ESD-TR-68-445

ESLF

ESD ACCESSION LIST

ESTI Call No. 64306

Copy No. 1 of 2 cys.

MTR-805

THE FIRST ACM WORKSHOP ON MICROPROGRAMMING

Joseph E. Sullivan

ESD RECORD COPY

RETURN TO
SCIENTIFIC & TECHNICAL INFORMATION DIVISION
(ESTI), BUILDING 1211

FEBRUARY 1969

Prepared for

DIRECTORATE OF PLANNING AND TECHNOLOGY

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 7120
Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
Contract F 19(628)-68-C-0365

THE FIRST ACM WORKSHOP ON MICROPROGRAMMING

Joseph E. Sullivan

FEBRUARY 1969

Prepared for

DIRECTORATE OF PLANNING AND TECHNOLOGY

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

This Workshop on Microprogramming was hosted by The MITRE Corporation, Bedford, Massachusetts, as part of the work effort for Project 7120 of the Electronic Systems Division. This report was prepared under contract number F19(628)-68-C-0365.

ABSTRACT

The Workshop on Microprogramming, sponsored
by the ACM with the cooperation of MITRE, took place
at MITRE on October 7-8, 1968. This paper is primarily
an historical account and summary of the workshop's
business, with some commentary.

ACKNOWLEDGMENT

TABLE OF CONTENTS

SECTION I

1.0 INTRODUCTION

The Workshop on Microprogramming, sponsored by the ACM with the cooperation of MITRE, took place at MITRE on October 7-8, 1968. The purposes of the workshop were to

- identify the work currently being done in the field;

- promote communication between microprogramming practitioners;

- identify promising new directions for investigation.

Ninety-one individuals attended the workshop: fifty-seven from industry, twenty-three from the academic world or university-associated laboratories, and eleven from non-profits or government. Eighty-nine were from the United States, with one each from Italy and Great Britain.

The workshop was conducted in four sessions. The first three were relatively formal, with scheduled speakers, and the fourth a less formal discussion session.

1.1 Current Work

Personnel from some seventy-three projects, more or less, were represented. These could be classified roughly as follows:

General Study or Teaching of Microprogramming (17)

The Design of Support or Automation Processes for Microprogramming (10)

The Use of Microprogramming for:

Higher-Level Language Support (8)
Emulation or Extension of Instructions Sets (11)
Direct Applications or Process Control (13)
Machine Architecture (14)

1

1.2 Communication

Several of the speakers alluded to microprogramming as a "bridge" between hardware and software. When someone schooled in hardware talked about hardware, and when software experts discussed software, the conference indeed served as a bridge of communication between the two groups. However, whenever interdisciplinary aspects of microprogramming were considered, the two points of view tended to diverge. In these cases, the workshop served as a forum for debate and as a medium for mutual astonishment, if not enlightenment, between the hardware and software schools. Even the definition of microprogramming was not resolved because, of course, each tradition sees this new technology as an extension of, and in terms of, its "old" technology - and the terms differ. The subjects of who should do microprogramming, how he should do it, and what he should use it for, likewise touched off much controversy, especially in Session IV. It should be remarked that the present writer is of the software school.

1.3 New Areas of High Potential Payoff

Emerging from the workshop were two central themes that, though not really new, will clearly become more important as the state of the art advances.

One of these might be called "tradeoff technology". While several papers were delivered touching on the subject, and while anguished cries would be heard for "guidelines" on the use of microprogramming, it was fairly clear that no practical work had been done towards establishing a set of rules for drawing near-optimum lines between hardware, firmware, and software portions of a system. Almost by definition, a significant payoff will reward anyone who can dent this problem.

The other subject that kept coming up was writable control stores. The topic generated much controversy (Session IV) but it is obvious that writable control stores will be produced and marketed as soon as the technology and market support them - and the market, it appears, is waiting. The proper use of this facility, so as to extract from it a maximum measure of the computer power and flexibility it seems to offer, is a virtually unexplored technology.

SECTION II

2.0 SESSION I

"THEORY OF MICROPROGRAMMING"

Chairman: Professor H. Gray
University of Pennsylvania

Microprogramming - interpreted as implementing
control logic, primarily by read-only storage -
cuts across the specialties of electronic module
design, logic design, mechanical languages, pro-
gramming, and system architecture. Microprogramming
is, therefore, a promising means for designing
integrated hardware-software systems. In this
session we will discuss topics fundamental to
microprogramming and these related specialties.

— from the Workshop Bulletin

M. V. Wilkes

The Growth of Interest in Microprogramming

Appropriately enough, the first session of the workshop was
opened by the man generally credited with introducing the term
microprogramming in the sense of "implementing control logic by a
control store - usually a read-only store".

