-time storage for a LED, large screen display. Sik Yuen gives details of the A/D converter for incoming video information. Finally, Debasish Bose writes about preliminary work on FROG. This is a new project under the supervision of Professor Ray which aims to design an autonomous mini-robot that can adapt to its environment.

M. Faiman (ed.)

2.1 OLFT (Project No. 12)

2.1.1 Operation of Intermediate Cooled System

Exploratory operation of the intermediate cooled Pockels tube continues, with testing of multiple-frame writing and manual pulsed erasure at substrate temperatures down to -30°C . The apparent charge-decay time-constants are about as anticipated, with a longitudinal τ on the order of two minutes at -30°C . The transverse, or surface τ appeared to be somewhat shorter, but could not be estimated due to image resolution limitations. The low resolution was caused by electronic components: the sweep circuits again contained excessive noise, and the isolation amplifier could not compensate for the large dc-level change caused by frame-gating. The effects of these and other electronic sources of image degradation have been minimized, and testing at lower temperatures will resume.

2.1.2 Optical System Design

Some of the detailed design for the required optical components has been completed. However, a fundamental modification is being considered which will simplify system construction and alignment. The change is the use of a parabolic mirror, rather than a lens, to form the Fourier transform. In addition, due to the high cost of commercial high-resolution positioning devices, several alternatives suitable for in-house construction are being considered, such as differential screws and piezoelectric translators.

Doug Sand

2.2   Tricolor Cartograph (Project No. 16)

2.2.1   Circuits

Recent work has been devoted to designing and building special circuits for the modified Tricolor Cartograph. These are as follows:

Figure 1.   Video Distribution Amplifier

Figure 2.   Horizontal Differentiator

Figure 3.   Synchronous Chopper

Figure 4.   Corning Delay Line Circuit

Figure 5.   Vertical Differentiator

Figure 6.   Video Gate

Don Hanson

Figure 1. Video Distribution Amplifier

Figure 2. Horizontal Differentiator

- NOTES :**
1. P1 and P2 are Frequency Controls
  2. NAND Gates : SN7400N
  3. Pin 2 is Ground
  4. Circuit Built on 1469-218 Card

Figure 3. Synchronous Chopper

Figure 4. Corning Delay Line Circuit

Figure 5. Vertical Differentiator

1469-554

NOTES

1. Q1: 2N2905A
Q2: 2N3905
Q3, Q4: 2N2219A
D1 - 1N995
D2, D3 - 1N4151
D4, D5, D6 - 1N4148
2. PINS 1, 2, 4, 6, 8, 10, 16 are ground.
3. INVERTER: SN7406N

Figure 6. Video Gate.

2.3 BLAST (Project No. 19)

2.3.1 Screen Signal

It was reported last time that the internal connection to the interwoven conducting stripes had broken down. It was conjectured then that this happened because of the "sandwich" type connection made to the nickel conductors on the lenticular screen. The tube manufacturer was contacted during this quarter and he indeed verified this conjecture. The precaution they had taken to prevent this was to wrap aluminum foil around the connection to give a greater contact area.

2.3.2 Dynamic Scan Correction

The scan correction required is different from the normal linearity correction used in cathode ray tube displays. The latter ensures that the raster is rectangular when viewed from the front. This correction is required since, with a linear yoke current, the raster will appear rectangular when viewed from the front only if the tube screen is the cap of a sphere concentric with the deflection center. Since most tubes have screens flatter than this, a correction is required. This is shown in Figure 1. For BLAST the requirement is that the horizontal scanning speed be constant since the video chopper-mixer frequency is constant. Calculation shows that the horizontal yoke current for a constant velocity scan is given by

$$I_H = k_1 \sin [(k_2 \cos \phi) t]$$

where k_1 and k_2 are constants, ϕ is the vertical deflection angle and t is time. Also, ϕ is related to the vertical deflection current I_V by

$$\sin \phi = k_V I_V$$

with k_V constant. But since ϕ is small, its sine can be well enough approximated by the argument, so that the horizontal current may be written

$$I_H = k_1 \sin [(k_2 \cos k_V I_V) t]$$

d_u : uncorrected deflection

d_c : corrected deflection, projection of
deflection on concentric screen onto
flat screen

I_u : uncorrected deflection

I_c : corrected deflection

O : Deflection center

Figure 1. Normal Linearity Correction

In designing the yoke driver several things have to be considered. The driver must be linear so that the current can be controlled precisely. This precludes using a flyback type of drive circuit since control is not maintained over the entire cycle. The severest requirements occur during retrace since only 10 microseconds are available to retrace the yoke from plus 300mA to minus 300mA. The horizontal yoke has an inductance of approximately 10mH. This would require at least -600v to retrace in 10 microseconds. Therefore it was decided to use external dynamic correction coils which have a lower inductance and a lower current requirement since the horizontal yoke would do the major deflection. A feedback type of coil driver is used and has been laid out on printed circuit cards. Some parasitic oscillations have occurred in the printed circuit cards which did not occur in the breadboard model. Currently these are being eliminated.

Larry Wallman

2.4 Semantrix (Project No. 28)

2.4.1 Summary of Project

Semantrix is intended to serve as an I/O device to a general purpose computer. The combined system may then be used to investigate the feasibility of the "World Model" approach to imbedding intelligent behavior in an electronic system.

Semantrix has a flat table top (3' x 3') upon which colored blocks may be placed at random. Up to 64 of these blocks may be used simultaneously and their X-Y coordinates are transmitted to either a computer or a TTY upon receipt of a suitable request. This is accomplished automatically at speeds compatible with either a TTY or a typical computer. The center of each block can be located to within a radius of 0.4" of the true center. Together with this Semantrix has an electromechanical arm which can move blocks to prescribed

points on the board. These points and the blocks to be moved can be specified by the TTY or the computer. Hence, either can request the position of any block and is also capable of altering its position.

The first investigation to be undertaken is to test the "correctness" of a world model for landscape scenes. Given a random array of colored blocks (blue, green, white, red, etc.) an 8 x 8 picture is synthesized by the world model (a Fortran program in the computer) through Sematrix.

2.4.2 Project Status:

The project subdivides into 3 main areas:

1. Block location
  2. Electro-mechanical arm
  3. Logical interface with both a computer and a TTY.
1. Most of the circuitry associated with this part completed in its final form, and is being installed in the table.
  2. With the recent arrival of a pair of suitable torque-proportional-to-input voltage motors, construction has started and is expected to be complete by the end of the year.
  3. Most of the logic has been fabricated, however debugging must await the arrival of the TTY.

Trevor Mudge

2.5 LINDA (Project No. 28)

2.5.1 Summary of the Project

LINDA is a attempt to build a LINE Drawing Analyzer capable of identifying a small number of simple line drawings. A given drawing is projected on a screen by means of a flying spot scanner. Edge information is extracted

by means of photocells. The photocells are arranged in a linear array and discontinuities in the figure are of prime interest (see previous reports for an explanation of the details of the procedure used).

2.5.2 Project Status

During the past quarter work has been directed at building a linear array having a suitable resolution. An array has been built which incorporates fiber optic light pipes (see Figure 1) and dual element photocells. The light pipes allow closer spacing than could be accomplished with photocells alone. There are 52 light pipes in about $2 \frac{3}{4}$ inches. The ends of the pipes touch the screen of the CRT. When a figure is swept past the light pipes the ordinate sum is taken and differentiated twice to produce pulses corresponding to discontinuities in the figure. The amplifier cards for the photocells are being built and at this time a complete test of the arrangement has not been made. Preliminary tests using only 10 photocells on a similar system indicate that the present arrangement will be satisfactory.

2.5.3 Future Work

In the coming quarter tests will be carried out to determine the resolution of the photocell light pipe arrangement and improvements will be made if necessary.

Work will also be directed at developing the logic for a system which can scan a pattern, separate it into its simple parts, and sweep these parts, one by one, across the linear array for identification. As a simple example, consider the "house" shown in Figure 2. The system logic will first determine that there are two simple patterns in this figure. It will then color one pattern so that it is all one color rather than a simple line drawing. The second pattern will then be erased and the solid colored figure

Figure 1. Photocell Arrangement Showing
Fiber Optic Light Pipes for Linear Array

Figure 2. Steps in Identification of a "House"

will be swept across the linear array for identification. It will then return this identified pattern, erase it, color in the second pattern and identify it. A triangle over a square in this figure will be called a "house". Many other figures are possible. At this time a vocabulary of about 25 figures are being considered. This will probably be expanded later.

Dick Blandford

2.6 Stereomatrix (Project No. 30)

2.6.1 Transformer

The transformer has been in operation since July. Rotation and translation coefficients from the coefficient generator have been supplied to the coordinate transformer unit and the X, Y, Z outputs of PAGAN have been used as the input to the system. A left eye view only (X_L , Y) of the PAGAN pattern obtained from the transformer was displayed on the CRT. The change in perspective could be observed by rotating the pattern about the coordinate axes or by translating it or by changing the observer's position.

Thus, all the features of the transformer have been very well tested. The scaling circuit has been modified and a new card has been made. The gain at the output of the system has been adjusted to make the transformer compatible with the input to the deflectors.

The driver and receiver cards needed to interface the transformer with the PDP-8I have been wired and are being tested. Picture and display control logic needed to communicate with the PDP-8 are being wired.

Thus, it is hoped that the transformer will be interfaced and tested with software from the PDP-8 in the next quarter.

2.6.2 Coefficient Generator

Dick Cheng has finished his thesis and left. However, there is a lot to be accomplished in the coefficient generator. It is expected that we will soon have the wiring and logic diagram of the coefficient generator checked and that the system will be working shortly.

Logic design for the digital read out of all twelve coefficients has been completed and cards have been laid out and tested.

Shiv Verma

2.6.3 The Display

The rear projection screen has arrived and does give a reasonable extinction ratio (of proper to spurious polarization component). The extinction is not quite as good as with the aluminized lenticular front projection screen, but it is quite acceptable. The shop is building a wooden frame for the screen to bolt on the viewing desk. The rear projection screen seems to allow much more front lighting without "washing out" the picture. Some figures could easily be viewed in the normally lighted room. With the good quality, coated optics that have been ordered there is some hope that the system will perform well enough to be used with normal room lighting.

The deflector has just been returned from the vendor and remains to be tested. The frequency instability problem was reportedly in the wideband power amplifier.

We hope to achieve three-dimensional pictures during the next quarter.

Steve Whiteside

2.7 Scantrix (Project No. 35)

2.7.1 4-Bit A/D Converter

A 4-bit analog-to-digital converter was built which would be used to change the video signal from a TV into its digital equivalent. By doing this, the digital form of the video signal can be stored in a buffer and hence "freezing" of a whole line of TV signal is achieved. Several buffers may be used to store several TV lines which are then displayed simultaneously on the corresponding LED-lines. This has the effect of improving the duty cycle of the LED's, hence allowing a brighter output picture. Improvement of duty cycle can also be achieved by mounting a storage device (e.g. a

Figure 1. 4-Bit A/D Converter

capacitor) behind each LED. However, calculations showed that this latter approach is too expensive.

The A/D converter, which is shown in Figure 1, can operate up to 10 MHz. The outputs of the ADC are given by the equations below:

$$\text{MSB} = H$$

$$\text{THIRD BIT} = D\bar{H} + L$$

$$\text{SECOND BIT} = B\bar{D} + F\bar{H} + J\bar{L} + N$$

$$\text{LSB} = A\bar{B} + C\bar{D} + E\bar{F} + G\bar{H} + I\bar{J} + K\bar{L} + M\bar{N} + O$$

Sik Yuen

2.8 FROG (Project No. 36)

2.8.1 Project Outline

FROG is a device which interacts adaptively with an environment. The core of the adaptive logic is a structured collection of logic units which act as a filter, modifying the feed-forward paths between input (stimuli) and output. The filter is equivalent to the variable part of the world-model of the device.

All of the logic elements in FROG are drawn from a set of nine basic elements, defined hereafter, which operate in the signal continuum $[0, 1]$. These basic elements have been defined, functionally and mathematically, and used in computer simulations whose results have been very encouraging.

To explain the central principle of the proposed device, let us assume the context of a frog. Even more specifically consider only the feeding functions and four stimulus input ($S_1 - S_4$) derived from any sensory organs. At birth, the device will attempt to feed upon a relatively broad range of objects. Each feeding attempt is followed by internally-generated

signals measuring the tastiness of the food object. Upon encountering a very bad tasting object, O_1 with stimulus S_1^* , S_2^* , S_3^* , S_4^* , the world-model adjusts to strongly reject as food any further object having stimulus S_1^* and S_2^* and S_3^* and S_4^* , to reject less strongly any object having triple joint combination of the stimuli, to reject even less strongly any double joint combination, etc. If some other object, O_2 , has bad taste and shares some properties with O_1 (say the joint occurrence of $S_1^* \cdot S_3^*$), then its effect is to strengthen the rejection of $S_1^* \cdot S_3^*$ as a property associated with good food, thereby tending to generalize that this joint property implies "inedibility". Figure 1 illustrates the basic concept in a 3-dimensional stimulus space. The darkness of shading is proportional to the strength of rejection of stimulus $[S_1^*, S_2^*, S_3^*]$ due to a single occurrence associated with a negative result (e.g., a bad taste). The occurrence of a negative result associated with stimulus $[S_1^*, S_2^*, S_3^*]$, for example will add increased rejection to the pair $S_1^* \cdot S_3^*$ as well as to the S_1^* and S_3^* volumes. In other words, evidence that the stimulus $S_1^* \cdot S_3^*$ is associated generally with "bad taste" is being accumulated. The adaptive development of the filter (partial world model) described thus far is essentially a multivalued nonlinear classification technique with continuous learning. One novelty, however, is the fact that multiple joint occurrences are adapted more quickly than single occurrences with the result that the device tends to reject large fractions of the stimulus range as food only when the evidence has become quite strong. This strategy makes possible monotone adaptation,

Figure 1. Adaptive Filter: Relation to a Stimulus Space of 3 Dimensions

i.e., always increasing rejection.[†] Additionally, all processes are specified in well-defined, certainly realizable, logic-like elements in a regular well-organized structure. We stress all of these qualifications because without them one obtains systems which are either very difficult to analyze or unimplementable or both.

We propose the design and construction of the autonomous mini-robot, FROG, whose objective is to sustain its existence. FROG, tentatively a movable block, will operate on a horizontal surface having a variety of surface textures and populated by a mixture of cohabitants corresponding to potential food supply (both edible and inedible) and predators. The FROG (see Figure 2) will receive external data of two types, visual and tactile. Internally, FROG will generate hunger, fatigue, and pain signals.

The visual signals are used by FROG to locate the direction and tentative properties of cohabitants which may be potential food or predators.

Tactile data add a means of correlating the surface properties with occurrence of food or safe resting places or predators.

The central mechanism for adjusting to the idiosyncracies of its environment is the adaptive filter. A "situation analysis" is formed continuously by a mode of activity network which combines the environment sensory data, interpreted through the adaptive filter, and the somatic signals, which measure the requirements of FROG for food and rest, and decides a course of action. The three modes of activity are (1) feed; (2) rest; (3) flee.

2.8.2 Basic Elements of the Adaptive Filter

The basic elements which compose the adaptive filter are defined below. All inputs, x_i , and outputs, y , have values in the continuum $[0, 1]$ unless otherwise noted.

[†] In simulation, we also provide a time dependent decrease of rejection, i.e., "loss of memory".

