SIMULATION OF THE MICRO-PROGRAM OF THE S-MACHINE

by 4589

JOE MING-HWA TIAO

B. A., Toyo University, Tokyo, Japan, 1964

M. A., Aoyama University, Tokyo, Japan, 1967

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Statistics and Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1970

K. E. Kemp
Major Professor

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
	Objectives of the Study	3
	Description of the Program	4
II.	MICROPROGRAMMING	5
III.	IMPLEMENTATION	7
	S-machine	7
	Micro-program	15
	Simulator	24
IV.	SUMMARY	27
V.	BIBLIOGRAPHY	28
VI.	APPENDICES	
	Appendix A. Simulator program	29
	Appendix B. S-machine program	47
	Appendix C. Micro-program	56
	Appendix D. Tracing	70

LIST OF TABLES

TABLE		PAGE
I.	ONE-address instruction for S-machine	11
II.	ZERO-address instruction for S-machine	12
III.	ZERO-address instruction for micro-program	20
IV.	FOUR-address instruction for micro-program	20
V.	Memory control for micro instruction code	21
VI.	Input bus 1 for micro instruction code	21
VII.	Input bus 2 for micro instruction code	22
VIII.	Output bus for micro instruction code	22
IX.	Function for micro instruction code	23
X.	Test conditions for micro instruction code	23

ACKNOWLEDGEMENTS

I wish to take this time to express sincere appreciation to Professor C. Walker, Major Instructor, for his guidance and suggestions during the preparation of the manuscript, and for providing his macro instructions to trace the contents of the program.

Special acknowledgement is due Dr. K. E. Kemp for his guidance and helpful suggestions throughout the program.

Also, I express my appreciation to all the members of the department of statistics and computer science and the computing center who most willingly gave assistance and advice during the course of this program.

Finally, I wish to thank my wife, Inger Ying-ying, for her encouragement and patience during the past two years.

CHAPTER I

INTRODUCTION

As computer industries grow in both quantity and specificity, a great number of special purpose languages have been created to fill individual needs. Ironically, not until the advent of a language known as S-machine language had any attention been paid to stack manipulation.

The compound meanings of stack and simulation are represented by the "S" in the name. S-machine instructions are written using zero-address and one-address instructions, creating operations and language characteristics that are extremely functional and easily assimilated.

The term simulation refers to imitation of one system by another, in this case one computer by another.[1] The entire patterned basis of computers in general, results in their self simulation being an extremely logical process. Whole machines or finite systems can be modeled. Many problems are difficult to solve analytically, but adequate criteria for success can be deduced from trial-and-error processes in which the model is dynamically studied.

The stack is a linear list for which all insertions and deletions are made at one end of the list.[2] A stack operation operates by using the top of two registers, putting the results in the second level and discarding the top level, leaving the results as the new top level.

---

[1]Gear, C. Willian, Computer Organization and Programming (New York: McGraw-Hill Book Co., 1969), PP 172.

[2]Knuth, E. Donald, The Art of Computer Programming (Addison- Wesley Publishing Co., 1969), I, PP 235 - 238.

Any program which is used in order to accomplish the execution of a single instruction in the user machine is called a micro-program.[3] A micro-program can be written in any way that provides a unique representation of each micro-instruction. The use of the microprogramming technique allows one to modify the actions of any and/or all of the S-machine instructions. It also provides a means to implement additional instructions.

An interpreter is a program that examines each statement or instruction in the language and causes the desired action to be taken immediately.[4] The main reasons for using an interpreter, or interpretation, are the following:

1) Interpreters are easier to write than compilers.
2) Interpreters are smaller and more compact than are comparable translators.
3) Interpreters are more informative during diagnostic stages of program running than compilers.

When a program is run on a computer often, run-time or object code efficiency is important, but during program debugging the effectiveness of error tracing is much more important.

A trace is often used to help in debugging since it prints out the contents of certain registers or memory locations everytime a branch is executed in the program. Thus when something goes wrong it is easy to follow the course of events leading up to the problem if a trace has been used.

[3] Wilkes, M. V., The Growth of Interest in Microprogramming (Comp. Surveys, Sept. 1969), I, No. 3, PP 139.

[4] Knuth, E. Donald, The Art of Computer Programming (Addison-Wesley Publishing Co., 1969), I, PP 197-198.

When the machine language of one computer is used on an other type machine by using an interpreting routine, the interpreter is often called a simulator. In other words, a simulator: (1) models all internal registers and data paths, (2) generates the sequences of operations that are identical to the control signals in the machine it is simulating.