Professor Wilkes attributed the origins of microprogramming to
dissatisfaction with less systematic methods of system design and
implementation. He traced the history of microprogramming through
an early period of interest (early 1960's) and through successive
periods of declining and (now) resurging interest. The intermediate
decline he attributed to a general lack of successful application in
the early days, due to (1) the lack of a fast and cheap ROS, and
(2) the shift in emphasis away from machine level efficiency that
accompanied a growing acceptance of higher-level languages. The
present renewal of interest, he claimed, is due to advances in memory
technology and the influence of the IBM 360 architecture, which
demonstrates the feasibility of implementing similar instruction sets
in small and large machines.

Professor Wilkes recalled his earlier opposition to modifiable control stores - an opposition based on an expected "chaos" that would result if every programmer had his own instruction set. He softened this position only to the extent that modification of the control store is reserved for special "system" programs or programmers using now well-understood protection techniques.

A. Tonik

Tradeoffs for When and How to Use Microprogramming

Mr. Tonik contrasted the "microbird's eye" view of a processing system as consisting of adders, shifters, etc., with the other view which sees it as an elaborate array of logic - gates, decoding networks, and the like. He listed the liabilities of the microprogramming approach as (1) a slower cycle due to the need for control memory access, and (2) an increase in hardware (including, presumably, the control memory itself). To substantiate the latter, he estimated that 900 modes in a small conventional machine are equivalent to 50,000 control memory diodes, and that 8,000 modes in a large conventional machine must be replaced by 300,000 bits of control memory to obtain an equivalent microprogrammed machine - plus, of course, the microprogram control circuits in both cases.

The standard list of advantages for microprogramming as a machine design tool was discussed: the ability to add or modify instructions, precheck a design by simulation, emulation, etc.

The architecture of the micromachine itself was also treated at length. One of the questions discussed was the best place for keeping the address of the next instruction - (1) in the instruction word itself or, (2) in a separate counter. The figures quoted were: a 15-35% larger word results if method (1) is followed; 10-25% more words (the unconditional branches) are used with method (2). Although the second alternative thus seems to be preferable in most circumstances, Mr. Tonik pointed out that one-instruction subroutines and a 5-10% improvement in running speed can be realized with the first method.

On the subject of writable control stores, Mr. Tonik pointed out the possibility of dynamic modification of the control, but said he was not in favor of it. (This, of course, is not quite the same thing as "user" modification of the control store in a non-dynamic fashion.)

5

Y. Chu

A Higher-Order Language for Describing Microprograms

Dr. Chu described a language suitable for the description of a micromachine - its hardware registers and state transitions. A broad-brush description of the language's main features and syntax was given, and a sample "program" in the language was exhibited. A simulator (for testing micromachine architectures), which accepts the language as input, has been constructed.

Although it would not be possible to evaluate the language itself from the level of detail in the talk, it is clear that such a language is valuable not only as a tool for design and design testing, but also as a means of rigorously defining a machine to microprogrammers - that would be a significant advance over functional (English) machine descriptions and/or wiring diagrams - particularly if used to supplement them.

G. Y. Wang

Micro-program Memory Technology

Mr. Wang discussed current developments in high-speed memory technology. The read-write characteristics of the technologies discussed are summarized in the following chart, taken from the first slide of the presentation.

Technology		Destructive Read-Out	Non-Destructive Read-Out	
			ROM	Electrically Alterable
Magnetic Core	1 core/bit	X	X	
	2 core/bit	X		X
	multi-aperture (BIAX)			X
Magnetic Thin Film	plated wire	X	X	X
	flat film	X	X	X
	mated film			X
Solid-State LSI	MOS		X	X
	Bipolar		X	X
	Diode Array		X	
Linear Coupler	Inductive		X	
	Capacitive		X	
	Resistive		X	

C. V. Ramamoorthy

User Microprogrammable Computers

Professor Ramamoorthy defined a "user microprogrammable computer" as one in which the user has some (restricted) access to the microprogrammable store. He distinguished two classes of users: (1) those who own the machine system and (2) those who solve their problems on the system. The first group he described as interested primarily in maximum utilization and also such specific problems as system expansion/ changeover transition problems, emulation, user-user and user-system protection, automated surveillance, and the like. The second group he characterized as interested mainly in maximum convenience, the "naturalness" of the languages provided, the production performance, turn-around and other aspects of program production, and, possibly, real-time dependency of the problem program.