Figure 2. Functional Sketch of Autonomous Mini-Robot (FROG)

<u>ELEMENT NAME</u>	<u>SYMBOL</u>	<u>NEAREST ANALOG</u>	<u>DEFINITION</u>
Complement (C)	$x \rightarrow \textcircled{C} \rightarrow y$	NOT	$y = 1 - x$
Minimum (m)	$\begin{matrix} x_1 \\ \vdots \\ x_k \end{matrix} \rightarrow m \rightarrow y$	AND	$y = \min_{i=1}^k (x_i)$
Maximum (M)	$\begin{matrix} x_1 \\ \vdots \\ x_k \end{matrix} \rightarrow \textcircled{M} \rightarrow y$	OR	$y = \max_{i=1}^k (x_i)$
Product (P)	$\begin{matrix} x_1 \\ x_2 \end{matrix} \rightarrow \textcircled{P} \rightarrow y$	--	$y = (x_1) \cdot (x_2)$
Equivalence (E)	$\begin{matrix} x_1 \\ x_2 \end{matrix} \rightarrow \textcircled{E} \rightarrow y$	Equiv. (fuzzy)	$y = x_1$ if $[x_1 - x_2] < \epsilon$ $y = 0$ otherwise.
Selective	$\begin{matrix} x_1 \\ x_2 \end{matrix} \rightarrow \textcircled{S} \rightarrow y$	Schmitt Trigger + AND	$y = x$, if $x_2 \geq 0.5$ $y = 0$, otherwise
Sense Indicator (SI)	$x_1 \rightarrow \textcircled{SI} \rightarrow y$	Schmitt	$y = 1$ if $x_1 > \epsilon$ $y = 0$, otherwise
Memory (MEM)	$x_1 \rightarrow \textcircled{MEM} \rightarrow y$	Fed-back OR	$y = 0$ when first powered "on". If x_1' is the first occurrence of $x_1 > 0$, $y = x_1'$ thereafter
Transmittance (T)	$x_1 \rightarrow \textcircled{T}_{(n)} \rightarrow y$	(?)	$y = 0$ initially. If x_1' is the first occurrence of $x_1 > 0$ at time t_1 , $y = \frac{1}{n}(x_1')$. If $i > n$, then $\frac{i}{n} \leftarrow 1$, $n = \text{posi-}$ tive integer.

Table 1.

The principal macroelement of the adaptive filter is a T* element (see Figure 3). A single T* element contains the relational memory between an elementary volume of stimulus space and result space. Each T* element is related to some mode of behavior by reason of its physical location in the filter.

If a stimulus arrives while $RESULT = 0$, then the T* element responds with an amplitude proportional to its correlation of that stimulus with the class of result. (This is a "READ" operation, essentially.) If a stimulus exists while $RESULT \neq 0$, then a cause-effect relation is assumed to exist; the T* responds by adjusting its stored correlation value in its internal transmittance (T) element.

2.8.3 Design Progress

At present work is being done on the design of the adaptive filter, the world model. Work has been done to chalk out the details of the entire scheme. Since the objectives of the project are still in the process of change nothing is finalized yet.

Debasish Bose

Figure 3. T* Element

Publications This Quarter Include:

Cheng, Richard. "Coefficient Generator and Cursor for the Stereomatrix 3-D Display System", Department of Computer Science Report No. UIUCDCS-R-71-484, used in partial fulfillment of the requirements for a Doctor of Philosophy degree.

Hadjistavros, Stavros. "COLORMATRIX. A Thermally Controlled Liquid Crystal Alphanumeric Display", Department of Computer Science Report No. UIUCDCS-R-71-486, used in partial fulfillment of the requirements for a Master of Science degree.

Kodimer, Dennis. "ALPHECON: Evaluation of a Developmental Video Storage Tube", Department of Computer Science Report No. UIUCDCS-R-71-485, used in partial fulfillment of the requirements for a Master of Science degree.

Partridge, Richard L. "PAGAN. A Three Dimensional Pattern Generator", Department of Computer Science Report No. 464, used in partial fulfillment of the requirements for a Master of Science degree.

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-1469-0196
Quarterly Report

2. TITLE

July, August, September

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)

Prof. W. J. Poppelbaum
Principal Investigator

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature

W. J. Poppelbaum

Date

September 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.

3. SOFTWARE SYSTEMS RESEARCH

A basic system for the graphical specification and numerical solution of differential equations now exists. Work this past quarter was directed towards providing a more flexible system that provides a more convenient interface for the user and works correctly with examples containing discontinuities. A complete solution to the latter problem has not been found, but an excellent start has been made.

Most of the originally planned work for the graphics end of the system has been completed for the PDP-8 system.

3.1 Numerical Processes

3.1.1 Ordinary Differential Equations (R. L. Brown)

The effects of four modifications to the original version of DIFSUB were investigated to determine if such modifications would significantly increase the rate of convergence without adversely affecting the accuracy or the amount of overhead involved in changes of step size or order. Rate of convergence was measured by the number of times the differentiating function was evaluated (number of calls to subroutine DIFFUN). Overhead calculations are primarily the number of inversions of the Jacobian matrix, PW, so the number of calls to MATINV was used as an indicator for that. The accuracy of the method was judged by comparing one of the variables $Y(1,I)$ to the proper value of the solution function at several values of T , chosen either for convenience or because that value is a critical point in the solution. Comparison was made by evaluating the known solution $F(Y,T)$ using double precision arithmetic and library functions, so the accuracy requested of the program ($EPS = 10^{-5}, 10^{-10}$) was always considerably less than the available accuracy of the solution ($\sim 10^{-15}$).

TESTS

Tests were taken from two sources: Appendix I of an as yet unpublished report by A. E. Sedgwick, University of Toronto, and "On Testing a Subroutine for the Numerical Integration of Ordinary Differential Equations," by F. T. Krogh, Jet Propulsion Lab Technical Memorandum #21F. Only tests with known closed solutions were chosen, and an attempt was made to use diverse problems affecting the numerical solution in different ways. The tests are:

1: $DY(1) = -Y_1 \quad Y_1(0) = 1$

$$\text{SOLUTION} = Y_1 = e^{-t}$$

2: $DY(1) = -Y_1 \quad Y_1(0) = 1$

$$\text{SOLUTION} = Y_1 = e^t$$

- 3: $DY(1) = Y_1 \cos(T) \quad Y_1(0) = 1$
 $SOLUTION = e^{\sin(T)}$
- 4: $DY(1) = Y_1/4 * (1 - Y_2/20) \quad Y_1(0) = 1$
 $SOLUTION = 20/(1 + 19 * e^{-t/4})$
- 5: $DY(1) = (Y_1 - T)/(Y_1 + T) \quad Y_1(0) = 4$
 $SOLUTION = R = (Y_1^2 + T^2)^{1/2} = 4e^{/2} * e^{-\text{THETA}}$
 $\text{THETA} = \arccos(T/R)$
- 6: $DY(1) = T(1 - Y_1) + (1 - T) e^{-T} \quad Y_1(0) = 1$
 $SOLUTION = e^{-T^2/2} - e^{-T} + 1$
- 7: $DY(1) = Y(2) \quad Y_1(0) = 0$
 $DY(2) = -Y_1 \quad Y_2(0) = 1$
 $SOLUTION = Y_1 = \sin T, Y_2 = \cos T$
- 8: $DY(1) = Y_2 Y_3 \quad Y_1(0) = 0$
 $DY(2) = -Y_1 Y_3 \quad Y_2(0) = 1$
 $DY(3) = 51 Y_1 Y_2 \quad Y_3(0) = 1$
 $SOLUTION = \text{Jacobi elliptic functions}$
 $Y_1 = 0, Y_2 = 1, Y_3 = 1 \text{ at } 0, 4K, 8K \dots (K = 1.86264 \dots)$
- 9: $R = (Y_1^2 + Y_3^2)^{1/2}$
 $DY(1) = Y(2) \quad Y_1(0) = .4$
 $DY(2) = -Y_2/R^3 \quad Y_2(0) = 0$
 $DY(3) = Y_4 \quad Y_3(0) = 0$
 $DY(4) = -Y_3/R^3 \quad Y_4(0) = 2$
 $SOLUTION = Y_1 = \cos(U) = 6, Y_3 = .8 \sin(U)$
 $U \text{ is the solution of } U - .6 \sin(U) = T$
 $U = T \text{ for } T = n, n = 0, 1, 2, \dots$

MODIFICATIONS

One modification made on DIFSUB was to extend the order of the Adams-Moulton procedure ($MF = 0$) to 12, which required additional $A(I)$ arrays, ($I = 1, N + 2$) to handle $N = 12$. Also, new values of PERTST were added. The scratch matrix SAVE was redimensioned to $17 \times N$.

Another modification involves suppressing the ability of DIFSUB to increase the step size on the step immediately following a decrease in step size. This might decrease the overhead involved in too frequent step size changes if such changes occur very often.

The number of steps the program will take after a change in step size or order before evaluating the need for another change, and the number of steps after that evaluation, if no step increase is possible, before re-evaluation, were modified in several ways. Both numbers were held at a constant value (initially at 5 and 10), both were allowed to vary as $P + 2$, P being the order of the method in use, and the first was kept at $P + 1$ while the second varied between 1 and 10.

Finally, the size of the factor that allows a change in step size was varied between 1.01 and 1.10 by steps of .01 to determine if any improvement could be made without sacrificing accuracy.

OBSERVATIONS

Of the first seven tests, only two ever made use of even the ninth order Adams method, and then only for one or two steps. Attempts to run more difficult tests have resulted in unexplained errors so no results are available for them.

The suppression of the ability to increase the step size immediately after decreasing it resulted in two more function evaluations in two tests, and identical results otherwise, possibly indicating that such occurrences are infrequent and when they happen are necessary.

It was found that the most efficient number of steps after a change in step size before evaluating further step changes was $P + 1$. The number of steps before the next re-evaluation, IDOVB, was varied between 1 and 10, with mixed results. See Figure 1. Anything through IDOVB = 5 gives very erratic results, although not much more can be said about IDOVB greater than 5. Since overhead supposedly decreases as IDOVB increases (this was not apparent for these tests), values of 8, 9, and 10 seem to be preferable, with 8 perhaps the best found in the study.

Finally, the size of the factor for allowing a change of step size shows erratic results for $1.01 \rightarrow 1.05$, more uniform performance for $1.05 - 1.08$, and uniform improvement over previous values for 1.08 to 1.10 . See Figure 2.

CONCLUSIONS

For stiff methods, setting IDOVB = 8 or 10 and the step size changing factor to 1.08 might result in a marginal improvement in rate of convergence without significant increase in overhead or loss of accuracy. Further study is needed on the effects of increasing the order of the Adams method to 12.

NFNS = NUMBER OF FUNCTION EVALUATIONS
 NW = NUMBER OF MATRIX INVERSIONS

TEST	IDOVb = 10		IDOVb = 8		IDOVb = 9	
	NFNS	NW	NFNS	NW	NFNS	NW
2	1832	6	1342	6	1836	6
3	755	13	745	15	713	7
5	254	10	256	10	248	6
6	483	17	443	18	455	17
7	571	6	697	6	721	6
8	1260	6	1312	6	1294	6
9	2179	7	2220	7	2248	13

Figure 1

TEST	FACTOR = 1.10		FACTOR = 1.08	
	NFNS	NW	NFNS	NW
2	1312	6	1312	6
3	945	14	945	14
5	254	10	257	10
6	483	17	483	17
7	597	6	597	6
8	1260	6	1305	6
9	2179	7	2179	7

Figure 2

3.1.2 Sparse Matrix Inversion (J. S. Deogun)

The generalized eigenvalue problem may be stated as follows:

Given $N \times N$ matrices A and B , find the non-trivial solutions of

$$(A - \lambda_i B) X_i = 0$$

where λ_i is an eigenvalue and X_i is a corresponding eigenvector.

Our concern is the special case where A and B are of sparse nature. As A and B are assumed to be singular and of rank lower than their order, Wilkinson's* method is considered useful in this case. In fact, Wilkinson's method reduces the effective order of the matrices. In a step or two the effective order of the matrices A and B reduces to rank of B so that any standard (non-sparse) routine can be used to find the eigenvalues.

The sparse eigenvalue problem is closely related to the sparse matrix inversion and was divided into three phases:

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

All three phases of various changes have been computed and the combination is being run on a small test example. Relationships between various routines are shown diagrammatically in Figure 1. Following is a brief discussion of various routines and their individual functions.

Subroutine SETUP has been changed to obtain a configuration of matrices A and B which is convenient for handling by pivot selection strategies and the SPARSE program. The configuration thus obtained allows the program to handle A and B separately as well as jointly, and as well as when required by the SPARSE program.

---

* For details of Wilkinson's method, see SIAM Journal on Numerical Analysis, 7, #4, December 1970.

,* These routines will be added to the package as soon as combination of others runs successfully and all the bugs have been removed.

Figure 1

Wilkinson's method, in one step, selects pivots from B and reduces B to the form shown in Figure 2, by Gauss elimination. Exactly the same operations are performed on A. In another step pivots are selected from A to reduce A to the form shown in Figure 3. Again similar operations are to be performed on B. The SPARSE program has been modified such that when similar operations are to be performed on both matrices, the

matrices A and B can be linked together in the proper way. Different pivot selection strategies are used for matrices A and B to achieve efficiency. Use of different pivot selection strategies are justified by the fact that different reduced forms are to be obtained of matrices A and B.

Figure 2

Figure 3

* R_B stands for Rank of B

As soon as the combination runs successfully on the small test example, it will be tested on big examples. This completes only the first part of the problem. The second part constitutes the developing of programs which set up matrix arrays from sparse representation of A and B and then computes eigenvalues of the reduced system by some standard procedure.

3.1.3 The Steady-State Package (B. van Melle)

The tests that have been made on the steady-state problem solver (MF=3) in DIFSUB all indicate that the best strategy for solving the steady-state problem actually has very little in common with the method used for the transient problem (MF=2). Hence a separate subroutine, DIFMF3, has been written to handle the case MF=3, while DIFSUB has been reduced to solving the single case MF=2.

The strategy used in DIFMF3 is essentially to start out with the predictor-corrector method of first order originally used in DIFSUB for this problem, but with a single corrector iteration. If no problems arise, the step size is adjusted toward one and the corrector step is phased out, resulting in simply the Newton method $y \leftarrow y - \left(\frac{\partial f}{\partial y}\right)^{-1} f(y)$,

which is quite fast in the vicinity of the solution. The rate at which the transition occurs is selected in an attempt to arrive at the solution quickly with a minimum number of function evaluations and matrix inversions. It is clearly necessary to re-evaluate the function at each step, but if $\frac{\partial f}{\partial y}$ does not change too fast we can avoid re-evaluating and inverting

the Jacobian. So at each step $D = \sum_{i=1}^n |DY(i)|$ is examined to see if we

are progressing towards the solution. If D has been reduced, we accept the step and continue the transition to Newton's method. However, it is not even necessary that D be reduced, and in some cases it is in fact not possible to reduce D , since we may have strayed in the wrong direction. So if D increases somewhat (the present program tolerates an increase of as much as 100-fold), we still accept the step. But if D suddenly blows up, we reduce the step size, re-evaluate the Jacobian, and repeat the step size, re-evaluate the Jacobian, and repeat the step, this time accepting it regardless. And even if no such catastrophe occurs, we periodically re-evaluate the Jacobian (presently every $5N$ steps, where N is number of components in vector Y), since this generally speeds convergence considerably, particularly when we are relatively near the solution.