One might ask, what is the difference between a computer simulator and an interpreter of a machine language? A simulator is a more involved than an interpreter. If the only desire is to interpret a language, it is of no importance to the user as to how this is done. However, a simulator should mimic the machine it is simulating as closely as possible in order to derive performance measurements.

The purpose of this report is to introduce a portion of the S-machine machine language and to demonstrate how microprogramming techniques can be used to simulate a computer system.

I. OBJECTIVES OF THE STUDY

The objectives of this study were:

1) To implement and test a computer program to simulate the S-machine.
2) To use the interpretation of the micro-program to simulate the execution of the S-machine instructions.
3) To write the simulator which acted as an interpreter to execute the micro-program, where the micro-program described the action of the S-machine instructions.
4) To write and test programs in the assembler language of the S-machine which has been implemented by using the macro facilities of the OS/360 Assembler Language.

II. DESCRIPTION OF THE PROGRAM

The simulation of the S-machine language was accomplished by means of three separate programs: the simulator program, the S-machine program and the micro-program. The simulator program, which is shown in appendix A, was written using the IBM OS/360 assembler language. The S-machine program presented in appendix B, was written using zero-address and one-address instructions. Finally appendix C contains the microprogram which was written by means of one-address and four-address micro-instructions. The macro facilities of the IBM OS/360 assembler were utilized extensively in writing all three segments.

CHAPTER II

MICROPROGRAMMING

Microprogramming is used to implement the control portion of a computer by effecting a number of register-to-register transfers of information. Some of these transfers take place directly and some through an adder or other logical circuitry in the process of carrying out the execution of a single machine instruction. Each of these steps can be thought of as the execution of an instruction for some machine. The steps needed to execute a single instruction in the user machine can be thought of as a program, usually called a micro-program. A micro-program can also be used for other necessary operations which are in a sense transparent to the programmer. For instance, fetching the next instruction or computing an effective addresses. In brief, microprogramming is a means for implementing the control function of a computer.

There are at least two approaches to the micro-program control of computers.[5] One called "vertical or sequential microprogramming", relies on the more traditional concept of programming in which an instruction contains an operation code, perhaps with some secondary modifiers, and one or more address fields. This was the method used in the simulation program presented in this report. The other approach, called "horizontal microprogramming", is to conceive of the micro-instructions as control words whose individual bits are used to select specific data paths within the machine. In this case there are no addresses other than implicitly specified by the bits of the control words.

---

[5] Rosin, R. F., Contemporary Concepts of Microprogramming and Emulation (Comp. Surveys, Dec. 1969), I, No. 4, PP 202-203.

In either case the micro-instructions are generally held in a memory whose cycle time (the time to fetch and execute an instruction) is usually faster than main memory. The majority of such systems use a nondestructive read-only memory (ROM) for micro-programs for reasons of speed, economy and memory protection.

The reasons for using microprogramming techniques in designing a particular computer system vary from situation to situation.

There are several interesting features why microprogramming is used, such as:

1. Microprogramming allows the redesign of an instruction set and offers the opportunity to provide instructions and features to the system.

2. New OP codes are easily implemented.

3. It is possible to structure a new system and simulate it with much less effort than is often required in the typical simulation scheme. (In the simulation program presented in this report there were one-hundred-twenty micro instructions needed to simulate twenty-six S-machine instructions.)

4. Microporgramming provides an economical means of having a large instruction set in a small machine.

5. Microporgramming is often easier to check out during the design state because the micro control can be easily simulated.

The following are some of the limitations of difficulties involved microprogramming.

1. Execution speed is slow due to inefficiency. For instance in the S-machine language, for one place shifts, there are at least six step actions to be taken.

2. The concept of machine interrupts is not easy to handle with microprogramming.

CHAPTER III

IMPLEMENTATION

As indicated earlier, the simulation of the S-machine was accomplished by means of a computer program divided into three separate segments; the S-memory segment, the micro-program segment, and the simulator segment. These segments will now be discussed separately.

I. S-Machine

The organization of the S-machine in the simulation program is classified into two parts, the memory and the control processing units (CPU for short). The S-machine processes information from the stack memory, through the CPU, and stores the results back into the stack memory.

The memory unit of the S-machine is a word. The length of the word is thirty two bits (four 8-bit-bytes). Each bit of a thirty-two bit word can be a 0 or a 1 independently, so that a word of the stack memory can store any one of 2^{32} different states.

The S-machine has byte addressing, the smallest addressable unit of the S-machine is one 8-bit-byte in memory. The bottom 2 bits of an address are ignored in memory READ and WRITE operations.