The user of either class with the desire and fortitude to formulate his own solutions to his problems can make good use of microprogramming, particularly from a timing standpoint. However, it was pointed out that merely giving access to the microprogrammed store is not enough; the following requirements (or, at least, desirable characteristics) were also listed:

1) A writable control store;

2) Reasonable user cost;

3) Simple micro-commands;

4) A language capable of describing micro-operations;

5) Relocatability and reentrancy of user microprograms;

6) Flexible addressing (e.g. vector, matrix, proximity);

7) Simplicity of the "visible" machine;

8) Parallel surveillance and user-user, user-system protection.

L. L. Rakoczi

Implications of Dynamically Changeable Microprogram Memory

Mr. Rakoczi gave a spirited presentation on what he called the "fourth generation" computer architecture: the "machine within a machine" - that is, a dynamically modifiable control store. Used in a more or less obvious way, such stores permit multiple emulators to be run on the same processor under a central control and with very little switch-over time. Mr. Rakoczi cited several successful emulation projects along these lines. More exciting was the "6000-E" machine which actually permits something close to dynamic modification of the control store (with restrictions, of course). The assembler for this machine allows new instructions to be defined in terms of the microcommand sequence which realize them; these get loaded into the control store when the "native" level program is loaded into the main store.

The talk touched off a lively discussion on the subject of writable control stores; it became apparent for the first time that opinions on the matter differed widely and that the subject would occupy much of the workshop's attention.

3.0 SESSION II

"MICROPROGRAMMING PRACTICES AND TECHNIQUES"

Chairman: Dr. R. Merwin
DOD

This session will concentrate on the mechanics
and techniques for using microprogramming to
implement hardware projects. Main topics of
discussion will be: control store designs,
simulation of microprogrammed systems, tech-
niques for writing machine-descriptions, and
the writing of associated microprograms.

Hardware and software tools required for these
tasks and descriptions of specific approaches to
microprogrammed computer systems will be discussed.
Tradeoffs between control and operational hardware
and a transformation technique for evaluating
these, based on microprogramming techniques, will
also be covered.

—— from the Workshop Bulletin

G. E. Hoernes and L. Hellerman

Experimental List-Type Micro-Code Assembler

Messrs. Hoernes and Hellerman discussed the motivation for, and
characteristics of, a "list-type" assembler to supplant the "box-type"
assemblers now in use at IBM. The intention is to utilize "conventional"
techniques and to ease the transition from higher-level to micro-code
for applications programmers with a need to perform optimization.

The assembler described is itself coded in PL/I. It has
"equation"-format input and a number of output options to satisfy the
needs of various users: the programmer who wants to see the logic
flow, the engineer who wants to see what bits are set in the resulting
control word, and the customer engineer who may want both.

The talk dealt mainly with problems peculiar to IBM's own
environment. For example, terms such as "edge characters" - meaningless
except in the context of IBM's "box" assemblers - were used. On the
whole, though, the talk was fairly well recieved.

11

J. R. Vollbrecht

Microprogramming Design Aid System (MIDAS)

MIDAS, as described by Mr. Vollbrecht, appears to be a standard set of microprogramming utilities - an assembler, a simulator, a flow charter, a manufacturing data puncher, etc., - remarkable mainly in that they are organized under a coherent "control" program. Well-established software utility principles seem to have been applied with considerable success.

G. Hoff

Development of a Microprogramming System for the H4200

Mr. Hoff's presentation was a fascinating (to a programmer) view
of microprogramming from a logic designer's standpoint. The machine
described - the H4200 - was logic designed and microprogrammed by the
same people, and the impression was that these were not, primarily,
programmers. Consequently, the project could be considered a case
study of the processor architecture that results under such circum-
stances.

The most significant design constraint was that the H4200 requires
a high degree of parallelism in order to keep up with the memory while
processing punctuation-delimited data fields. The micromachine command
structure arrived at to satisfy this requirement might be described
as a "45°" machine partaking of some "horizontal" (one-bit-per-gate)
qualities and some "vertical" (bus or register-oriented) qualities.
It had, for example, curious six-way branches which do not occur
until after the instruction following the branch has already been
executed.

13

E. Stabler

Microprogram - Micro-Operation Transformations

Professor Stabler described an automatic process for determining appropriate state reductions - in effect moving logic elements from the control unit to the operations unit of a computer. The method represents a theoretical approach to the important practical problem of balancing hardware-software tradeoffs.

It was not clear that the method had practical application in its present form, inasmuch as the sequential-state description of a typical process (say, a floating-add) in a real computer would be a formidable task, and in any case one cannot be sure that the reduced process is unique and independent of the original detailed process. Nevertheless, the study has potential practical use as well as theoretical interest.

