ESD-TR-70-122

MTR-1747

MICROPROGRAMMING PROJECT
—
YEAR END REPORT

Barbara J. Huberman

MAY 1970

Prepared for

DIRECTORATE OF PLANNING AND TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract F19(628)-68-C-0365

ESD-TR-70-122

MTR-1747

MICROPROGRAMMING PROJECT

—

YEAR END REPORT

Barbara J. Huberman

MAY 1970

Prepared for

DIRECTORATE OF PLANNING AND TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract F19(628)-68-C-0365

FOREWORD

This report has been prepared by The MITRE Corporation under Project 700A of Contract F19(628)-68-C-0365. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, L. G. Hanscom Field, Bedford, Massachusetts.

REVIEW AND APPROVAL

Publication of this technical report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

ANTHONY P. TRUNFIO
Technical Advisor
Development Engineering Division
Directorate of Planning & Technology

ABSTRACT

This document summarizes the activities of Project 7120, C^3 Computer System Organization, for Fiscal Year 1969. The project objectives included experimentation with microprogramming on an Interdata 3 computer installed at The MITRE Corporation.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

The title of this project is C^3 Computer Organization. This is an ESD/MITRE computer technology project, initiated last year to support ESD in the area of development and acquisition of large-scale computer systems.

One of the primary goals of a technology project should be to evaluate and relate new technology to application requirements, and that is certainly a goal of this project. One of the problems is that it is often costly in time and money, or not even feasible, to evaluate new hardware or software techniques, and system designs must be accepted on faith.

Therefore, a more specific goal of this project has been investigation of microprogramming as a technique for creating an environment in which experimental hardware and software can be modeled and evaluated. Figure 1 summarizes some of the expected benefits of the investigation of microprogramming.

Our first step in this project was the acquisition of hardware. In the preceding fiscal year, an Interdata 3 computer was purchased. This is a small and inexpensive computer, well suited to this project for two reasons:

1. It is easy to microprogram.

2. Interdata supports microprogramming by users with a microprogram assembler and a simulator.

Considerable effort has been spent this year in acquiring devices for the computer, including a disk and a card reader.

At the beginning of the year, none of the staff had any experience microprogramming. Therefore, the first thing we did was to write a microprogram. This microprogram, called CALLIOPE, supported the same instruction set supplied by Interdata. However, CALLIOPE achieved a gain in execution speed of twenty to thirty percent.

Encouraged by our success with CALLIOPE, we then undertook our major effort of the year. This was first to select an area for the experimental application of microprogramming, and then to design and implement the microprogram.

- EXPLOIT ABILITY TO SPECIALIZE COMPUTER

  e.g., to aid software production

- INVESTIGATE TRADEOFF FLEXIBILITY

  hardware - microprograms - software

- INFLUENCE SPECIFICATION OF FUTURE SYSTEMS

- INFLUENCE NEXT-GENERATION TECHNOLOGY

  Software for effective microprogram use

  Computer design for microprogramming

Figure 1. Expected Project Payoffs

One problem central to the development of all computer based systems has been the production of software - the costs have been too high and the quality of the product too low. For these reasons, the goal of the next microprogram developed at MITRE, the Venus Multiprogramming System, has been to improve the process of software production. Based on the experience at MITRE, both in developing data management systems and monitoring the development efforts of other Air Force contractors, the architecture of the Interdata 3 machine was changed in several significant ways. Among other features, Venus supports virtual memories and debugging aids.

The Venus system is composed of both microprograms and software. Considerable time was spent on design of the system in general. Then the microprogram was designed in detail and implemented, and work on software design was begun.

SECTION II

HARDWARE

A large amount of hardware had been added to the I-3. Each piece of hardware proved difficult to install, usually because of faulty parts or incorrect design on the part of the manufacturer. In most cases these problems have been ironed out over time. A brief description of each piece of hardware is given below.

The facility configuration is described in some detail in MTR-921, "The Microprogramming Facility".

FACILITY

In the early part of the year, the facility suffered from space congestion, stale air, and inconvenient operating conditions. In December the facility was moved to more spacious quarters.