The results of DIFMF3 tested on the various examples is shown in the following table, compared with the results from the old DIFSUB. In the simpler examples the greatest differences can be seen as a result of eliminating needless function evaluations and Jacobian inversions. On some examples the choice of initial values had a substantial effect on the performance of the old method, but the new strategy arrives at a solution with reasonable speed even where the old method had trouble or failed completely.

A new feature included in writing DIFMF3 is the ability to specify certain variables as given and others as unknowns. The present method MF=3 assumes the first derivatives y' are known and have value zero, while the variables y are unknown and are to be solved for. The new case MF=4 provides more flexibility in that the user supplies an indicator vector of the same length as y specifying for each component whether y or y' is the unknown. If y' is the unknown, the routine handles this simply by computing $\frac{\partial f}{\partial y'}$, instead of $\frac{\partial f}{\partial y}$ for the corresponding components of the Jacobian, and adjusting the value of y' instead of y in the predictor-corrector steps. The performance of this case MF=4 was tested and found to be much the same as that for MF=3. In fact, MF=3 is a special case of MF=4.

	EXAMPLE #	N	INITIAL VALUES	OLD DIFSUB			NEW DIFMF3		
				NS	NFNS	NW	NS	NFNS	NW
MIXED DIFFERENTIAL & ALGEBRAIC	1	2	1.0	134	686	122	44	58	6
			0.0	16	44	9	12	19	3
	2	2	1.0	17	49	10	15	22	3
			0.0	17	50	10	22	31	4
	3	2	1.0	26	70	9	41	54	6
			0.0	24	59	9	39	53	6
	4	3	1.0	19	65	11	9	16	2
			0.0	23	89	15	20	30	3
	5	2	1.0	18	50	10	13	20	3
			0.0	17	48	10	7	12	2
	6	9	1.0	170	1601	165	23	49	4
			0.0		FAILED*		129	213	13
	7	6	1.0	20	66	11	8	15	2
			-1.0	20	66	11	8	15	2

N - NUMBER OF EQUATIONS
 NS - NUMBER OF STEPS TO SOLUTION
 NFNS - NUMBER OF FUNCTION EVALUATIONS
 NW - NUMBER OF MATRIX INVERSIONS
 INITIAL VALUES ARE THOSE ASSIGNED TO THE VARIABLES y BEFORE FIRST STEP

* If the solution is not achieved in 200 steps we quit, but in this case the method just blew up entirely.

3.1.4 Plot Package (W. Chung)

The interactive transient analysis subroutine TRANAL (TRNØUT formerly) has been further refined and tested.

Calling sequences of P1, P2, and TRANAL are changed as follows:

1. EPS1 is deleted from the parameter lists since it is only used by P1
2. TEND is deleted from P1 and P2 lists.
3. Meanings of variables--N = # of total equations
M2 = # of nonlinear variables

Features which make the program more flexible and sophisticated in terms of interactive performance are:

1. In P2, the user can interactively change the initial conditions for y and yL to recall DIFMF3 when KFLAG < 0 for steady-state analysis.
2. The number of integrations (INTEG) is separated from the number of points of plotting on the screen (IPNT) for the purpose of providing flexibility in plotting and reducing the time spent in numerical integration to a minimum. Numerical integration is done by calling DIFSUB as many times as the number of integrations (or maybe plottings), which is rather expensive if it is repeated every time the user wants to change the graph.

Therefore, once a relatively large number of integrated points have been saved for a specified time limit TEND, the user can shorten or extend the time range of graph by specifying a smaller or larger IPNT ($\leq$ INTEG), respectively. In other words, the range of T on the display screen is controlled by a user such that $0 < T \leq TEND$ and

$$T = TEND * \frac{IPNT}{INTEG}$$
Of course, if the user needs more graph beyond TEND he will eventually have to change TEND and the integration is re-initiated.

3. Coordinates on the screen can be shifted up or down to display either positive or negative values of y according to the user's request. Thereby, useless portions of the graph can be eliminated from the display screen. This magnifies the remaining portion on the screen. This is easily implemented by assigning different values to the variable $IØRG$ which is the zero position of the T axis on the screen.

Moving the present coordinate is not very cumbersome, as it is done by recomputing the positions of points without re-integration.

4. After completion of a graph, the user is allowed to back up to different points in the program to change parameters and regraph. Moreover, a user can restart the procedure from scratch by backing up to the initial point of the program. In that case all the derivatives of y should be set to zeros.
5. To display or receive text lines and messages--MESSAGE/REPLY pair is used instead of FORTRAN READ and WRITE statements except when numerical data is included since the former is faster.

Flowchart of TRANAL:

3.1.5 Numerical Package (J. Frazao, B. van Melle, A. Whaley)

The numerical package was reworked this quarter. All routines besides DIFSUB were rewritten or are new. Each routine will be discussed separately. The major change was to separate portions of the matrix inversion which may be done less frequently than the inversion is now done. The techniques used identify variables and operations compiled to invert the matrix according to what they depend on. Some results of certain operations only change when the problem parameters are modified, requiring them to be evaluated only once before starting to solve a new problem. Operations of this type are now placed in subroutine MATIN1 (matrix inversion subroutine #1). Operations depending on only time and/or parameters need be re-evaluated at most once each time step, and are placed in subroutine MATIN2. Operations with results depending on other variables in the problem are placed in the remaining routine MATIN3. MATIN3 will be called when the single routine existing now (MATINV) is presently called. It is anticipated that a significant number of operations may be moved out of MATIN3.

3.1.5.1 Elimination (J. Frazao)

Changes are underway to modify elimination (WEED) to replace SETUP by doing SETUPs function symbolically. SETUP determines partial derivatives by numerical differencing, which is not as accurate or reliable, and is impossible when discontinuities are encountered.

3.1.5.2 SETUP (A. Whaley)

SETUP was rewritten to accomodate the separated subroutines outlined above. It also constructs a data structure for MATSET, telling how to take the partial derivatives numerically, where to store the results, and which variables may be differenced at the same time because no equation contains both (or all) of them. This section of the program

will be retained later, when SETUP is discarded in favor of the symbolic version in WEED.

Two new routines called INSTAL and REMOVE are now used to add and delete elements from the sparse matrix data structure, making SETUP and SPARSE much easier to understand.

3.1.5.3 SPARSE (A. Whaley)

SPARSE was rewritten due to the changes mentioned above, and in order to simplify and clean up many unwieldy features of the program. An attempt was made to improve the modularization of the program by removing functions of the program to subroutines that were very messy to handle in SPARSE. Some changes were mentioned in remarks about SETUP. The decision about whether to load an operand into a register because it was to be used frequently was resolved when it was discovered that there was only one way that would give correct results (not the way formerly used). The decision about whether to reload a register because the operand is already there was removed to subroutine COMPIL.

Other changes to SPARSE will result in its being able to handle more different kinds of problems. As SPARSE inverts a matrix, the matrix cannot be singular, which it almost always is. Ways were, therefore, investigated to handle singular matrices. It has been found to be common (and correct) to occasionally have an extra, redundant equation in the system of equations, leading to linearly dependent rows in the Jacobian coefficient matrix. The nonsquareness of the matrix is, therefore, ignored, and as long as the numerical integrator can solve the equations, only a nonfatal indicator message is given to the user. It is usually not possible to identify the useless equation, especially since it is generally possible to remove any of several. For this reason, the system does not attempt to identify extra or missing equations. An independent program by John Koch will be constructed to give the user a map of his equation structure so that he may easily identify the missing equation.

In the case of missing equations, after reducing the matrix to its most basic form it is discovered that the values of some variables must be specified (in a sense supplying the missing equations) before the integration may be continued. These variables are set to zero, and the user informed. Oddly, this is a normal occurrence, as in the following example:

It is specified that the difference in voltage $V(B) - V(A)$ is 1.5 volts, but what is the voltage at A? One of these variables (A or B) will be selected and set to zero. The user really is not interested in this process at all.

A reduced matrix appears thus:

I	α
0	

where I is an identity matrix, 0 is zero, and α is a reduced set of equations that cannot be solved. Each variable in α can have its value specified.

The following problem was considered:

In this problem either $V(A)$ or $V(B)$ should be specified (also $V(C)$ or $V(D)$). At first it was wondered whether the section α would contain one variable from the set of A and B , and one variable from C and D or an incorrect combination such as C and D with none of A and B . This speculation arose because of many other problems caused by not knowing the values of the elements in the matrix while blindly generating code to invert it. It appears (not proven, however) that α will always consist of a correct combination. Different pivot strategies will affect whether A or B is one of the unspecified voltages, but it does not appear possible for both of them to appear in α as unspecified.

Inconsistent equations are detected by the numerical integration package when it is discovered that a given equation cannot be satisfied. It is then known that this equation or one of the other equations, multiples of which were added to this one during the Gaussian elimination, must contain the error. Identifying this subset of equations when trouble occurs has not yet been implemented.

In general, the requirements that the Jacobian be square or non-singular have been suspended.

3.1.5.4 MATSET (A. Whaley)

This program has been modified so that it only computes partials for a given subroutine number (1 to 3 for `MATIN1`, `MATIN2`, and `MATIN3`). In addition, `MATSET` calculates the partials for several variables simultaneously in all equations. The group of variables involved do not appear in any equation together (see the description in `SETUP`). In other respects this program resembles its predecessors.

3.1.5.5 COMPIL (A. Whaley)

This program was virtually coded in absolute machine language, so no attempt was made to work with the old copy. A new version was written with several improvements besides handling multiple subroutines. Each subroutine starts off with 1024 bytes, a good size to avoid fragmentation with IBM's defective GETMAIN routine. The old COMPIL had a fixed area of 8K each for MATINV and MATMUL. The four subroutines now being compiled each start at 1K, but may be extended by 1K increments (separately) to any size.

3.2 Non-Numerical Packages

3.2.1 Item Analysis (J. Koch)

Item analysis worked correctly for all networks put through it this quarter. Work was begun near the end of the quarter on generating an extra equation and variable for each occurrence of the natural logarithm, and square root functions. These extra variables would then be used to avoid the points of singularity and complex results.* Two other routines were written, FSCAN and SPIEXT. Called from a FORTRAN program, FSCAN will scan a buffer for a blank, equal sign, or comma and return the previous characters in another buffer. Also, if the scanned characters were legal numbers, they were converted into a floating point number. The calling sequence is as follows:

```
CALL FSCAN(INBUF,INLEN,OUTBUF,OUTLEN,STACHR,ENDCHR,CONVER,  
          STPCHR,LENMOV,CODE)
```

where:

INBUF - the input buffer where the characters to be scanned are located.

INLEN- length (in bytes or characters) of the input buffer.

OUTBUF - the output buffer where the characters scanned from STACHR
to the delimiter are placed.

OUTLEN - length of the output buffer.

STACHR - beginning with 1, the character position of the first character
to be scanned. This parameter is updated to one character
past the delimiter upon return to the mainline.

ENDCHR - last character to be scanned.

---

* These functions have caused problems in the search for an initial steady state. Since "random" initial values are inserted to develop the sparse matrix structure and to start the iteration for initial values, the appearance of SQRT(exp) in another expression can lead to a negative value for exp. This is overcome by adding the variable Z, which is constrained to be initially positive, replacing SQRT(exp) by SQRT(Z) and adding the equation $Z - \exp = 0$ to the system. The Newton or other iteration will then restrict the changes to Z such that it never goes negative.

CONVER - word where the floating point number corresponding to the number scanned is placed upon return (see CODE).

STPCHR - the delimiter (blank, comma, or equal sign) will be placed in the first byte of this word upon return.

LENMOV - an integer count of the number of characters moved into OUTBUF.

CODE - '0' = the characters scanned were not numbers.

'1' = the characters were numbers and the floating point representation is in CONVER.

The routines CHRBIN, CHRFLT, FROUND, and TABLE1 are used to do the conversion from the EBCDIC to binary floating point.

The routine SPIEXT is used to patch up any floating point overflow, underflow, or division by zero detected by the 360 operating system. Without this routine, the equation analysis would be halted whenever one of these errors detected, and the numerical package would not be able to correct these errors by other means. In order that the operating system know where to go for overflow, underflow, and division by zero errors, a SPIE macro must have been previously executed with entry point 'SPIEHOLE'. When one of these errors occurs, the result register is replaced by X'3F8 00000 00000000' or D'0' (for underflow), an error count is incremented by 1, a search is made to pinpoint the beginning of the erring equation, and execution is allowed to continue. After an equation or set of equations has been run, a function call to SPIEXT is made (Note: the call must be 'SPIEXT($\emptyset$)'; $\emptyset$ is a dummy variable that is not used or modified. This is because FORTRAN expects at least one argument in each function call). The error count is returned in register 0 and the error count is zeroed out.

3.3 Graphical Remote Access Support System (GRASS)

Considerable effort this quarter has gone towards documentation and consolidation of the system. Further debugging also took place, with the result that the local package is now quite stable. Several important additions to the system were made and are documented below.

3.3.1 Disk Monitor System (R. Haskin)

Reasonably reliable production use of the DMS continued this quarter. The disk was returned to Data Disk for repairs on August 16, and has not come back up since it returned. In the interim, a DECTape system program called "DMS" was developed to allow the use of the disk monitor from Swaptapes. "DMS" consists of a new monitor head which maps disk I/O requests (SYSIO calls) to a Swaptape on unit 8. When run (by typing "DMS" instead of "SWAP" at the start of a programming session), the replacement monitor head is read-in in place of the standard one, and operation can proceed in a normal, if grossly slow, fashion.

3.3.2 Display Terminals (R. Haskin)

ACID continued to perform well this quarter. Continued hardware debugging plus some software changes contributed to a decrease in "garbaged" pictures. ACID modifications were

1. Changing all IOT wait loops to time out and go to an error routine if the associated device is not ready. This corrected problems with the terminal not returning coordinates after read-cursor commands, and with users inadvertently turning off the terminals or the PDP-8.
2. Modifying the display interrupt handler to be more intelligent about releasing buffers when done displaying segments, thereby improving memory utilization.

3.3.3 Information Retrieval (R. Haskin)

The package sent to Griffiss Air Force Base has been successfully integrated into their system.

The local version was in production use with GRASS all quarter. Two modifications are in progress, the first implementing user codes in order to allow some form of file separation and protection; the second being the investigation of the final implementation of a more sophisticated protection scheme.

3.3.4 Monitors (M. J. Michel, R. Haskin, J. Nickolls)

Several reports describing in detail the operation of the system were written this quarter, including:

SYSTEM OVERVIEW, Report 465

REMOTE FACILITIES GUIDE, Report 466

TERMINAL USER'S GUIDE, Report 467

SYSTEM SOFTWARE DESCRIPTION, Report 468

EXTENDED REMOTE FACILITIES GUIDE, File 867

GRASP (360 Remote Monitor)

Production use has been reliable all quarter, no changes were made or are anticipated.

2701 Data Link

The interface between the PDP-8 and the 2701 PDA has been reliable all quarter.

GLASP (PDP-8 Local Monitor)

Production use has continued reliably all quarter.

Program Segments

CALLER: The "purgeout" feature has been changed from a menu pick function to a command which must be typed in on the keyboard. It was felt that this was necessary to prevent users from accidentally doing a purgeout by inadvertently hitting the wrong screen area.