G. S. Badger

The Use of Microprogramming in a Course in Computer Operating Systems

It would seem that a first course in operating systems - on a "dirty" machine - would be hardly the place for the subject of microprogramming to come up. Mr. Badger's reasons for making the study of microprogramming an integral part of such a course summarize the view of microprogramming as a medium of communication, especially pedagogical communication:

1) The breakdown of instructions into micro-instructions gives some idea of instruction commonality;

2) The principles of interpreter architecture are illustrated;

3) A non-hardware description of the machine is afforded by the microprogram;

4) An idea of what is time-consuming for the machine to do may be gained from putting together a microprogram;

5) The logical arbitrariness of the hardware-software "line" is illustrated, and the practical considerations that go into drawing this line are more fully appreciated after a microprogramming exercise.

H. C. Forsdick and Dr. R. Merwin

Microprogram Control Design and Simulation System

Dr. Merwin, speaking for Mr. Forsdick, described a system composed of a hardware description language, actual microcode language, and simulation language, useful as a pedagogical tool and design experimentation aid.

SECTION IV

4.0   SESSION III

"APPLYING MICROPROGRAMMING TECHNIQUES"

Chairman: Mr. J. D. Babcock
Allen-Babcock

(represented by P. R. DesJardins, Allen-Babcock)

The advent of read-only-store computer design
has suggested the ability to "extend" the use
of such a machine beyond that of a general-
purpose production tool. This session will
concentrate on projects in which the use of
micro-programmed machine resulted in an extended-
machine (as opposed to a limited-machine) design.

The main discussion will present specific
projects, including statistical reports on the
application (thus, why was it microprogrammed
in the first place), costs of the programming
effort, and the required training and back-
ground needed for the implementors. Some emphasis
will be directed toward anticipating future
needs in implementing microprogramming applications.
This information will be based on the experience
gained on the projects described.

— from the Workshop Bulletin

R. T. Borovec

<u>A Report on the Use of Microprogramming for the</u>
<u>Illiac III Image Processor</u>

The session opened with a presentation by R. T. Borovec on the
Illiac III Image Processor. Developed for high-energy physics
applications, it is now used also in biological and weather photo
analysis. The central control point or "pattern articulation unit
(PAU)" is microprogrammed; this unit was the focus for most of the
talk.

Interesting features of the Image Processor are a transfer
memory accessible either as 1024 48-bit words or 48 1024-bit words
and a 32 x 32 bit "iterative array" which may be used as an associative
memory. The organization may be depicted as follows:

———————▶ = Control Path

═══════▶ = Data Path
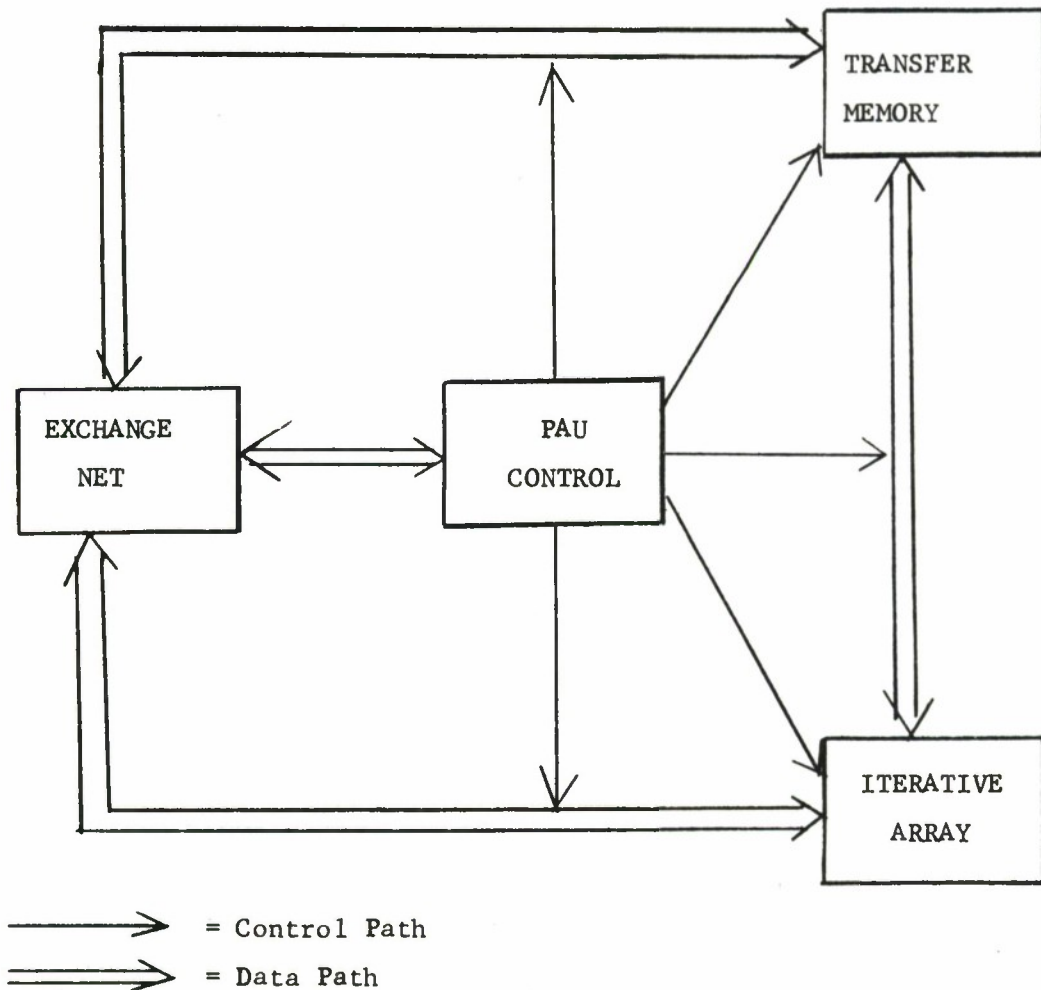
P. R. DesJardins

The Use of Microprogramming to Enhance Machine Performance of a Time-Sharing Programming System

Mr. DesJardins described an extension of twenty-two instructions to the 360/50 manufacturer-supplied set. The instructions described were in support of an interactive time-sharing system (RUSH). Those singled out as having the most payoff were variants of a chained list search. Other 'list' instructions, "reentrant utility" instructions, and genuine floating decimal instructions were also discussed. (The microcode is available from IBM as RPQ's.)

D. Boyle

Microprogramming of Data Logging and Data Reducing Equipment

Mr. Boyle described a simple processor capable of driving and
sequencing the steps of a physical experiment. The system could be
characterized as a sort of "immediate" process control, or a computer
with radars, A-D converters, and the like as components but largely
lacking in the usual things such as memory or arithmetic units.
Mr. Boyle himself said the system was simple and did not claim that a
sophisticated or advanced application of microprogramming was involved.
Some discussion was triggered by a remark from I. Flores to the effect
that the system was too simple to be interesting and not even micro-
programming - a view not generally supported by those present.

C. L. Mathis

Emulating the 7094 on the 360/85

Mr. Mathis of IBM described various approaches to the specific
problem of 7094 emulation on the 360/85, with a performance goal of
doubling the 7094 speed. Two approaches were discussed, (1) a "sub-
routine interpretive mode" involving a slight extension of the 360/85
instruction set for decoding 7094 instruction and executing a few of
the more frequently used ones, and (2) a "microprogram interpretive
mode" wherein sequences of 7094 instructions may be executed in micro-
code. Approach (2) required various ad hoc additions to the 360/85
hardware, for example, an inhibit of the instruction look-ahead.

21

B. Caruthers

Microprogramming to Support FORTRAN Functions

When an instruction set is implemented without "excess garbage" - from a FORTRAN point of view - the result is a faster running FORTRAN. This perhaps predictable consequence was documented by Mr. Caruthers in his discussion of such an instruction set. Speed was improved by the following factors:

≤ 2 in general computation;

5 to 7 in subscript calculation;

"hundreds" in computation of exponentials.

H. Lawson

Microprogrammed Higher Level Language Computers

Mr. Lawson opened with a set of definitions (with credits to Julien Green) that included: "Microprogramming is programming the interpreter". Mr. Lawson went on to present the case for microprogrammed truly higher-level languages. The reasons he advanced and the general properties he postulated for such a language seemed to imply something on the order of PL/I or even beyond. The benefits to be derived were stated to be as follows:

(1) Simplicity - fewer levels of language to cope with;

(2) Efficiency - "super" operations may be performed in high-speed control;

(3) Compatability - historically, more success has been realized with higher-level languages than with machine languages.

SECTION V

5.0 SESSION IV

"SELECTED SHORTS ON THE PAST AND FUTURE"

Chairman: Mr. A. Siegel
Decision Systems

This session will assess the extent and direction,
if any, of the influence of microprogramming
concepts upon the design and utilization role of
user-prepared microprograms, the relationship
between engineering and programming, the impli-
cations of writable control storage, and the
feasibility of standardizing micro-processors
and/or microprogramming languages.