In order to accommodate new equipment, the computer was transferred from the original single rack into a four-rack layout. The central processor and core memory, along with the console, teletype, card reader and printer, were operational in the new configuration within a day. The other equipment, except for the disc, was functioning soon thereafter.

MEMORY

Main memory was increased in December by 24K to a total of 40K. In June the final 24K of core memory arrived, bringing the total to the maximum 64K.

CLOCK

The clock was delivered in March and installed successfully.

SELECTOR CHANNEL AND DISC

The selector channel was first delivered in December. The disc, along with a new selector channel, arrived at the end of January and appeared at first to be working well. However, Interdata discovered and reported an error in the selector channel. In May, Interdata swapped the selector channel with one of their own, in which this problem was fixed; the MITRE selector channel was returned for repairs.

Then the disc stopped working, apparently due to head damage. Interdata had already been having a great deal of trouble with this particular model disc and had promised us a replacement when the reliability problems had been solved. In August the new disc was delivered and installed successfully.

CARD READER

In September, the first of three successive card readers was delivered. This first card reader never really worked very well. By December the second reader had been installed. This, too, failed to operate well. Finally, the last card reader arrived in April and is generally working well.

ASR-33 TELETYPES

Both ASR-33 teletypes arrived damaged, a week late, and were returned immediately for repairs. Shortly thereafter, they were redelivered. Installation was delayed by the discovery that the interface was wired for connection to an I-4 instead of an I-3. After correction of this problem, the teletypes were successfully installed.

KENNEDY TAPE RECORDER

The Kennedy tape recorder was shipped to Electronic Marketing Associates in Kensington, Maryland, in September to have an interface constructed. Along with its interface, it arrived back in December. Both were inoperable when they arrived. After acquiring and installing some parts from Interdata, the Kennedy tape unit is now working well.

ARDS

The ARDS arrived in early November. It ran at teletype speed off a teletype interface through a small conversion box built locally. Existing programs for the teletype ran correctly with the ARDS, including both the MI-3 assembler and the interactive FORTRAN.

The high speed 202-C2 interface originally scheduled for March was first delayed to April and finally arrived in May. Although it was apparently successfully installed, programs for the ARDS have not yet been checked out using the new interface. Since May, Interdata has reported that they have found an error in the ARDS interface and will supply us with the correction as soon as they have one.

MOTOROLA PRINTER

The interface, printer and translator arrived in early May.

SECTION III

MICROPROGRAMS

During this year three microprograms were prepared. Two of them use a single read-only memory (ROM), while the third uses two. One read-only memory has room for 1024 microinstructions (at 16 bits per instruction).

CALLIOPE

CALLIOPE was the project's first attempt at microprogramming. As such, it was primarily a training exercise. The goal was to illustrate how good design and coding could result in more efficient execution of instructions. The instructions supported were the same as those supplied by Interdata. A new design of the microprogram was made. It required only one half of the ROM (512 instructions) to support all instructions. The remainder of the ROM was used to implement an I/O channel, a program-loading code (BOOTS) and a completely revised handling of the general purpose display panel.

In October testing of the CALLIOPE ROM under the simulator was completed. A tape was sent to Interdata to be wired.

The development of CALLIOPE, in essence a new computer using the same hardware, brought up questions of copyright and patentability. A form letter was written in which Interdata, when wiring the control memory, agrees not to copy the microprogram or sell it to the government.

The CALLIOPE ROM arrived in early December. Several bugs were found in the I/O channel and the display section. These were corrected by hand wiring patches to the ROM, a delicate and tedious business with no room for error. The wiring was done with 100% accuracy in about six hours. The bugs in CALLIOPE arose from logical errors and misconceptions about the nature of the I-3 which were not detected by the Interdata-supplied simulator.

Finally, tests were made to compare CALLIOPE with the microprogram supplied by Interdata. The experiment proved to be very successful; for example, CALLIOPE ran a FORTRAN problem 29% faster than the standard I-3. Other results are shown in Figure 2.