There are two registers which serve for READ and WRITE operations in the S-machine. They are the memory data register (MDR) and the memory address register (MAR). The MDR can store one word while the MAR can store enough bits to represent any address in the stack storage.

If the S-machine is instructed to read a number from memory (READ) the contents of the memory location specified by the address in the MAR are copied into the MDR. If the S-machine is instructed to store the information

into memory (WRITE) the contents of the MDR are copied into, and replace the contents of, the memory location specified by the address in the MAR.

There are six registers in the control processing unit (CPU) of the S-machine. Each register contains thirty-two bits except for the instruction register (IR) which is a five-bit register. The A and B registers are two working registers. The A register contains the data from the top of the stack and the B register holds the effective address which is to be used by the instruction. The X register is the single index register. The CC register is the control counter which is increased by four each time a micro-instruction is executed. The IR resister is the instruction register and is used to extract five bits from the bottom of the branch address bits in a micro word, and the result is transferred to the micro control counter. The STK register is used to hold the address of the top of the stack.

There are three data paths connecting the registers in the S-machine. These are called buses, two for input to the logical unit, and one for output. The logical unit is used to form logical and arithmetic combinations of the input buses. The latch, a thirty-two bit register, is included in the logical unit and serves to retain the output of the logical unit after the inputs are turned off. This way it is possible to return a result to a register which contained an input operand. Figure 1 shows the data paths for S-machine.

Gating controls the movement of data from one register to another. If there are two registers to be connected, the gate controls the information flow. When the gate is open a copy of the information in the first register is placed in second register. The S-machine is controlled