Based upon their own experience, all partici-
pants will be encouraged to express their views
on the benefits or dangers of the more widespread
use of microprogramming. The discussion is
expected to culminate in meaningful conclusions
concerning the future of microprogramming and
its role as a lasting influence, or a passing fad,
in computer technology.

Since vigorous discussion is anticipated, partici-
pants who wish to ensure themselves of an oppor-
tunity to speak may do so by contacting the session
chairman prior to the meeting.

— from the Workshop Bulletin

Proceedings

Session IV was by design an open-floor session with only short, informally scheduled and informally delivered talks from the rostrum.

Dan Zatyko of General Electric lucidly illustrated the problems that arise when microprogramming is defined in terms of language level - e.g. as "implementing control logic" - because, of course, any level can be thought of as interpreting and executing a "language" at a higher-level. He referred to the level below microprogramming as "picoprogramming" - a term which hasn't caught on, at least not yet.*

Julien Green characterized micromachines by such things as their sparsity of registers and the treatment of main memory as an I/O device.

Still other definitions of microprogramming were advanced, some in terms of hardware ("regular" vs. "irregular" memories), and others in terms of characteristics such as parallelism. This writer offered the view that "micro" is, after all, a quantitative prefix and that precise definition was neither possible nor desirable - just as a line cannot be drawn between "big" and "little". This failed to settle the question, but at least the discussion moved on to other questions soon thereafter.

Bob Rosin of SUNY exhibited the architecture of a hypothetical computer he uses for pedagogical purposes and for which a simulator has been built.

C. Billings of Honeywell presented the case for a higher-level language to be compiled into microcode (analogously to FORTRAN and machine code). The arguments were the standard ones for higher-level languages. When he finished and asked for opinions, the consensus seemed to be that efficiency problems would make such languages impractical. Of course, this was the strongest argument against FORTRAN in its early days also, and time has shown the argument to be hollow. Nevertheless, the two cases are not quite equivalent and so the issue was not resolved.

---

*but see Briley, P. E., "Picoprogramming: A New Approach to Internal Computer Control", AFIPS 1965 Fall Joint Computer Conference, pp. 193-98; May-June 1966.

The topic of writable control stores prompted a vigorous floor debate on the advisability of allowing users to modify the control store - a "freedom vs. security" question, in this writer's opinion. Not surprisingly, manufacturers expressed a reluctance to offer this facility, envisioning a documentation and maintenance nightmare. One even expressed concern that users would "hang themselves" (perhaps with too much rope memory?). It seemed clear from the discussion that writable control stores are destined to come very much into demand, and that the technology governing their proper use will have to advance rapidly.

APPENDIX A

ATTENDEES

Andress, J. M.
UNIVAC DPD
2276 Highcrest Drive
Roseville, Minnesota   55113

Andrews, James H.
Control Data Corporation
4201 North Lexington
Arden Hills, Minnesota   55112

Badger, G. F.
Assistant Director
University of Pittsburgh
The Computer Center
800 Cathedral of Learning
Pittsburgh, Pennsylvania   15213

Bessel, D.
Office of Computing Services
Langmuir Laboratory
Cornell University
Ithica, New York

Billings, C.
Electronic Data Processing Division
HONEYWELL, Inc.
200 Smith Street
Waltham, Massachusetts   02154

Blair-Smith, H.
MIT Instrumentation Laboratory
75 Cambridge Parkway (MS35)
Cambridge, Massachusetts

Borovec, R. T.
Department of Computer Science
University of Illinois
Urbana, Illinois   61801

Boyle, D.
U. S. Department of Commerce
R/4404060
Environmental Science Services
Administration
Boulder, Colorado   80302

Carr, J. W.
Chairman
Computer and Information Sciences
The Moore School
University of Pennsylvania
Philadelphia, Pennsylvania

Caruthers, B. G.
INFOCOM
Wellesley, Massachusetts

Chu, Y.
University of Maryland
College Park, Maryland

Connors, T. L.
The MITRE Corporation
P. O. Box 208
Mail Stop D083
Bedford, Massachusetts   01730

27

Cook, R. W.
IE/MS Department
Northwestern University
Evanston, Illinois   60201

Curtis, R. G.
The MITRE Corporation
P. O. Box 208
Mail Stop D083
Bedford, Massachusetts   01730

Davis, R. D.
Scientific Control Corporation
14008 Distribution Way
Dallas, Texas   75234

DesJardins, P. R.
Allen-Babcock Computing, Inc.
Century City, Gateway East
1800 Avenue of the Stars
Los Angeles, California   90067

Douglas, J. R.
RCA
Riverton, New Jersey

Drum, R. O.
RCA
Camden, New Jersey

Ellis, R. A.
Computer Systems Laboratory
Washington University
724 South Euclid Avenue
St. Louis, Missouri   63110