	COMPUTER AS DELIVERED	WITH REVISED MICROPROGRAM
ADD TIME	42.9 μsec.	27.5 μsec.
READ-ONLY MEMORY USED FOR INSTRUCTION PROCESSING	550 registers	350 registers
NEW INSTRUCTIONS	--	3
MAXIMUM SPEED I/O DEVICE	891 char/sec	5958 char/sec

Figure 2. Comparison of CALLIOPE Microprogram with Standard Interdata

POLYHYMNIA

POLYHYMNIA is a microprogram which operates microprograms stored in main memory simulating (or emulating) the operation of the Interdata 3 micromachine. The errors in the CALLIOPE ROM (discussed above) would have been detected by POLYHYMNIA but were ignored by the Interdata software simulator.

By December POLYHYMNIA existed as a deck of cards. Because of lack of manpower, further effort has not been expended on this aspect of the project.

VENUS

The major task of the project this year has been the design of the Venus multiprogramming system. Our primary goal in designing this system was to ease the problem of program production. Experience at MITRE has indicated to us that major problems arise in the areas of storage allocation, procedure interface, and debugging. Venus is designed to provide solutions to these problems. In addition, Venus is a multiprogramming system since this makes more efficient use of the computer. A brief description of the major features of Venus follows.

Storage Allocation

Storage allocation is handled automatically in Venus by providing users with named virtual memories called streams. Streams normally reside on a disc and are divided into pages of fixed length. The microprogram automatically maps stream addresses into core addresses. In the process of making this mapping, it may discover that the desired stream page is not in core. In this case, it starts a software program to fetch the page from the disc. The software routine then returns to the microprogram by means of a special instruction.

Procedure Interface

The basic problem about procedure interface is that each user tends to define a different interface. This leads to difficulties when programs are being checked out. Therefore in Venus a standard interface, in the form of CALL and RETURN instructions, is supplied and programmers must use it.

Debugging Aids

Debugging can be defined as finding out what your program is doing and then doing something about it. Venus provides interrupts which allow the user to find out what his program is doing. Software enables these interrupts; the microprogram or software turns them on. When an interrupt is both on and enabled, a software routine is started to allow the user to do something about the action of his program. Examples of interrupts turned on by the microprogram are: every instruction, CALL instruction, and stack overflow/underflow.

Multiprogramming

The Venus system supports up to 16 processes. The microprogram decides which of the competing processes is allowed to use the processor. Provision has been made for priorities and time slicing.

There are two special operations, first proposed by Dijkstra, for synchronizing sequential processes and the sharing of resources (data or devices). To the best of our knowledge, these operations had never before been implemented in hardware or microcode. Because a body of theory and application has already been developed for them and their usefulness demonstrated, they were included in Venus.

The design of the multiprogramming system was simplified by providing a multiplex channel. Although it is not unique to have a multiplex channel, one major difference in the Venus multiplex channel is that it does not interrupt. Instead it makes use of the synchronization primitives mentioned above to 'wake up' jobs which have indicated that they are waiting for I/O.

Work on the overall design of Venus continued for several months. By April, we felt we were ready to design and write the microprogram. This task was made simpler because we were able to use a large part of the CALLIOPE microprogram. The instruction set supported by Venus contains almost all standard I-3 instructions, in addition to many new instructions.

Checkout of the Venus microprogram under the simulator was completed in early July and the microprogram was sent to Interdata to be wired. The ROM was returned in August. Testing of Venus was then begun and several bugs detected.

SECTION IV

SOFTWARE

During the year most of the emphasis was placed on the acquisition of hardware and the development of firmware. A fairly large number of small software programs were written, usually for the purpose of smoothing facility operations. The programs described below represent more important efforts.

MI-3

Numerous changes to the MITRE-developed assembler were made.

1. Two new kinds of symbols known as Local and Global were added. Both kinds of symbols are used in a more global fashion than the symbols originally offered by MI-3.

2. An SOD pseudo-op was added to control the printing of the listing and its destination.

3. An SID pseudo-op was added to allow specification of a symbolic input device.

4. A new pseudo-op, OPD (Operation Definition), was added to make it easier to incorporate new instruction codes. This pseudo-op has proved extremely useful, because it permitted us to convert MI-3 to produce instructions for Venus. A new instruction table was defined for MI-3 using the OPD's; then a few changes were made in the MI-3 deck and it was assembled by itself. The result was an MI-3 for Venus.