GNDRW: Several important functions have been implemented this quarter. All move and delete functions are currently implemented, allowing easy modification and correction of an existing picture. The "display all terminal connections" function has been added to allow the user to see the invisible as well as the visible terminal connections.

A "picture recompile" feature has been added. This function, invoked from the keyboard via a R<mnemonic name>, "recompiles" the picture to include the latest instance definition of the specified mnemonic. The instance and terminal blocks are modified to reflect any changes which may have been made to the instance definition since the picture was constructed; and error checking is done to insure that the new definition is at least topologically equivalent to the old one. The feature eliminates the need for deletion, respecification and reconnection of each instance of a mnemonic when its definition is changed, making it a great pain and work saver.

Scheduled for implementation in the next quarter are all currently unimplemented functions plus a feature allowing horizontal and vertical constraints to be applied to all drawings to enable construction of neater looking pictures.

New Program

LLSD: Local Library Service Discographer

A new program segment was written this quarter to facilitate maintenance of the local PDP-8 library.

LLSD is selected from the menu in CALLER.

The user's menu is displayed and may be scrolled up and down by joysticking the PREVIOUS and NEXT light buttons as in GNDRW.

All commands to LLSD are entered via the keyboard. Only the first two characters of the command are examined, and a space separates the command from the first operand. Two commands are currently implemented: COPY and REACT.

COPY copies the first file to the second file. A space separates the two filenames. A file may be a mnemonic instance or a picture. Filenames must contain a picture or mnemonic name, optionally followed by a user ID. If no user ID is specified, the user's logon ID is used. Filename conventions are identical to those used in LSD (see DCS Report No. 466), with the omission of the library name.

Some examples follow:

```
COPY_PICTUR/0072POOHBEAR_MYPIC  
COPY_DIODE.1_DIODE.1/9999CSJOCK  
COPY_NEURON.6_NEURON.4  
COPY_NET5/0072NICKOLLS_NETWORK/0072HASKIN
```

REACT has no operands; it transfers control to the program segment REACT for access to the remote 360 library via LSD. Joysticking the RETURN box in REACT will return the user to LLSD. This feature allows interaction between LSD and LLSD, facilitating communication between the remote library and the local library.

Several additional functions for LLSD are being planned: a PROTECT command for specifying protection for any file, a CATALOG command for displaying a user's menu, and a DISPLAY command for displaying a user's file.

GLASP Status

The monitor and all program segments appear at this time to be quite stable. With the publication of the five reports mentioned earlier, it is also felt that the system is well enough documented to allow new users to familiarize themselves with the system and use it with a minimum of trouble. Software documentation is also in a state where it would not be difficult for a competent programmer to use, add to, or modify the existing system software.

Applications Support

The extensive use of the package for the development of the Simulation and Modeling System has continued this quarter. Many networks have been specified and sent to the remote system for analysis. Work is currently proceeding towards implementation of graphical communication in the opposite direction.

A generalized graphical output program is currently being written to provide visual output from the integration routine through the motions of objects on the screen.

A user at a terminal may visually observe the progress of the integration phase in the solution of his problem. The user first creates pictures to be moved later across the terminal screen. He then associates the coordinates (size and rotation optional) of each picture with a pair of variables being integrated in the problem; one variable for X and one for Y. A new picture is sent to the terminal at each step in the integration, causing the pictures to move as the values change. A "trailer" may be attached to a moving picture to record its path, or

successive pictures can be displayed over each other without erasing in between. For example, if the picture is the letter "O", a "trailer" would produce a conventional graph with circles at the data points and straight lines connecting the points.

More elaborate pictures could display movement of bodies in potential fields, traffic density and flow in a model intersection, or deflection in a structure due to stress and strain.

A program using graphical input and output for investigating methods for the numerical integration of stiff equations has been developed using the remote data structure utilities (COMMUNE, etc.).

Illustrating the generalized capabilities of the GRASS system, work has begun on the implementation of a graphical matrices language based on the OL-2 language and using the terminals for graphical input and output.

3.4 Computer Maintenance and Construction

3.4.1 Graphics-8 Hardware (C. E. Carter)

The total Graphics-8 hardware has been moved next door to room 31J and up on a false floor. The operation, maintenance, and demonstration of the equipment are much better than before. Also, there should be some improvement in reliability of the system.

All the time the hardware was powered down for the move, the Data Disk was shipped back for examination and overhaul at the California plant. This disk has been in use for some time during the developmental phase of the project and will have to be applied during the time that a search for a replacement is made.

New Disk Acquisition

While this was proceeding, a bid specification was prepared and sent to possible suppliers. The "hard" part of the specification and the analysis sheet is contained here.

Specification of Rotating Magnetic Storage Device

"The device in question is a rotating magnetic storage device (disk or drum) along with its controller. Its interface requirements are those established by Digital Equipment Corporation in the (DM01) multiplexor on the PDP-8 processor. The performance specifications are:

1. Capacity (Min. useful data) 256K (12 bit) words,
2. Block Size Variable from 1 word to 4096 words,
3. Average Latency (No greater than) ~ 17 milliseconds,
4. Transfer time/word (No greater than) ~ 19 milliseconds,
5. Reliability--No more than one nonrecoverable error in 10^8 bit transfers.
6. Interface requirements--PDP-8 levels of 0V. to -3V. or TTL levels of 0V. to +3V. (Preferably not mixed).

In addition to the above, detection and correction of recoverable and detection of nonrecoverable errors shall be provided. All of the testing and checking required to establish reliability or to carry out diagnostics is assumed to be a part of the disk, its controller or the software supplied with the disk or its controller."

BID ANALYSIS (K00C017) October 1, 1971

COMPANY	PRICE \$	DELIVERY	WARRANTY	TRAINING	EXPANSION	COMPUTER SOFTWARE	AV. ACC	WORD XFER	SECTORS/ TRACK	WORDS/ SECTOR
1. Jackson Assoc. Xerox Disc	16K									
2. DEC	15.58K									
3. Digital Develop. Corp. Single Disc Single Cont. Single (Disc- Exp.) Cont. Single Disk Dual Cont.	 10.635K 10.935K 11.935K	90 days	1 year 90 days Cost Free	 \$900	 Double Heads Additional Drives	 Not Mentioned	 8.8ms	 235KHZ	256	16
4. System Industries AMC Disc	12.22K	60 days	1 year all labor and parts	No Charge	Can double heads--can add 3 addi- tional drives	Can use DEC software	8.7ms	255KHZ	16 or 32	256 or 12
5. Data Disc	11.2K	60 days	2 years 90 days Cost Free	\$1200		Patches Provided	16.7 ms	125KHZ	64	128

The work resulted in our requesting a disk and controller from System Industries.

3.4.2 Equipment Maintenance Log Summary (H. Lopeman, B. Miller)

PDP-8

1. SW register noise on PDP-8 console. Wiggling switches will cause the processor to halt. (No solution as yet)

Inktronic

1. Paper feed clutch frozen. (Lubricated and cleaned)
2. Blown high voltage power supply. (Replaced supply)

PDP-8/I

1. High speed reader motor bad. (Replacement motor ordered)

Comutek

1. Terminal #2, bit 2 on board "B" intermittent. (Cleaned and repaired board, OK)
2. Terminal multiplexor still giving spurious problems.

PDP-7

1. Several problems existed in the PDP-7 during this period.
  - a. The instruction decoder area of the 2701 interface contained two bad nand gate boards.
  - b. The interrupt area of the PDP-7 CPU contained a bad pulse amplifier.

- c. The program counter of the PDP-7 CPU had a bad flip flop board, and a carry chain pulse amp.
- d. The paper tape punch was cleaned, lubricated, and adjusted.

Boards replaced in the PDP-7 were

1I26-B204

1D23-B201

1D29-B201

1E24-B620

All filters have been changed and cleaned in the PDP-7, 630, PDP-8, and 338 during the month of August 1971

3.4.3 Stereomatrix Interface (I. Cunningham)

The interface logic for the stereomatrix display and the PDP-8/I has been checked out and is now working. The display was simulated by a square wave generator in order to check all functions. A pushbutton box has been constructed. The logic control for it at the display end is now being wired.

A software system has been organized and memory allocators, display and clock interrupt controls, basic file management, and straight line generator routines have been written. This basic system will permit the drawing of simple 3D objects. The organization and coding of these routines will hopefully aid in identifying system pathologies.

All hardware work is analyzed until the stereomatrix display comes up. Checking and expansion of the software will continue in the next quarter.

Reports Printed

- Haskin, R., Nickolls, J. and Michel, M. GRASS: Remote Facilities Guide, Department of Computer Science Report No. 466, University of Illinois, Urbana-Champaign, July 1971.
- Michel, M. GRASS: System Overview, Department of Computer Science Report No. 465, University of Illinois, Urbana-Champaign, July 1971.
- Michel, M. and Koch, J. GRASS: Terminal User's Guide, Department of Computer Science Report No. 467, University of Illinois, Urbana-Champaign, August 1971.
- Michel, M. GRASS: System Software Description, Department of Computer Science Report No. 468, University of Illinois, Urbana-Champaign, August 1971.

File Numbers Printed

- Haskin, R. GRAPHICS 8: System Maintenance (Software), Department of Computer Science File No. 865, University of Illinois, Urbana-Champaign, July 1971.
- Michel, M. and Haskin, R. GRASS: Extended Remote Facilities Guide, Department of Computer Science File No. 867, University of Illinois, Urbana-Champaign, August 1971.
- Nickolls, J. and Michel, M. GRAPHICS 8: System Maintenance (Hardware), Department of Computer Science File No. 866, University of Illinois, Urbana-Champaign, August 1971.

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

(See Instructions on Reverse Side)

1. AEC REPORT NO.

C00-1469-0196

2. TITLE

3rd Quarterly Progress Report 1971 (July, Aug., Sept.)

3. TYPE OF DOCUMENT (Check one):

- ☒ a. Scientific and technical report
☐ b. Conference paper not to be published in a journal:

Title of conference _____

Date of conference _____

Exact location of conference _____

Sponsoring organization _____

- ☐ c. Other (Specify) _____

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

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

5. REASON FOR RECOMMENDED RESTRICTIONS:

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

C. W. Gear, Professor
and Principal Investigator

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature

Charles W. Gear

Date

October 27, 1971

FOR AEC USE ONLY

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

8. PATENT CLEARANCE:

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

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

A balanced program of experiments with an image processing computer is being carried out. Developments of this past quarter include:

1. Atlas, an extension of Show-and-Tell, our package for interactive picture processing, is being designed and implemented to provide the capability of correlating graphically-defined Atlas information with corresponding pictorial information. (See Section 4.1.2)

2. The flying spot microscope became operational during this quarter. Full X, Y, and Z axis motion is possible and under PDP-8e program control. (See Section 4.1.4)

3. Development of the theoretical basis (signal detection theory, covering techniques, etc.) for plane parallel image processing procedures critical for rapid preprocessing and local feature extraction is being developed. A report entitled "Texture Analysis" by S. N. Jayaramamurthy is in final editing. (See Section 4.2.1).

4. Cervical smears: Using the PAX II package running under Show-and-Tell on the IBM 360/75 and PDP8/e, programs have been written which can perform attention-centering and some feature extraction on single, selected microscope fields where the degree of cell overlapping, clumping and degeneration is not too high. (See Section 4.4.1)

5. Transfer Memory II: The design phase is now largely complete and construction bids are being prepared. (See Section 4.5.3)

6. Taxicrinic Processors: Work over the past quarter involved the publication of Volumes Two and Three (DCS Reports #475, 476) of the Taxicrinic Processor (TP) design. The design of this basic machine is essentially complete.

4.1 INTERACTIVE PICTURE PROCESSING

4.1.1 Show-and-Tell

During this quarter, Show-and-Tell (S & T) was used extensively as a research tool, primarily in the development of programs for analysis of cervical smears (see Section 4.4.2). At present all image processing is done on the IBM 360/75 of the Computing Services Office, with the local S & T software performing image acquisition and display and picture transmission back and forth between the 360 and the PDP8/e.

Application development has now progressed to the point where it is necessary to be able to process large quantities of pictorial data. This is not feasible to do on the 360/75 for economic, technical and administrative reasons. Also, the hardware Pattern Articulation Unit is nearing the checkout stage. Because of these considerations, we are implementing the local image processing functions of S & T, as defined in the S & T System Specification (DCS Report #429).

This quarter, some essential subroutines for a software PAU simulator were completed. The simulator will be used for PAU hardware checkout and debugging, as well as for image processing. As the hardware PAU becomes available, the simulator modules can be phased out.

4.1.2 Atlas Extension of Show-and-Tell

Atlas, an extension of Show-and-Tell, is being designed and implemented to provide the capability of correlating graphically-defined Atlas information with corresponding pictorial information.

Areas of application include the following:

1. Brain mapping at a gross anatomical level.
2. Correlating satellite multispectral data with topographical

and geophysical maps.

3. Fault diagnosis of LSI circuitry, given a scanning electron microscope as a digitally-directed probe and layouts/diagrams of the LSI circuitry.
4. Interpretation of microscopic pathology in histological sections, given an atlas of descriptive histology.
5. Correlating anatomical information with X-rays, e.g. from textbook illustrations of the spine interpret an X-ray of the same.
6. Correlating remotely-sensed thermal and radiation pollution of urban areas with area maps.

Ideal Capabilities of the System

1. The Atlas material - considered here as largely comprising a set of line drawings - would be read into the machine through any SMV input port: TV camera or flying spot scanner (using a film intermediary). This graphical material is to be digitally encoded and stored.

2. The image material also would enter the system through any SMV scanner or TV camera. This material normally would not be digitized or only selectively digitized in a manner described below. Immediate requirement is that the graphically (i.e. Atlas-) derived information can overlay the optical image and be used by the operator to interrogate the incoming scanner image.

3. In general the optical image will exhibit distortions (of stereo-projection, normal biological variation, etc.) and will also include areas which will match incorrectly or incompletely the Atlas information. Normally, it will be necessary to correct or distort the Atlas-derived graphical image to enhance its match to the optical image. This capability

requires two largely separable facilities:

- i) The ability, using the available Atlas information, to detect the actual boundaries in the picture and therefore parameterize the required deformations and corrections of the Atlas-derived information.
- ii) The ability to generate the correct deformation or corrections of the graphical image.

A rather simple case of the above is the facility to scale the Atlas materials to match rectified images of the corresponding areas.

4. The Atlas information, though entered into the computer possibly as a set of serial cross-section graphs, would be handled internally, when applicable, as a coherent three-dimensional model. For example, in brain mapping we wish to speak of the boundary of a given nucleus. Also cross sections (sagittal, coronary, etc.) perpendicular to more than one axis may be required.

5. There is also the necessity of realizing that the image data and possibly the Atlas data may be viewed at more than one resolution. Specifically, with a slide under the microscope, we would like to zoom up and down in resolution by changing the microscope objective and to systemize this additional information in an augmented Atlas. Changes of magnification will be needed, e.g. in brain mapping to identify nuclei boundaries.

6. The graphical information will be coordinated by a hierarchical structure of directed labeled graphs. For manipulating these latter, it is hypothesized that a variant of the Structure Operation Language may be applied. Pointing to a region on the CRT can be considered as generating a pointer to the region node in the appropriate digraph structure.

7. The Atlas information, particularly the digraph structure, should also play a key role in later extensions of the system to provide information storage and retrieval. Specifically, pharmacological reports (journals, articles, etc.) noting drug responses can be indexed by the anatomical site of excitation/detection. Analogous techniques can be devised for other Atlas-described materials.