Feder, M.
IBM Corporation
326 E. Montgomery Avenue
Rockville, Maryland   20851

Fidlow, D.
Bendix Corporation
Navigation and Contract Division
Teterboro, New Jersey

Flatt, H. P.
Washington Scientific Center
IBM Corporation
11141 Georgia Avenue
Wheaton, Maryland   20902

Flores, I.
Baruch College
City University of New York
17 Lexington Avenue
New York, New York   10010

Forsdick, H. C.
Yale College
1500 Yale Station
New Haven, Connecticut

Furman, A.
INTERDATA Corporation
2 Crescent Place
Oceanport, New Jersey

Galey, J. M.
IBM Corporation
SDD Laboratory - Department 55
Monterey and Cottle Roads
San Jose, California   95114

Goode, J. M.
RCA
P. O. Box 588
Burlington, Massachusetts   01801

Grasselli, A.
University of Pisa
Instituto di Elaborazione del
Informazione
Pisa, ITALY

Gray, H. J.
Professor of Electrical Engineering
The Moore School of Electrical
Engineering
University of Pennsylvania
Philadelphia, Pennsylvania   19104

Green, J.
Julien Green Associates
340 West State Street
Trenton, New Jersey   08618

Greenfield, M. N.
Electronic Data Processing Division
HONEYWELL, Inc.
200 Smith Street
Waltham, Massachusetts   02154

Greninger, L.
Department of Computer Science
University of Illinois
Urbana, Illinois   61801

Guglielmi, P.
Spaceborne Technology Branch
NASA Electronic Research Center
Cambridge, Massachusetts

Haight, R.
Bell Telephone Laboratories
Holmdale, New Jersey   07733

Heiss, S.
RCA
Palm Beach Gardens
Palm Beach, Florida   33403

Hellerman, L.
IBM Corporation
Systems Development Division
P. O. Box 390
Poughkeepsie, New York   12602

Hoernes, G. E.
IBM Corporation
Systems Development Division
P. O. Box 390
Poughkeepsie, New York   12602

Hoff, G.
Section Head
Central Processor Logic Design
HONEYWELL, Inc.
200 Smith Street
Mail Station #459
Waltham, Massachusetts

Hornbuckle, G. D.
MIT-Lincoln Lab
Lexington, Massachusetts

Huberman, B. J.
The MITRE Corporation
P. O. Box 208
Mail Stop D083
Bedford, Massachusetts   01730

Hurley, D. T.
Senior Data Systems Engineer
Aerospace Computer Engineering
General Electric Company
Defense Electronics Division
Utica, New York   13503

Husson, S.
IBM Systems Research Institute
United Nations Plaza
New York, New York

Kalin, R. B.
MIT-Lincoln Lab
Lexington, Massachusetts

Kearney, R.
Bell Telephone Laboratories
Holmdale, New Jersey   07733

Kelley, F.
RCA
P. O. Box 588
Burlington, Massachusetts   01801

Kennedy, J. R.
Computer Sciences Corporation
8300 South Whitesburg
Huntsville, Alabama   35802

Lanxner, J. M.
IBM Poughkeepsie Laboratory
Department C-02
Building 706-2
Poughkeepsie, New York

Lawson, H. W., Jr.
Polytechnic Institute of Brooklyn
333 Jay Street
Brooklyn, New York   11201

Layer, R.
Association for Computing Machinery
211 East 43rd Street
New York, New York

Mathis, C. L.
International Business Machines Corp.
Systems Development Division
Poughkeepsie, New York   12602

Matsuda, T. R.
AC Electronics
Defense Research Laboratories
6767 Hollister Avenue
Goleta, California   93017

McLean, J.
(EMIIF)
Rome Air Development Center
Griffiss Air Force Base
Rome, New York   13440

Merwin, R. E.
DOD
Arlington, Virginia

Miiller, H. S.
RCA
Princeton, New Jersey

Montano, P.
IBM Corporation
Department B92, Building 706
Poughkeepsie, New York   12603

Moreland, C.
INTERDATA Corporation
2 Crescent Place
Oceanport, New Jersey

Moros, D.
Philip Hankin, Inc.
Arlington, Massachusetts

Mui, M. T.
Electronic Data Processing Division
HONEYWELL, Inc.
200 Smith Street
Waltham, Massachusetts   02154

Nestle, E.
INTERDATA Corporation
2 Crescent Place
Oceanport, New Jersey

O'Brien, J. A.
Principal Engineer
Systems Integration Department
Raytheon Company
Space & Information Systems Division
Sudbury, Massachusetts   01778