OBJECT DECKS

Since we lack a card punch on the I-3, we have developed a method for punching object decks on the IBM 360 via the Phoenix computer.

An I-3 program called PUSS creates a paper tape which corresponds to a set of specified core locations. This tape is suitable for direct loading by the Bootstrap loader built into the CALLIOPE and VENUS ROMs. It can also be taken to the Phoenix computer where it can be converted into a magnetic tape.

This tape is then used as input to a program called BOOTRAN, written in PL/1 on the IBM 360. BOOTRAN punches a deck of cards which is directly readable by the Bootstrap loader. A large saving in daily startup time is thus achieved.

This method is fairly complicated and has not proved completely satisfactory. Including time necessary to rerun because of operator and other undetermined errors, the time required to convert a program from a paper tape to a working object deck is about two weeks. Therefore, a new approach is currently being investigated to input a paper tape from the I-3 directly into CPS on the IBM 360. This will build a file processable by batch PL/1.

VENUS SOFTWARE

As soon as initial testing of the microprogram was complete, work was begun on software for Venus. The greatest need existed for the program which performs paging between core and the disc. This program is closely connected with two other programs for creating and deleting virtual memories. These three programs were designed and coded by the end of the fiscal year.

SECTION V

MICROPROGRAMMING WORKSHOP

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.

SECTION VI

OTHER ACTIVITIES

During September a Software Technology Seminar on Microprogramming was held for department personnel. Also a meeting to report on the ACM convention was presented to division personnel.

In October a briefing on microprogramming was prepared and delivered to the Air Force Science Advisory Board.

Throughout the year a member of the staff has attended Share meetings. In addition to attending general sessions, he chairs a subcommittee on programming languages of the GIS committee.

An advanced development program, Computer Systems Organization, has been proposed for fiscal '70-'71 in a paper prepared and delivered to ESL and DOL.

In February a briefing on a proposed technical need in scheduling dependent tasks was prepared and presented to the DOL coupling meeting between MITRE and RADC. This briefing was also presented to the management of the division.

In March Professor E. W. Dijkstra of the University of Eindhoven in The Netherlands came to speak with us about the design of the system described in his paper, "The Structure of the 'THE'-Multiprogramming System"*. In April a member of the staff presented a department seminar reviewing this paper.

In May a member of the staff attended the International Joint Conference on Artificial Intelligence.

In June video tapes of a lecture series given by Professor Dijkstra at MIT were presented to the MITRE technical community.

Two films acquired from Stanford representing progress reports on the Artificial Intelligence Group with emphasis on the robot arm were presented.

In August a member of the group attended a SIGPLAN Language Definition Symposium.

---

* Communications of the ACM, May 1968

The abstract of a paper co-authored by Dr. Barbara J. Huberman and Robert G. Curtis, entitled "Venus Multiprogramming System", was submitted for presentation at the Second Annual Microprogramming Workshop at Phoenix, Arizona, on October 13 and 14.

REFERENCES

1. J. E. Sullivan, "The First ACM Workshop on Microprogramming," ESD-TR-68-445, The MITRE Corporation, Bedford, Massachusetts, February 1969.

2. R. W. Cornelli, "Microprogramming Facility," ESD-TR-69-358, The MITRE Corporation, Bedford, Massachusetts, November 1969.

Security Classification

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)* The MITRE Corporation Bedford, Massachusetts	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

MICROPROGRAMMING PROJECT — YEAR END REPORT

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

N/A

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

Barbara J. Huberman

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May 1970	22	2

8a. CONTRACT OR GRANT NO. F19(628)68-C-0365 b. PROJECT NO. 700A c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) ESD-TR-70-122 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* MTR-1747

10. DISTRIBUTION STATEMENT

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

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

13. ABSTRACT

This document summarizes the activities of Project 7120, C^3 Computer System Organization, for Fiscal Year 1969. The project objectives included experimentation with microprogramming on an Interdata 3 computer installed at The MITRE Corporation.

DD FORM 1 NOV 65 1473

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
MICROPROGRAMMING						
MULTIPROGRAMMING						
VIRTUAL MEMORIES						