8. Journal illustrations, particularly in electrophysiology but also in agriculture and geophysics, can often be considered as providing augmented local Atlas information. Accordingly, the graph structure representing the Atlas should be considered to be growing and constantly modified as new scientific reports appear.

9. Large-scale files (see McCormick and Richardson, "Design Concepts for an Information Resource Center with Option of an Attached Automated Laboratory", DCL Report #203) are now becoming commercially available. In particular, Foto-Mem, Inc. is now building for the New York Times a retrieval system where the information retrieved is stored in microimage form. In general, the card handling capabilities built into the trillion-bit photo memories using laser-generated bit patterns can be and are being commercially adapted to provide large-scale microimage storage. It is anticipated the Atlas system could evolve to provide a powerful storage and retrieval mechanism for such image-based systems.

Implementation

1. The system cannot be implemented at a single pass since extensive experimentation is needed at each step. Accordingly, thus far the implementation has proceeded as a series of small experiments. The experiments have been ordered so as to provide us enhanced image processing

capabilities at each stage.

2. Implementation studies decompose into two largely separable activities:

- i) Initial encoding, analysis and representation of Atlas information. Here the existing Show-and-Tell System is being used to digitize Atlas "cartoons" and associated image data. The graphical overlay procedure makes extensive use of the incremental mode of display at the monitor.
- ii) Analysis and implementation of the Graph Structure Language (see Section 4.3.2) and its associated file system.

3. The use of the Atlas-derived information to control image processing is being left largely to individual investigators (e.g. brain mapping. See Section 4.4.2). Many general tools are becoming available here. These tools include: Interval Covering Techniques, Binary Factorization Procedures, Clustering Techniques, and the formal thesis work of S. N. Jayaramamurthy and J. C. Schwebel.

Progress on Atlas Transformations

Since Show-and-Tell is currently embedded in FORTRAN, we have written two sets of FORTRAN subroutines, SHAFT and SHAFTRANS.

1. SHAFT allows us to create, interactively, a data structure which represents a planar labelled cartoon, a cartoon being defined as a collection of labelled polygonal regions. As in Figure 1, the cartoon SECl consists of three regions A, B, and C. Each region consists of a list of pointers to a table of currently-used coordinates which define the vertices of that region.

Regions may be created either under program control (e.g. as output from a boundary-detection routine) or by the user using Show-and-Tell.

Additional hardware to enable the latter method (function switches and a hardwheel-driven cursor) is currently being constructed.

ii) SHAFTRANS allows affine transformations of a three-dimensional object defined by a set of lines. For example, a cube may be defined by four sets of lines as in Figure 2. The object thus defined may then be transformed and displayed on a plotter. Transformations currently supported are translation, rotation, scaling, and projection.

Figures 3 and 4 show a windowed view of a 50 x 50 grid, and two copies of that grid rotated 45° .

4.1.3 Terminal Communications

During the last quarter specifications were drawn up for a programmed I/O controller for LINC Tapes. The design of the controller is about 50% complete. The tape marking program and a program comparable to the DEC routine RWTAPE have been coded to make sure the control is usable.

The IBM 2701 interface has, at random times, missed the select and completion flags. The problem was found and will be corrected by adding two new IOT's: CLEAR DONE FLAGS and CLEAR SELECT FLAGS.

Due to budgetary constraints no work was performed on the multiple teletype network.

4.1.4 Scan/Display Devices

The flying spot microscope became operational during this quarter. Full control of X, Y, and Z axis motion is possible and under PDP-8e program control. X and Y steps move the slide by $34.4\mu\text{m}$ increments, while the Z-axis drive (focus) moves the slide by $0.12\mu\text{m}$ per step. The stage motors may be driven at up to 200 pps bidirectionally. Safety limit switches are not yet installed but will be. Users stand warned.

FIGURE 1.

FIGURE 2.

FIGURE 3.

FIGURE 4.

A five lens turret will take any standard microscope objective (three powers are currently provided). A scheme was devised to add a reference photomultiplier before the spot image passed through the objective by using one of the optical charger parts. This was expedited, and an improved image was obtained since product detection was then possible. Resolution is very good but as yet unmeasured.

The CRT currently in use in this device is a Litton L-4238 with P-16 phospor. This tube is being operated at an excessively high beam current to overcome the light loss in the microscope optical system due to a small entrance pupil and to absorption of the CRT's blue UV light output by the large number of glass elements. It is anticipated that this situation can be improved by replacing the P-16 equipped CRT with a tube using one of the new broad-spectrum phospor; this will have the additional benefit of enabling experimentation with color data.

It was agreed that the abominable technique of reprinting 35mm film into 46mm strips could be eliminated by converting the spare 46mm transport into a sprocketed 35mm unit at minimal cost and effort. Plans for bringing up such a unit are in the wind. However, the new controller must be brought up to run the unit as this becomes the third operational scanner.

The second scanner, for use with video digitization techniques, is currently under construction. All outside contracted work for wiring of PCBs and back panel wiring have been completed. In-house additional back-plane wiring is in progress. Intra-drawer back-plane wiring is about 5% complete. Inter-drawer wiring is about 30% complete.

Four additional sections of the SMV manual have been completed. No additional work was done on the CIU or I/O sections.

4.2 PARALLEL PROCESSING

4.2.1 Extensions of Signal Detection Theory

The development of ROC technique for the analysis of texture has been described in earlier reports. This technique has been successfully used to extract textural regions, as described earlier. This can also be extended to extract textural borders in a combined scene, given the protosamples of the textures present in the scene.

The events extracted at the boundary where two textural regions meet in general are expected to be different from those that are extracted from the core of either region. This is so because these events are made up from the points from two different regions. We can say that the 'border texture' is different from either of the textures it is separating. Sometimes this may not be the case - the template may be too small to pick up these differences.

Let us consider the combined scene as T^1 and the union of all protosamples as T^0 . We perform ROC analysis (Jayaramamurthy, S.N., "Application of Signal Detection Theory for the Analysis of Texture", not yet published) and obtain sets F^1 , F^ϕ and F^0 . Obviously F^0 is the null set because all the events in T^0 happen to occur in T^1 too. But the set F^1 contains the events that occur exclusively in T^1 , which are nothing but the events extracted at the borders. We can go back and 'color' the combined scene for F^1 alone and locate the border.

This technique has been applied to extract deformations in a textural scene, borders between natural textures, etc. A detailed account of this and other applications can be found in the reference cited above.

4.2.2 Interval Covering

Work on interval covers was resumed in mid-September. Experiments

were done with the operational program used to determine interval covers which distinguish elements of a set F^1 from those of a set F^0 according to the technique presented in Michalski, DCS Report #450. Work was started on the problem of generating interval stars from explicit information about F^1 in the case where F^0 is too large to be worked with conveniently.

4.2.3 PAX Language Support

A second number of the PAX Users Group Newsletter was distributed this quarter to the thirty individuals returning a survey form included in the first issue. This survey revealed that as of July 1971, PAX II (or a variant) is operational at nine locations on four different computer types: CDC 3600, IBM 360, UNIVAC 1108 and PDP-10.

A name, "Plane Talk", for the PAX User's Group Newsletter has been arrived at by a somewhat democratic process.

Only a small number of minor corrections and modifications were made to the PAX II package this quarter.

4.3 STRUCTURAL INFERENCE

4.3.1 Scene Segmentation by Clustering

A careful review of published clustering techniques gave rise to the following observations:

1. A cluster is usually considered as a set of patterns in a multidimensional attribute space such that its density is large in comparison with neighboring sets of patterns. To measure the density of patterns in an attribute space, it is necessary to define at least a pseudo-metric in this space. No method to define a distance function between different attributes has been developed which is acceptable for scene analysis.

2. The clustering techniques available do not describe efficient algorithms to find clusters for the purpose of scene segmentation.

To overcome these difficulties, we are trying to develop a general model that allows us to define scene segmentation by clustering as an optimization problem. To find an efficient clustering method, we are considering the following approaches:

1. Global approach: The overall density distribution of picture points in an attribute space used to find cluster centers in a scene.

2. Local approach: The cluster centers found in the global approach serve as centers for locally formed clusters using the nearest neighbor classification method.

4.3.2 SOL Language Support

This quarter the work centered around checking syntax of this graph-oriented language and preparing it for implementation. We have

specified the root operations (mentioned in the previous Quarterly Progress Report) that are sufficient for expressing network transformations. They were prepared with the goal of embedding them in PL/1.

We have generated an algol program which differentiates SOL statements from PL/1 and checks for legitimacy of their syntax. At present we are attempting to implement the language with a two-pass compiler, where the first pass will replace SOL statements with their PL/1 equivalents and the second pass will actually be a PL/1 compilation.

4.4 APPLICATIONS

4.4.1 Cervical Smears

Development of this application has been completed to the limit of the current hardware/software tools. Using the PAX II package running under Show-and-Tell on the IBM 360/75 and PDP8/e, we have written programs which can perform attention-centering and some feature extraction on single, selected microscope fields where the degree of cell overlapping, clumping and degeneration is not too high.

These preliminary results were presented at two conferences, the Engineering Foundation's Research Conference on Automatic Cytology at Henniker, New Hampshire (July 26-28, 1971) and the Conference on Two-Dimensional Digital Signal Processing at Columbia, Missouri (October 6-8, 1971).

Work on refining and evaluating the programs will continue using the current facilities while the local image-processing capability is completed (See Section 4.1.1).

4.4.2 Brain Mapping

Clustering methods that are being developed for scene segmentation (cf. Section 4.3.1) will be applied to isolate nuclei of individual brain cells from background tissue as well as the delineation of boundaries in the gross anatomy of the brain.

Simultaneously we are developing optical filters for recognizing brain cells using the interval covering technique (cf. Quarterly Technical Progress Report, April-June 1971, Section 4.2.3.3).

A summary of our results will be issued later this year.

4.4.3 Cytospectrometer

During this quarter a number of schemes for particle (droplet) generation were investigated. Principally for the purpose of generating finer streams, and incidentally for avoiding clogging problems, a technique of forming a very fine-tipped cone of liquid (Taylor cone) on the order of a few tens of molecules radius by means of a high electrostatic field was tried. A number of runs with whole blood, fractions and bacteria were made onto glass slides with a crude device. Results were promising enough - single cell isolation in some cases - to justify the design of a better device. Construction of a six inch diameter modular vertical instrument is underway. Initial results with even the unstable jet were sufficiently interesting to suggest at least the temporary abandonment of the vibration type (Rayleigh jet) particle generator. Detection, stabilization and control schemes are proceeding apace.

4.4.4 Remote Manipulation

We have defined a class of objects as two-dimensional "angularly simple" polygons. Each such polygon has a convex set called its feasible set (Maruyama: unpublished, May 1971). To encode such a polygon into a unique radial pattern sequence, $S_r(x_0)$, we have developed computer procedures which first define a unique point x_0 in a given angularly simple polygon.

Our current interest in this class of polygons stems from problems of geometrical shape perception. From a psychological viewpoint (e.g. discrimination of an odd shape among many, detection of similarity or dissimilarity between pairs of shapes, etc.) angularly simple polygons form a particularly attractive collection of shapes. The purpose of this

study is to develop a computer system that gives solutions similar to those given by human subjects, and therefore provides a mechanism for naming objects in a scene in accord with given prototypes.

4.5 COMPUTER SYSTEMS

4.5.1 IBAL Assembler

The final version of syntax specification and language definition for IBAL (Illiac III Basic Assembler Language) has been completed and is reported in Report No. 469. IBAL is the basic language which will be used for all sophisticated system programs planned for the Illiac III system.

Bottom-up compilers are very attractive due to the fact that they provide a straight forward method for error recoveries. Considering Compiler Compiler systems we found TWS (Translator Writing System - Illiac IV) compatible with our requirements. This system generates a compiler by producing Floyd productions for syntax and linking the passer to different routines for semantic actions. Attempts to produce the first version of the compiler through TWS have been unsuccessful because after the generation of Floyd productions we found bugs in the later phase of the system, and unfortunately inadequate information was available to debug and retry the TWS.

We then tried to generate a top-down compiler by using the TWST system. We produced a syntax analyzer which is a long algol program. After making sure that the syntax was free of bugs, we are now using the 360 version of TWST called TACCS (A Table Driven Compiler Compiler System) to build a compiler for IBAL.

In the first pass of the translator we deal mainly with syntax and data declarations and produce an intermediate code to be processed by the second pass. The structural tables and codes from the first pass are the input to the second pass, which generates the code for the actual memory allocation and pointer register allocation, and translates the

intermediate code to actual machine language codes for Illiac III.

4.5.2 Operating Systems

Illiac III's operating system is described in the Illiac III Reference Manual, Volume IV: Supervisor Organization, which was printed during this past quarter on August 12, 1971.

The following concepts evolved during the past quarter:

1. In our multiprocessing environment, it is possible to have as many as four taxicrinic processors working on a given task. As was mentioned in a previous quarterly report, an INCK instruction prevented one interrupt from destroying the information peculiar to another interrupt. We have established that there will be a circular buffer of four pointers to available space to take care of each successively occurring interrupt. One extra space in this circular buffer will point to a first-in-first-out queue that stores these pointers. The queue is necessary when there are even more than four interrupts at one level, in which case the pointers will be overwritten and therefore lost if not stored in the queue.

2. In case further general interrupt information must be stored, provision has been made to allow storage in successive locations following the circular buffer.

3. All segments will be in a hierarchical structure. The structure is formed by using the accessing rights information stored in the Directory and in the Global Segment Table.

4.5.3 TM II

The following progress on the second generation Transfer Memory, TM II, was noted during the last quarter:

1. Design of PRE complete.
2. Design of discrete interface circuits between TM II and the IA complete.
3. Strategy for memory card and in particular chip selection complete.
4. Power supply (5v., 100A) ordered for evaluation.

The following tasks are expected to be completed during the next quarter:

1. Letting of bids for drawers and wiring for PRE.
2. Layout and fabrication of discrete interface cards.
3. Specifications for cabinet, cooling equipment, and necessary cabling.
4. Finalization of memory and card design.

5.4.4 TP, AU, and IOP

Work over the past quarter involved the publication of Volumes Two and Three (DCS Reports #475, 476) of the Taxicrinic Processor (TP) design. The former describes the basic machine control sequences and flow charts and the latter describes TP instructions. The design of the basic machine should be completed in the next quarter. Instruction sequence control logic design has been initiated. Completed instructions include those concerned with the operand stack, unary logical operations and list processing features of TP. A total of 183 TP control logic drawings have been completed. Of these, 77 have been mapped into completed cards and 51 are in the layout process. Thirty-one of the completed cards have final back panel wiring.

Documentation for the Volume Two of the Arithmetic Units (AU) describing the control sequences and their logical design was started

and is continuing. It should be available in the coming quarter. A final index for the AU processing hardware and control logic drawings was prepared. 435 of 470 logical cards for processing hardware are in the machine after being checked out. Logic layout was completed for all of the 66 control logic cards and 15 more were wire listed and will soon go for wiring purposes.

Printed circuit board (discrete components) testing has been completed on 206 of the 250 boards required for the Input/Output Processor (IOP) processing hardware.