Pandeff, E.
CII-IRIA Les Clayes
Sous Bois, FRANCE or
32 Chestnut Street
Princeton, New Jersey   08540

Patel, C. M.
Foxboro Company
Neponsit Avenue
Foxboro, Massachusetts   02035

Pearce, B.
Control Data Corporation
4201 North Lexington
Arden Hills, Minnesota   55112

Quigley, R. J.
Microprogramming and Simulation
Development
Radio Corporation of America
Electronic Data Processing
Camden, New Jersey   08101

Rakoczi, L. L.
Vice President
Standard Computer Corporation
1411 W. Olympic Boulevard
Los Angeles, California   90015

Ramamoorthy, C. V.
Department of Electrical Engineering
and Computer Sciences
The University of Texas
Austin, Texas   78712

Rath, K. G.
Standard Computing Corporation
New York, New York

Reiner, J.
General Electric Company
French Road
Utica, New York

Rosin, R. F.
SUNY at Buffalo
Buffalo, New York

Siegel, A.
Vice President
Decision Systems, Inc.
East 66 Midland Avenue
Paramus, New Jersey   07652

Smith, H.
Staff Member
Washington Scientific Center
IBM Corporation
11141 Georgia Avenue
Wheaton, Maryland   20902

Snyder, Allen D.
Department 899, Building 705
IBM Corporation
Poughkeepsie, New York

Spratt, J. J.
RCA Information Systems Division
Building 215-2
Camden, New Jersey   08101

Stabler, E.
Associate Professor of Electrical
Engineering
Syracuse University
206 Crestwood Drive
North Syracuse, New York

Stevens, W. Y.
IBM Corporation
Box 390
Poughkeepsie, New York   12602

Sullivan, J. E.
The MITRE Corporation
P. O. Box 208
Mail Stop D083
Bedford, Massachusetts   01730

Swanton, J. B.
The MITRE Corporation
16915 El Camino Real
Beta Building
Houston, Texas   77058

Tausner, M.
Decision Systems, Inc.
East 66 Midland Avenue
Paramus, New Jersey 07652

Thaler, H.
MIT-Lincoln Lab
75 Cambridge Parkway
Cambridge, Massachusetts

Tonik, A. B.
Consultant
Advanced Systems Design
UNIVAC
P. O. Box 8100
Philadelphia, Pennsylvania   19101

Tucker, A.
IE/MS Department
Northwestern University
Evanston, Illinois   60201

Vandling, G. C.
International Business Machines Corp.
Oswego, New York   13827

Vollbrecht, J. R.
Leader, Programming Planning
RCA
Building 2042
Cherry Hill, New Jersey

Wallis, D. E.
Center for Exploratory Studies
IBM Corporation
Federal Systems Division
1730 Cambridge Street
Cambridge, Massachusetts   02138

Wang, G. Y.
Computation Research Laboratory
Chief, Spaceborne Technology Branch
NASA Electronic Research Center
565 Technology Square
Cambridge, Massachusetts   02138

Ward, R. L.
Sanders Associates
Bedford, Massachusetts   01730

Wegbreit, B.
Computation Laboratory
Harvard University
Cambridge, Massachusetts

Weiss, D.
IBM Corporation
50 Timberline Drive
Poughkeepsie, New York   12603

Whinihan, M.
Computer Usage Company/
Information Research Association
307 Elliot Street
Newton Falls, Massachusetts

Wilkes, M. V.
University Mathematical Laboratory
Corn Exchange Street
Cambridge, ENGLAND

Zatyko, D.
General Electric
13430 No. Black Canyon
Phoenix, Arizona

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*	2a. REPORT SECURITY CLASSIFICATION
The MITRE Corporation, Bedford, Massachusetts	UNCLASSIFIED
	2b. GROUP

3. REPORT TITLE

THE FIRST ACM WORKSHOP ON MICROPROGRAMMING

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

N/A

5. AUTHOR(S) *(First name, middle initial, last name)*

Joseph E. Sullivan

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February 1969	37	1
8a. CONTRACT OR GRANT NO. F 19(628)-68-C-0365	9a. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO. 7120	ESD-TR-68-445	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	
d.	MTR-805	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
N/A	Directorate of Planning and Technology, Electronic Systems Division, Air Force Systems Command, USAF, L. G. Hanscom Field, Bedford, Massachusetts

13. ABSTRACT

The Workshop on Microprogramming, sponsored by the ACM with the cooperation of MITRE, took place at MITRE on October 7-8, 1968. This paper is primarily an historical account and summary of the workshop's business, with some commentary.

DD FORM 1 NOV 65 **1473**

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
MICROPROGRAMMING						