4.6 DOCUMENTATION

4.6.1 External Documents Issued

Report No. 469	Schwebel, J.C., Revised by Masumi, A.E., "IBAL MANUAL - The Illiac III Basic Assembler Language", July 1971.
UIUCDCS-R-71-470	Nordmann, B.J. Jr., "Speech Display Simulation System for A Comparative Study of Some Visual Speech Displays", August 1971.
UIUCDCS-R-71-472	McCormick, B.H., Nordmann, B.J. Jr., et. al., "ILLIAC III REFERENCE MANUAL - VOLUME IV: Supervisor Organization", August 12, 1971.
UIUCDCS-R-71-473	McCormick, B.H., Nordmann, B. J. Jr., et. al., "ILLIAC III REFERENCE MANUAL - VOLUME III: Input/Output", August 13, 1971.
UIUCDCS-R-71-475	Nordmann, B.J. Jr., "ILLIAC III COMPUTER SYSTEM MANUAL - Taxicrinic Processor, Volume 2", August 24, 1971.
UIUCDCS-R-71-476	Nordmann, B.J. Jr., "ILLIAC III COMPUTER SYSTEM MANUAL - Taxicrinic Processor, Volume 3", August 24, 1971.
UIUCDCS-R-71-479	Nordmann, B.J. Jr., "A Comparative Study of Some Visual Speech Displays", September 10, 1971 (Ph.D. Thesis).
File Note No. 863	Borovec, R. T., "IMAGE 8: The Real Time Clock", June 16, 1971.
File Note No. 864	McCormick, B.H. (ed.), "Illiac III Biblio- graphy Condensed Listing - Supplement", July 13, 1971.
File Note No. 868	Lewis, G.T., "IMAGE 8: Illiac Paper Tape Reader Control", August 10, 1971.

4.6.2 Logic Drawings Issued

The following new logic drawings were issued uring the past quarter:

TP Control Logic	30 drawings
AU Control Logic	17 drawings
Power Turn-on Logic	20 drawings

4.6.3 Engineering Drafting Report

During the past quarter a total of 459 drawings, including new logic drawings, drawing changes, layouts, flow-charts, theses, report drawings and drawings related to the Opto/Mechanical design of the Illiac III project have been processed by the 2118 drafting section.

4.7 ADMINISTRATION

4.7.1 Personnel Report

Senior Staff

Professor Bruce H. McCormick - Principal Investigator
Assistant Professor R. S. Michalski

Professional Staff

Robert C. Amendola
Richard T. Borovec
John S. Read

Research Engineering Assistant

S. Paul Krabbe

Electronic Engineering Assistant

Joseph V. Wenta

Digital Computer Technician II

George T. Lewis

Drafting

Stanislavs Zundo

Research Assistants

Jerry Chen
Walter Donovan
Steve Fierce
Lakshmi Goyal
S. N. Jayaramamurthy
Ahmad E. Masumi
Kiyoshi Maruyama
Frank Murakawa
Dan Pitt
Peter Raulefs
Kuo Wen

Secretarial

Mrs. Judy Arter

4.7.2 Computer Usage Log Summaries

1. The PDP/8e was used approximately 250 hours.
2. The Scanner-Monitor-Video System was used as follows:

Production and Demonstration	129 hours	35 minutes
Preventive Maintenance	12 hours	10 minutes
Corrective Maintenance	27 hours	0 minutes
3. The IBM/75 utilization totaled \$3,878.85.

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

(See Instructions on Reverse Side)

1. AEC REPORT NO.
C00-2118-0025

2. TITLE
Illiacc III Quarterly Progress Report

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)

Bruce H. McCormick
Professor of Computer Science and Physics

Organization
Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature

Date

12/1/71

FOR AEC USE ONLY

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

8. PATENT CLEARANCE:

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

5. ILLIAC IV

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

The contract between Automation Technology Incorporated and the University of Illinois expired during the quarter. Under its new contract the Center for Advanced Computation has prime responsibility for applications on the ILLIAC IV. Progress will be reported under Contract No. DAHCO4-72-C-0001 monitored by the United States Army Research Office-Durham.

The ILLIAC IV software neared completion this quarter. Users are in a position, via the ILLIAC IV simulator, to get first hand experience with all the ILLIAC IV languages and library routines in all areas except the operating system. The operating system has been integrated with the ILLIAC IV simulator, and all modules have been tested in concert. A more sophisticated version is being developed that will allow simulation using the actual I/O subsystem when it becomes available.

The operation of the B6500 installation at the University of Illinois has improved considerably during the quarter. The system is now available approximately ninety percent of the time.

Project expenditures through September 1971:

Burroughs Corporation	\$27,159,707.00
University of Illinois	8,257,550.00*

*Estimate.

HARDWARE

The contract between Automation Technology Incorporated and the University of Illinois expired during the quarter. No future hardware reports will be made.

SOFTWARE

5.1 Operating System

The three major Operating System activities progressed well during the quarter. The first, integration of the ILLIAC IV Operating System with the ILLIAC IV Simulator, was successfully completed in July and has been undergoing refinement since then. This integration is known locally as the "Link Trainer," mainly because it allows users to submit ILLIAC IV jobs to the B6500 just as if they were submitting them to the B6500-ILLIAC IV constellation. Even though there are several restrictions, a number of test cases were successfully run with ASK and GLYPNIR programs. During the course of integration a number of minor incompatibilities within the operating system were found and corrected. Work is now proceeding on the next stage of integration — the "Steerman Trainer," which, although still using the simulator, will exercise the ILLIAC IV I/O subsystem. This work is proceeding well but will soon be blocked by the unavailability of the I/O subsystem at Paoli.

Secondly, the interface between the University Operating System and the Burroughs on-line diagnostics was defined and is being implemented by both parties.

The third system, the interface with the debugging package being developed by Bolt, Beranek and Newman, has been designed. Both the University and Bolt, Beranek and Newman are working on modifications to OS4 (the ILLIAC Resident Module of the Operating System) to accommodate this package.

Preparations are under way to transfer from the B6500 to ILLIAC IV some modules of the Operating System whose algorithms are inherently parallel. This work is being done so that these changes can be made as soon as the present Operating System has been checked out on the hardware.

5.2 COCKROACH

The virtual completion of a production version of the COCKROACH-to-GLYPNIR translator and its associated documentation was accomplished. Much time was spent improving error detection and recov-

ery facilities in the translator. Various storage allocation studies for arrays of row dimension greater than 64 were investigated with a view to selecting one which provides efficient calculation and ease of column and plane indexing for implementation into the translator. Implementation (with some restriction) occurred for array storage and manipulation facilities for arrays of any dimension. Straight storage and physical skewing were selected after considering the results of the storage allocation study. Binary vectors longer than 64 were executed. These provided mode vectors for the array scheme.

The loop variable was used in place of the mode asterisk. At the suggestion of users the construction

DP #1 = BINARYVARIABLE

was admitted into COCKROACH. This removes the need to repunch parallel sections of the code. The provision of a COCKROACH-to-FORTRAN translator on the B6500 allows users to develop parallel algorithms in COCKROACH without the high overhead of simulating on the ILLIAC IV Simulator. This work is now nearing completion. The COCKROACH translator was translated from Burroughs Compatible ALGOL into Burroughs Extended ALGOL.

5.3 GLYPNIR

Maintenance and improvement of the GLYPNIR compiler continued over the quarter. There were three main activities: 1) The compiler was translated from Burroughs Compatible ALGOL to Burroughs Extended ALGOL. This raised its compile speed from about 200 to about 350 cards per minute. 2) Work continued on the new and faster scanner for GLYPNIR. This will also provide macro facilities for the language. 3) ILLIAC (as opposed to simulator) I/O facilities are being added to the language, and work is progressing well in this area.

Because the GLYPNIR compiler is being used by many and is working well, its development was slowed down. Some of the GLYPNIR team were temporarily re-assigned to more urgent work such as managing the B6500 installation and helping with Operating System development.

5.4 ILLIAC IV Assembler (ASK)

The Assembler underwent considerable refinement over the

quarter. Its present peak compiling speed for GLYPNIR generated code is 2300 cards per minute. It was made into a standard B6500 compiler, and facilities were added to allow a "compile and simulate." No significant changes to the language itself were made during the speedup.

5.5 Documentation

Reference and maintenance manuals for all major software modules developed by the University were completed during the period. In addition, introductory manuals were provided to help people become acquainted with the B6500 and the ILLIAC control language. They are available for distribution.

5.6 Burroughs B6500 Facility at Illinois

The performance of the B6500 installation at the University of Illinois improved considerably during the quarter. In mid-July the air-conditioning system was settled into its summer regime and since then has given little trouble.

In the beginning of August level 1.10 software was used. This checked the rising rate of "Halt Loads" (when the processor stops and has to be re-initiated by hand). Since then the Halt Load value has remained constant at about 100 a week. The middle of August saw a marked improvement in hardware faults. The average amount of downtime attributable to hardware malfunction was less than 8 hours a week.

Two extra memory modules of 16K words each have been obtained. One was connected in mid-September with resulting improvement in machine performance. At present about 10,000 jobs are processed weekly by the B6500. Detailed accounting procedures will be implemented on the machine in the near future.

APPLICATIONS

The Center for Advanced Computation of the University of Illinois at Urbana-Champaign now has responsibility for applications on the ILLIAC. IV. Progress will be reported under Contract No. DAHCO4 72-C-0001 monitored by the United States Army Research Office-Durham.

ADMINISTRATION

5.7 Financial Report

Expenditures and obligations for July, August and September
1971

	<u>July-Actual</u>	<u>August-Estimate</u>	<u>September-Estimate</u>
Burroughs.	\$ 26, 265.00	\$155,397.00	Not available
University	\$289,996.00	\$121,000.00	\$55,000.00

Estimated expenditures and obligations through September 1971

Burroughs Corporation	\$27,159,707.00
University of Illinois	\$ 8,257,550.00

THESES

- Kelly, M. W. "Reliability of Computer Systems." Master's Thesis.
Department of Computer Science, University of Illinois at
Urbana-Champaign, 1971.
- Miura, Ken'ichi. "Parallel Radiation Transport Code: A Monte Carlo
Method Applied to ILLIAC IV and Its Statistical Consideration."
Master's Thesis. Department of Computer Science, University
of Illinois at Urbana-Champaign, 1971.

DOCUMENTS

- Abel, N. "A FORTRAN-based Language for Array Processing." ILLIAC IV
Document No. 246, ILLIAC IV Project, University of Illinois
at Urbana-Champaign (March 30, 1971).
- Arnold, C. "Roth's Algorithm for Diagnostic Test Generation: Theory
and Implementation." ILLIAC IV Document No. 240, ILLIAC IV
Project, University of Illinois at Urbana-Champaign (January
22, 1971).
- Knapp, M. and D. McIntyre. "Bulk Storage Applications in the ILLIAC IV
System." ILLIAC IV Document No. 250, ILLIAC IV Project,
University of Illinois at Urbana-Champaign (August 3, 1971).
- Nakamoto, H. and A. Hashimoto. "Optimal Frequency for Hardware Confi-
dence Testing." ILLIAC IV Document No. 241, ILLIAC IV Project,
University of Illinois at Urbana-Champaign (January 22, 1971).
- S. T. "ICL Your Fancy." ILLIAC IV Document No. 252, ILLIAC IV Project,
University of Illinois at Urbana-Champaign (September 1, 1971).

6. SWITCHING THEORY AND LOGICAL DESIGN

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

The branch-and-bound method finds an optimal network early during computation but proving the optimality of that network takes the majority of computation time. So we tried to find procedures to find an optimal network as early as possible. If such procedures are available, a reasonably good network could be obtained even if a computer is halted. New procedures which appear to be more powerful than any currently known were devised and we started to program them. How powerful they could be is to be seen soon.

Optimal one-bit adders were further designed for additional combinations of different types of gates.

S. MUROGA

In order to obtain a near-optimum network of NOR gates, one approach is to transform a network into one of less cost without altering the output functions. We developed five such transformation procedures.

- I: (1) A sufficient condition to detect redundant connections and gates in a network.
(2) A necessary and sufficient condition to detect redundant connections and gates in a network.
- II: A procedure for transforming a network by adding connections and then removing different connections.
- III: A procedure to remove more than one connection or gate simultaneously.
- IV: A procedure to remove one gate using I (1) and III.

V: A procedure to remove one gate (a generalization of the procedures of Ellis).

Procedure I (1) is a generalization of a previous procedure developed by our group.

Although these procedures appear more effective than previously known transformation procedures, further modifications and generalizations may be necessary.

(Y. Kambayashi)

The study of network transformations by which a non-optimal network of gates is reduced to a (hopefully) near-optimal network (while preserving the original output functions) was continued.

Procedures for five such transformations were outlined by Kambayashi. Of these, Procedures II, IV and I (1) were to be programmed by me. Programming is nearly completed, with Procedure II apparently debugged and operating.

Not enough results have been obtained so far to permit a reasonable discussion of the effectiveness of these procedures.

(J. N. Culliney)

Two procedures, procedures I and V, to transform a given NOR network to a network with lower cost were coded into the following three FORTRAN subroutines:

RDTCNT: Remove all single-redundant connections by necessary and sufficient conditions.

An entry point REMCNT (GY, GZ) was included in this subroutine.

By calling this entry point the connection from GY to GZ will be examined to see whether it is removable or not.

PROCV: Check whether a gate GI can be removed or not by adding connections to the gates originally fed by GI.

RQRNW: Calculate requirements for each gate except GI, and for each connection from GI. This subroutine will be called by PROCV.

A main program to read in NOR networks and to transform them to networks with lower cost by calling the above two procedures was coded.

(H. C. Lai)

The synthesis problem of networks for negative functions by MOS cells with a minimum number of FET'S was continued. Some preliminary results on the synthesis of optimal MOS networks for multi-output functions were obtained.

The design of optimal one-bit full adder with different types of gates was considered again. In addition to the four types of networks in Report 425, optimal networks of nine different combinations of restrictions were obtained and cataloged.

(T. K. Liu)

Report No. 455

Nakagawa, Tomoyasu and Saburo Muroga, "Comparison of the implicit enumeration method and the branch-and-bound method for logical design," Department of Computer Science, University of Illinois, June 1971, 94 pages. (The implicit enumeration method and the branch-and-bound method are compared in terms of ease of programming and computation time when they are applied to logical design. Generally the latter is faster in computation time but more time-consuming in programming than the former.)

Report No. 462

Nakagawa, Tomoyasu, "A branch-and-bound algorithm for optimal AND-OR networks (The algorithm description)," Department of Computer Science, University of Illinois, June 1971, 39 pages. (A branch-and-bound algorithm to derive optimal networks of AND and OR gates for a given switching function is described.)

Report No. 471

Nakagawa, Tomoyasu, Hung Chi Lai, and Saburo Muroga, "Pruning and branching methods for designing optimal networks by the branch-and-bound method," Department of Computer Science, University of Illinois, August 1971, 59 pages. (When optimal networks are designed by the branch-and-bound method, an optimal network is found early during computation but proving its optimality consumes major part of the computation. In this paper some procedures to further speed up the finding of an optimal network are discussed.)

Report No. 480

Culliney, Jay Niel, "On the synthesis by integer programming of optimal NOR gate networks for four variable switching functions," Department of Computer Science, University of Illinois, September 1971, 57 pages. (Optimal networks of NOR gates are designed by the implicit enumeration method for some functions of four variables.)

Operations Research, Vol. 19, No. 4

Baugh, C.R., T. Ibaraki, and S. Muroga, "Results in using Gomory's all-integer algorithm to design optimum logic networks," July-August 1971, pp. 1090-1096. (Optimal networks of NOR gates are designed by Gomory's all-integer integer algorithm. Computational efficiency for this logical design problem is discussed.)

7. SOUPAC

(Statistically Oriented Users Programming and Consulting)

July - September 1971

SCANSOUP to codecheck a SOUPAC program on Express is available.

Several new programs also became available to users:

MULTIVARIATE PROBIT and the REGRESSION Package,
ORTHOGONAL PROCRUSTES, VIEWPOINTS and VARISIM for factor analysis,
CROSS-SPECTRAL ANALYSIS for autocorrelations, and
MANN-WHITNEY for nonparametric statistics.

Documentation for these programs and revised write-ups for roughly 50% of the older SOUPAC programs will appear in the next SOUPAC manual.

A procedure called TESTSLOT which allows any functioning FORTRAN program to be temporarily incorporated into SOUPAC has long been available. Recently this technique was used by a Commerce class to run a specialized routine in connection with the statistics of SOUPAC.

A SOUPAC consultant has been keeping regular office hours at Commerce. The purpose is twofold: to help the many Commerce users with their SOUPAC problems near their terminal and to interact with Commerce personnel on questions of mutual concern. A Seminar with Commerce faculty on SOUPAC was scheduled for October.

Of the new programs added to SOUPAC this quarter, the Regression-Correlations Package is likely to have the most far-reaching consequences. This program was conceived and written so that users desiring a number of statistics derived from the same correlation matrix or a subset of that matrix could obtain their results more efficiently. Even in SOUPAC, a series of calls to the same program, each one involving a calculation of

of the correlation matrix, might be necessary to do several regressions from the same data. The recalculation has now been eliminated. Matrix manipulations to select variables are also not necessary. A single statement executed under the control of Regression selects and reorders variables as needed. The programs available under Regression are Simple, Multiple and Stepwise Regression, and Partial and Canonical Correlation. The previous versions of these programs are still available for users who do not require the power of Regression.

SOUPAC usage during the period was more than 3600 jobs per month or 117 jobs per day, although useage in September was less than half that of July due to vacations.

October 30, 1971
JM;jk

8. MACHINE AND SOFTWARE ORGANIZATION STUDIES

(Supported in part by the National Science Foundation under Grant Number US NSF GJ 27446)

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

8.1. Fortran Parallelism Detection - (S. Chen)

The programs to extract parallelism have been largely rewritten. The old scanner generated the tables of information for the DO-Loop subprogram in such a way that the analysis of DO loops had to start from the innermost independent blocks and progress outward treating each new block independently. This strategy might inhibit some potential parallelism which could otherwise be exploited by combining the inner blocks with the outer one. For example, the following structure can be transformed to the equivalent structure without effecting its logic, but with more parallel operations found by the new extracting programs.

DO I=1, 10	DO I=1, 10
A(I)=B(I)	A(I)=B(I)
DO J=1, 10	DO I=1, 10
C(I,J)=A(I,J)+D(J)	DO J=1, 10
	C(I,J)=A(I,J)+D(J)

After scanning a real program, the new extracting programs use a DOHANDLE routine to keep the list structure of the original DO loop, and rearrange the combination of blocks before its actual analysis of parallelism.

This also leads to reducing a lot of effort (machine time and storage) spent in the old DO-Loop subprogram in order to "synthesize" the block structure from the fixed table information supplied by the old scanner.

In response to this new strategy, the DO-Loop subprogram has been greatly simplified. Now it serves only as an essential part of the DOHANDLE routine, performing the function of finding possible separations within any nested DO loop and evaluating the parallelism in a single statement block after the proper combination has been found.

The old DO-Loop subprogram has been discarded and the new version is integrated into the DOHANDLE routine of the new extracting programs, which have been tested successfully for many real FORTRAN programs. The next step will be the incorporation of the assignment subprogram ASSIGN with its new backsubstitution and recurrence features.

8.2. Tree Machine Simulator - (P. Budnik)

A program to simulate the arithmetic unit and memory of a highly parallel and very flexible computer has been written and debugged. It has been used to estimate execution time for several sections of FORTRAN programs. We plan to expand this program further, and to test different hardware configurations as well as run many more experiments with different FORTRAN programs.

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

During the previous quarter, theorems were developed to parse arithmetic expressions such that tree-height is minimized when the expression consists of a sum of terms or a product of factors, and the operator weights are not uniform. These expressions may be represented by:

$$f = \sum_{i=1}^n t_i \quad \text{and} \quad t = \prod_{i=1}^m f_i,$$

where $\sum$ implies addition or subtraction and $\prod$ implies multiplication or division. Furthermore, if we permit distribution then an algorithm which reduces the tree-height of t was shown to be valid.

A PL/1 program was written during this quarter which accepts a FORTRAN statement, $e=f/t$, as input, and with given operator weights, parses e such that the tree-height of e , $h(e)$, is reduced for all cases except when e becomes $e'=(\prod_{i=1}^n f_i)/(\prod_{i=1}^m f'_i)$, where $m > 1$. This case will be completed during the next quarter. We also include memory access (fetch and store) as an operator and we modify expressions of the form $e=-t_1-t_2-\dots-t_n$ to $e'=e+0$ to eliminate the unary minus problem.

8.4. Simulation Processor - (E. Davis)

A system is being programmed to evaluate performance of the simulation machine. This includes a GPSS simulator of the machine, a PL/1 program to prepare test simulation programs for input, a 360 assembly language routine to extract run time parameters from test programs, and an assembly language routine to insert the parameters into the model when running tests on the simulator. The extraction and insertion routines provide information related to an actual run so that the same execution routines will be simulated. In the next quarter the system should be completed, then experimentation and evaluation can begin.

For those parts of simulation programs where the ratio of decision statements to assignment statements is high, an algorithm has been written which concentrates the decisions into a binary tree free of assignments.

Hardware has been designed for the specific purpose of evaluating the tree. A program has been written which converts FORTRAN arithmetic IF statements into logical IF's which have the two way branches necessary for the tree. Design effort will continue on the interface between the decision evaluator hardware and the processors.

8.5. Memory Hierarchies - (D. Gold)

The collection of algorithms which operate on FORTRAN-like programs to produce latency-free I/O mappings for small primary memory systems have been largely formalized. They are now being committed to paper in a comprehensible form. Furthermore, a program which obtains the I/O patterns for random perturbations of data partitions has been debugged, thus verifying the algorithm.

9. COMPUTER SYSTEMS ANALYSIS

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

The goal of this research is the development of analytical tools for system modeling and analysis of real time computer networks. A queueing theory model for a geographically distributed computer network is being developed. Priority assignment and job dispatching rules for efficient job processing in the network are being investigated.

9.1. Computer Network Modeling (W. McKinney)

We have formally defined the queueing theory model our multiserver system. The model includes J job types and K processors. Each job type has its own arrival rate λ_j and its own queue. Each of the J queues has its own length m_j and ρ_j priority classes. The processing rates for the K processors are given in a matrix μ_{jk} which gives the processing rate of type j jobs on processor K . The use of a processing rate matrix enables us to have identical or heterogeneous processors, and allows each processor to have preferred job types.

To date we have considered nonpriority, nonpreemptive priority, and preemptive priority disciplines. We are currently developing a GPSS model to facilitate studying the effects of priority assignment, job dispatching, and load regulation within the network.

9.2. Center Throughput Analysis (S. Mamrack and W. Barr)

We are currently investigating an algorithm which minimizes the maximum (nonlinear) cost function in throughputting n jobs on 1 computer when the jobs arrive simultaneously and have no precedence constraints. A dynamic programming approach has been successful in minimizing the cost function, but involves a prohibitive number of calculations for large n . Approximation techniques developed so far do not guarantee approaching

optimality. We plan to combine the linear programming and branch-and-bound techniques to reduce the number of calculations by an order of magnitude, and still guarantee optimality. If this work is successful for the n/1 problem we will extend the results to the more general case.

We are also developing a new measure of goodness for evaluating center throughput. We define a measure of goodness, γ , as follows:

$$\gamma = \left(\frac{L_q}{M}\right)^\alpha \left(1 - \frac{M-L_q}{\sum_{i=0}^{R-1} \beta(i)}\right)$$

where

L_q is the number of jobs in the queue

M is the maximum length queue

R is the number of priority classes

and

α and $\beta(i)$ are control parameters.

We are considering a family of $\beta(i)$'s, ranging from the nonpriority case to the preemptive dynamic priority case. After having established the γ function as a desirable measure of goodness, we will relate the α 's and β 's to the center's control parameters which are under the designer's (user's) control. Then, we plan to extend the technique to determine throughput values for a geographically distributed network of computers. The existence of the γ measure will provide us with the capability to compare centers within a net and hence to quantitatively determine criteria for job dispatching between centers.

(E. K. Bowdon)

10. THEORY OF DIGITAL COMPUTER ARITHMETIC

(Supported in part by the National Science Foundation under Grant Number US NSF GJ 813).

10.1. Continued Fraction Arithmetic

A study of the general use of continued fractions in digital computer arithmetic was begun during the quarter. The main goal is to find the class of functions that can be efficiently evaluated by the use of continued fraction methods.

Let f_1 be a given function, and let

$$f_1 = a_0 + \frac{b_1}{a_1} + \frac{b_2}{a_2} + \dots + \frac{b_n}{a_n}.$$

Then the recursion

$$P_i = a_i P_{i-1} + b_i P_{i-2}$$

(1)

$$Q_i = a_i Q_{i-1} + b_i Q_{i-2} \quad i = 2, 3, \dots, n,$$

$$P_0 = 0, P_1 = b_1, Q_0 = 1, Q_1 = a_1$$

defines uniquely f_1 as $f_1 = P_n / Q_n$.

The computational algorithm of (1) involves four multiplications for each iteration. To reduce this number, simple continued fractions were considered, i.e. $b_i = 1$, $i = 1, 2, \dots, n$. Furthermore we hope that a_i can be simplified in the binary sense, i.e. $a_i = 1, 1/2$ or $1/4$ for all i .

Our research began with a study of simple continued fraction presentations of functions as they appear in the literature. The a_i are assumed to be integers, and in many cases they include a variable x .

Example:

$$\tan x = \frac{1}{x} - \frac{1}{2x} - \frac{1}{2x} - \dots$$

The result is that algorithm (1) requires at least two multiplication operations for each iteration, hence this approach was abandoned.

Our new approach is to attempt to develop new algorithms in order to avoid the high number of multiplications.

Amnon Bracha

10.2. Continued Fraction Algorithm for the Square Root

An algorithm to solve the set of quadratics $x^2 + b_k x - c_k = (x-u)(x-v) = 0$ with $b_k = 0$ (i.e., the square rooting problem) is now available. A proof of convergence of the algorithm has also been obtained.

Here u is to be expressed in the form of an infinite continued fraction,

$$u = \frac{1}{q_1} + \frac{1}{q_2} + \dots + \frac{1}{q_i} + \dots$$

where $q_i \in \{1/4, 1/2, 1\}$.

Recursion relations were developed earlier and have been reported.* The problem remaining was to determine the rules for selection of q_i . These rules have now been found.

Assuming that we are dealing with floating point numbers, the mantissa of c_k will be in the range $[1/2, 1]$. But since the exponent can be either odd or even and since for square rooting, we need to have an even exponent, conditional unnormalizing by 1 bit is necessary. Thus the range of the mantissa of c_k doubles. Either $[1/4, 1]$ or $[1/2, 2]$ can be chosen. In this algorithm the former choice has been made. As a result $1/2 \leq u \leq 1$.

Kishor S. Trivedi

*Quarterly technical progress report, Department of Computer Science, University of Illinois, April-June, 1971, pp. 142-143.

10.3. Evaluation of Some Elementary Functions in Radix 16

During this quarter some properties of previously obtained algorithms have been studied. One interesting feature of these algorithms, which are essentially based on a continued product representation of numbers, is the possibility of predicting larger and larger numbers of successive coefficients (s_k) at each step. This represents one way of achieving faster convergence of iterative processes, provided that a corresponding increase is made in the number of levels in a stacked adder structure.

The chosen algorithms have also been checked and the formalization of the results in the form of an M.S. thesis is in progress.

M. D. Ercegovac

Publications

Report No. UIUCDCS-R-477

Diamond, Martin, "A factorization procedure that is equivalent to successive over relaxation," Department of Computer Science, University of Illinois, September, 1971.

Report No. UIUCDCS-R-478

Diamond, Martin, "The divergence of Stone's factorizations when no parameters are used," Department of Computer Science, University of Illinois, September, 1971.

11. COMPUTER LANGUAGES FOR MATHEMATICS AND NUMERICAL ANALYSIS

We present abstracts of Digital Computer Laboratory reports and papers which have appeared during the last six months. This work has been supported by the National Science Foundation under NSF Grant GJ-328.

11.1. COMPUTER LANGUAGES:

Phillips, J. Richard, H. C. Adams, "Dynamic Partitioning for Array Languages. (To appear in CACM.)

ABSTRACT: The classical process of partitioning an array into subarrays is extended to a more useful array language operation. Various modes of partitioning are defined for different types of arrays, so that subarrays may vary over the original array in a nearly arbitrary manner. These definitions are motivated with several realistic examples to illustrate the value of partitioning for array languages.

Of general interest is the data structure for partitioning. This consists of dynamic tree structures which are used to derive and maintain the array control information. These are described in sufficient detail to be of value in the design of other array languages. The description presented in this paper is implemented in a new array language, OL/2, currently under development at the University of Illinois.

11.2. NUMERICAL ANALYSIS:

Saylor, Paul E., "Second Order Strongly Implicit Symmetric Factorization Procedures for the Solution of Elliptic Difference Equations". (To appear.)

ABSTRACT: H. L. Stone has proposed an iterative method, called a factorization procedure, to solve elliptic difference equations. Because of the complexity of the algorithm defining the iteration matrix, Stone described the procedure as strongly implicit. Analysis of its convergence properties is difficult. One reason is that the iteration matrix is non-symmetric. This motivates the construction of symmetric strongly implicit factorization procedures. In this paper the class of symmetric strongly implicit factorization procedures is determined. From this, it is possible to determine the class of such procedures that are second order. This is a property of the Stone procedure that seems fundamental to its success. Since the iteration matrix of any of the resulting procedures is defined by an algorithm with unpleasant properties, no satisfactory second order symmetric strongly implicit factorization procedure exists.

Bracha, Amnon, "A Symmetric Factorization Procedure for the Solution of Elliptic Boundary Value Problems," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 440, April, 1971.

ABSTRACT: The convergence properties of a symmetric factorization procedure for solving elliptic difference equations due to H. L. Stone is studied. Convergence depends on bounds on a parameter, and computable values of these bounds are obtained. The procedure and the results are generalized to an arbitrary number of dimensions.

11.3. ALGORITHMS:

Maruyama, Kiyoshi, "Parallel Methods and Bounds of Evaluating Polynomials," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 437, March, 1971.

ABSTRACT: A lower bound of the maximum evaluable degree of polynomial for each step s is given, and is $N^*(s) > 2^{s(1-\delta)}$, $\delta \approx (2/s)^{1/2}$. An upper bound is derived for the number of steps required to evaluate polynomials of degree n , and is given by $T(P_n) < (1+\epsilon)\log_2 n$, which approaches theoretical lower bound $\lceil \log_2(2n+1) \rceil$ as $n \rightarrow \infty$, where $\epsilon \approx (2/\log_2 n)^{1/2}$.

An upper bound of the number of operations required to achieve the theoretical lower bound of steps required to evaluate polynomials of degree n is given as Cn , $C > 2$. Furthermore a systematic construction of computation trees, multi-folding and modified multi-folding methods, which achieves the bound is given. Our dual problem approach leads to better results than Brent's [1] primal problem approach, and shows the existence of a simple scheduling algorithm to evaluate polynomials within the bound.

Maruyama, Kiyoshi, "A Procedure for Detecting Intersections and Its Application," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 449, May, 1971.

ABSTRACT: The procedure described here employs face-to-face intersection analysis to determine whether two or more polyhedral objects intersect. As means to minimize the number of pairs of faces which should be examined for face-to-face intersection analysis, a solution box approach, mutual divisibility and visibility of two faces are considered. Intersection detection between two faces is done by the determination of their parity mode.

As an application of the procedure developed, a solution for the path finding problem in geometrically constraint space is given.

REFERENCES

- Bracha, Amnon, "A Symmetric Factorization Procedure for the Solution of Elliptic Boundary Value Problems," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 440, April, 1971.
- Maruyama, Kiyoshi, "Parallel Methods and Bounds of Evaluating Polynomials," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 437, March, 1971.
- Maruyama, Kiyoshi, "A Procedure for Detecting Intersections and Its Application," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 449, May, 1971.
- Phillips, J. Richard, H. C. Adams, "Dynamic Partitioning for Array Languages. (To appear in CACM.)
- Saylor, Paul E., "Second Order Strongly Implicit Symmetric Factorization Procedures for the Solution of Elliptic Difference Equations". (To appear.)

12. NUMERICAL METHODS, COMPUTER ARITHMETIC AND ARTIFICIAL LANGUAGES

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

12.1 An Adaptive Algorithm for the Solution of Finite Difference Equations

By modifying the algorithm we developed last quarter, we have obtained a second adaptive algorithm for the solution of large systems of linear equations. Both algorithms solve the system $Ax = q$ by using the factorization iteration

$$(A+B)x_{n+1} = (A+B)x_n - \tau_n (Ax_n - q),$$

where B is the auxiliary matrix defined by the factorization. They are adaptive in the sense that they generate the parameters $\{\tau_n\}$, from quantities computed on previous iterations, so that the rate of convergence approaches the rate obtained when the optimal set of parameters is used.

The sequence of parameters is made up of a number of subsequences, $\tau_0^i, \tau_1^i, \dots, \tau_{N-1}^i$, $i = 1, 2, 3, \dots$, each defined as a Tchebychev sequence on an interval (a_i, b_i) . In the first algorithm, the interval (a_i, b_i) is used to define $x_0^i, \dots, x_{N-1}^i$. Then after determining the improved interval and defining $x_0^{i+1} = x_{N-1}^i$, the iterates $x_1^{i+1}, \dots, x_{N-1}^{i+1}$ are calculated.

In the modified algorithm, x_0^{i+1} is defined as a linear combination of $\{x_n^i\}_{n=0}^{N-1}$ so that x_0^{i+1} equals the vector that would have been calculated if the improved interval had been used in the last N iterations. This modification increases the rate of convergence significantly since the iterates $x_1^i, \dots, x_{N-1}^i$, which are originally calculated without any knowledge of the optimal interval, are combined to yield x_0^2 determined by the interval (a_2, b_2) which is derived using the knowledge gained in the first $N-1$ iterations.

(M. Diamond)

12.2 ETS Operating System

Progress on ETS (An Educational Timesharing System for the PDP-11) has been slowed somewhat by assorted hardware/software/human problems; however, some major check points have been reached.

The timesharing monitor is essentially debugged. It has supported up to 8 consoles interactively for extended periods in test situations. The monitor filing system is also essentially debugged. Some new non-resident service routines are still to be added which will be service modules for specific system software running under the monitor.

A new command string interpreter has been written and is in the debugging stage; it is to replace the present "test only" command string interpreter which is both sensitive to user errors and requires an understanding of the system beyond that which could be expected from a beginning student.

The outline text editor (EDETS) is now complete and undergoing exhaustive testing. Manuals outlining its use are being written.

The assembler (ASSETS) is so far the only software which is not nearing completion. A somewhat restrictive version of DEC's PAL-11A is being adapted to the system for temporary use. The final assembler for ETS is still being designed. We hope to begin coding in about a month.

The interpretative debugging package (DEBETS) is now operational on a "stand-alone" basis. The necessary linking routines which will load DEBETS and a user program and link them together have been written and are being debugged. The interpreter is a very powerful routine which not only

protects the system from the user (the PDP-11 has no relocation or memory protection hardware) but also provides the user with a complete interactive debugging package which provides break points, a full trace mode, a dis-assembler, symbolic referencing, dump facilities and monitoring facilities for variables or tables.

(D. Oxley)

12.3 Computer Aided Instruction

A CAI program (GIZMO) which was written by CS 201 students in the spring of 1971 is fully operational. However, work is proceeding on a new version, again as a project for CS 201 students.

(A. Davis)

12.4 Publications

Three papers have been submitted and accepted for the ACM SIGPLAN Conference on Pedagogic Languages and to be held in January 1972. These are: "The Use of Mini-Computers for Teaching Assembly Language and System Programming," D. B. Gillies, J. D. Miller, Proceedings of the SIGPLAN Symposium on January 1972 (to appear); "Instructional Software for an Assembly Language Course Taught on a Mini-Computer," D. B. Gillies, et. al., Proceedings of the SIGPLAN Symposium on January 1972 (to appear); and "ETS - An Educational Timesharing System for the PDP-11," D. B. Gillies, T. T. Chen, D. W. Oxley, Proceedings of the SIGPLAN Symposium on January 1972 (to appear).

(A. Davis, D. Gillies, J. Hart
S. Kerr, D. Oxley, D. Pasta, K. Phillips)

12.5 PDP-11 Hardware Additions

The Data Products 2410 line printer which was donated to the project by DEC arrived in late August. It was promptly interfaced and

became operational within four days of delivery. A 200 cpm GDI card reader (also donated by DEC) arrived in late September complete with all necessary interface boards. It has performed well even under a production environment requiring continuous operation. The addition of these two peripherals and better operating systems have significantly improved throuput and quality of service.

The 4K Ampex memory interface was completed and is working well. Total core is now 12K words. Design and construction of a ROM module for the PDP-11 and a ROM programmer is underway.

The teletype situation has improved but is still less than ideal. A persistent transient fault was eliminated with a modification to the TTY interrupt PC boards. A test jig is being built for off line testing of the PC boards. Nine TTY's are now on line and being used in testing of ETS. Before his retirement, Mr. Huffman built a prototype sound insulating TTY case. Twelve duplicates were built and delivered by the carpenter shop. The cases will be ready for use as soon as cooling fans can be installed.

(J. Miller)

12.6 General Programming

During this period, extensive experimentation with DEC's Disk Operating System (DOS) for the PDP-11 led to modifications which allowed programs to dynamically invoke other programs. Various other modules were replaced in DOS to allow access to local peripherals. A routine to speed up system loading for DOS and other systems was developed (including DECTape copying facilities).

Hardware diagnostics developed include a multi-station teletype noise generator and a line printer tester.

Debugging aids written include a trace routine and an anti-assembler routine (both are operating system independent).

(R. Atkinson)

12.7 SPOOLING

SPOOLING, a batch system for the PDP-11, became operational during this quarter. This system provides fast turnaround through spooling of all card decks and listed output to disk, simple I/O calls for reading data cards and printing lines, user alterable job time and line limits, and debugging aids in the form of core and register dumps. Using the new peripherals (a 200 cpm card reader and 500 lpm line printer), some 750 student jobs have been run in a 15 hour period without saturation and without assistance from the 360/75.

(J. Miller)

12.8 Assembly Language Instruction

Considerable work has been done to improve the structure and resources of CS 201, the introductory course in assembly language and system programming. Major goals and course content have been established, and a book of instructional problems for the course has been written.

The major goal of the course is to teach the students how to write good programs. By stressing such programming principles as planning, documentation, debugging and optimization, as opposed to simply "working code," we help the student develop a knowledge of what good code is, as well as useful techniques.

The course is divided into three sections: basic programming skills, advanced programming techniques, and a study of assemblers. Basic programming includes PAL order code, addressing modes, simple data structures,

stacks, and internal documentation. Advanced programming techniques cover program structure, subroutine, debugging, internal and external documentation, optimization, tricks, I/O, traps and interrupts. Both top down and bottom up approaches to assemblers are taught in the last third of the course.

The CS 201 Problem Book is a collection of problems suitable for homework, quizzes, and in class discussion. It covers simple problems useful in the first third of the course, as well as more complex problems and suggested machine problems. The book has been bound and distributed to all students and staff. Answers to approximately half of the problems are given in the back of the book. A complete solutions book has been assembled for the teaching staff. As we collect more problems, the Problem Book will be expanded and revised.

(M. Stone, A. Davis, S. Hansen)

13. GENERAL DEPARTMENT INFORMATION

13.1 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	24	4	26.27
Visiting Faculty	2	0	2.00
Graduate Research Assistants	1	68	34.50
Graduate Teaching Assistants	0	27	13.25
Professional Personnel	10	1	10.50
Administrative and Clerical	20	0	20.00
Nonacademic Personnel (Monthly)	19	1	19.50
Nonacademic Personnel (Hourly)	0	50	17.60
TOTAL	76	151	143.62

*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, M. F. Faïman, H. G. Friedman, C. W. Gear, D. B. Gillies, W. J. Kubitz, D. J. Kuck, B. H. McCormick, R. G. Montanelli, 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, J. E. Vander Mey, D. S. Watanabe, and T. Wilcox.

13.2 Bibliography

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

File Numbers

- No. 863 Borovec, R. T., "IMAGE 8: The Real Time Clock", June 16, 1971.
- No. 864 McCormick, B. H., "ILLIAC III Bibliography Condensed Listing", July 13, 1971.
- No. 865 Haskin, R., "GRAPHICS 8: SYSTEM MAINTENANCE (SOFTWARE)", July 1971.
- No. 866 Nickolls, J. and Michel, M. J., "GRAPHICS 8: System Maintenance (Hardware)", August 1971.
- No. 867 Michel, M. J. and Haskin, R., "GRASS: Extended Remote Facilities Guide", August 1971.
- No. 868 Lewis, G. T., "IMAGE 8: Elliott Paper Tape Reader Control", August 10, 1971.

Report Numbers

- No. 250 Knapp, M. and McIntyre, D., "Bulk Storage Applications in the ILLIAC IV System", August 3, 1971.
- No. 252 Abel, N., "ICL Your Fancy", September 1, 1971.
- No. 465 Michel, M. J., "GRASS: SYSTEM OVERVIEW", July 1971.
- No. 466 Haskin, R., Nickolls, J., and Michel, M., "GRASS: Remote Facilities Guide", July 1971.
- No. 467 Michel, M. J., and Kock, J., "GRASS: Terminal User's Guide", August 1971.
- No. 468 Michel, M. J., "GRASS: System Software Description", August 1971.
- No. 469 Schwebel, J. C., (Revised by Ahmad E. Masumi), "IBAL MANUAL: The Illiac III Basic Assembler Language", July 1971.
- No. 470 Nordmann, B. J., Jr., "SPEECH DISPLAY SIMULATION SYSTEM FOR A COMPARATIVE STUDY OF SOME VISUAL SPEECH DISPLAYS", August 1971.
- No. 471 Nakagawa, T., Lai, H.C., and Muroga, S., "PRUNING AND BRANCHING METHODS FOR DESIGNING OPTIMAL NETWORKS BY THE BRANCH-AND-BOUND METHOD", August 1971.

Bibliography (cont.)

- No. 472 McCormick, B. H., Nordmann, B. J., Jr., Atkins, D. E., Borovec, R. T., Goyal, L. N., Katch, L. M., Lansford, R. M., Schwebel, J. C., and Tareski, V. G., "ILLIAC III REFERENCE MANUAL VOLUME IV: Supervisor Organization", August 12, 1971.
- No. 473 McCormick, B. H., Nordmann, B. J., Jr., Atkins, D. E., Borovec, R. T., Goyal, L. N., Katch, L. M., Lansford, R. M., Schwebel, J. C., and Tareski, V. G., "ILLIAC III REFERENCE MANUAL VOLUME IV: Supervisor Organization", August 12, 1971.
- No. 475 Nordmann, B. J., Jr., "ILLIAC III COMPUTER SYSTEM MANUAL: TAXICRINIC PROCESSOR VOLUME 2", August 24, 1971.
- No. 476 Nordmann, B. J., Jr., "ILLIAC III COMPUTER SYSTEMS MANUAL: TAXICRINIC PROCESSOR VOLUME 3", August 24, 1971.
- No. 477 Diamond, M., "A FACTORIZATION PROCEDURE THAT IS EQUIVALENT TO SUCCESSIVE OVER RELAXATION", September 1971.
- No. 478 Diamond, M., "THE DIVERGENCE OF STONE'S FACTORIZATIONS WHEN NO PARAMETERS ARE USED", September 1971.

Theses

- No. --- Kelly, M. W., "Reliability of Computer Systems ", (M.S.), August 1971. (To be published).
- No. 464 Partridge, R.L., "PAGAN: A THREE DIMENSIONAL PATTERN GENERATOR", August 1971.
- No. 474 Alster, J. M., "HEURISTIC ALGORITHMS FOR CONSTRUCTING NEAR-OPTIMAL DECISION TREES", August 17, 1971, (M.S.).
- No. 479 Nordmann, Jr., B.J., "A COMPARATIVE STUDY OF SOME VISUAL SPEECH DISPLAYS", September 10, 1971, (PhD.).
- No. 480 Culliney, J. N., "ON THE SYNTHESIS BY INTEGER PROGRAMMING OF OPTIMAL NOR GATE NETWORKS FOR FOUR VARIABLE SWITCHING FUNCTIONS", September 1971, (M.S.).
- No. 485 Kodimer, D. A., "ALPHECON: EVALUATION OF A DEVELOPMENTAL VIDEO STORAGE TUBE", September 1971, (M.S.).

13.3 Colloquia

" The Distributed Computer System", by Professor David J. Farber,
Department of Information and Computer Science, University of Cal-
ifornia at Irvine, Irvine, California 92664, September 27, 1971.

13.4 Drafting

During the third quarter, a total of 234 drawings were processed by the general departmental drafting section:

Large Drawings	96
Medium Drawings	27
Small Drawings	68
Layouts	0
Report Drawings	20
Change Order Drawings	14
Miscellaneous	<u>9</u>
Total	234

(M. Goebel)

13.5 Shop's Production

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

	<u>AEC 2118</u>	<u>AEC 1469</u>	<u>Other</u>
Machine Shop		15	3
Electronic Shop	2	64	7
Chemical Shop	2	65	6
Layout Shop		61	5

(F. P. Serio)

DEPARTMENT OF COMPUTER SCIENCE
GRADUATE COLLEGE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

REPORT REQUEST FORM

Report Number

Title

_____
_____
_____
_____
_____

_____
_____
_____
_____
_____

Fold and Staple as shown

NAME:

_____
_____
_____
_____

ADDRESS:

Fill out Mailing Label

NAME:

_____
_____
_____
_____

ADDRESS:

CUT ALONG THIS LINE

FOLD

Stamp

Mail Room
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

FOLD

STAPLE