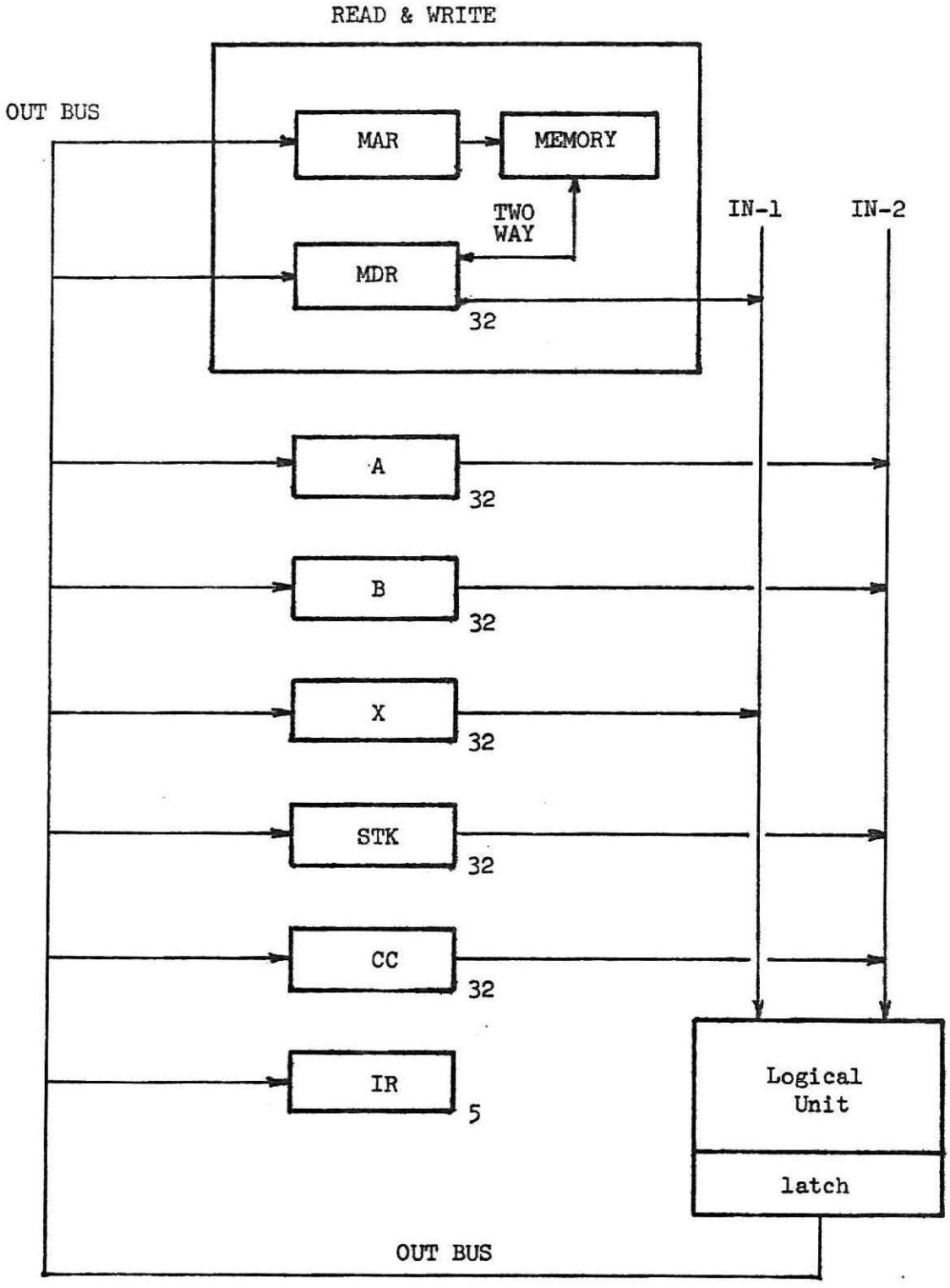

Figure 1. S-machine Data Paths

by such a sequence of gate signals and these signals are generated by the control section of the computer.

Each S-machine instruction occupies a thirty-two bit word, which means its location address is a multiple of four bytes. The formats of the instructions are shown in Figure 2. There are two types of instruction formats, one-address and zero-address instructions. They are distinguished by the first bit of the instruction code, that is, bit two of the thirty-two bit word. The code is zero for one-address instructions and one for zero-address instruction. The code for one-address instruction is between 000000 and 011111, whereas the code for zero-address instructions is from 100000 to 100000 to 111111 because of the bit significance of the top bit. The code of one-address instructions and zero-address instructions in the program are shown in Table I and Table II.

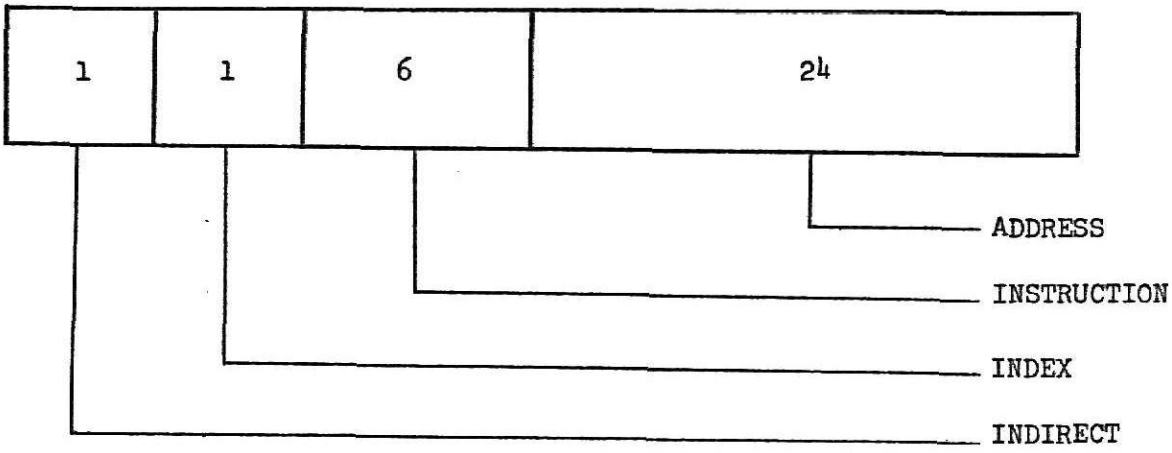

Figure 2. S-Instruction Format

TABLE I

ONE-ADDRESS INSTRUCTIONS

Code	Bit Position	Mnemonic	Action
0	000000	LOAD	Load stack from memory location.
1	000001	LDI	Load immediate. Loads top of stack with address.
2	000010	STORE	Stores top of stack in memory.
3	000011	TRA	Transfers control to specified location.
4	000100	TPL	Transfers control if top of stack is positive or zero.
5	000101	TMI	Transfers control if top of stack is negative (non-zero).
6	000110	TZE	Transfers control if top of stack is zero.
7	000111	TNZ	Transfers control if top of stack is non-zero.
8	001000	ENTER	Enter a subroutine by placing CC on top of the stack and transferring control.
9	001001	LDX	Load index with contents of memory location.
10	001010	LDXI	Load index immediate, that is, with the address in this instruction.
11	001011	LOOP	Increment index register by 4 and transfer control if it is non-zero.
12	001100	LS	Left shift N places, where N is the address.
13	001101	RS	Right shift N places.

TABLE II

ZERO-ADDRESS INSTRUCTIONS

Code	Bit Position	Mnemonic	Action
32	100000	ADD	Add top two levels of stack.
33	100001	SUB	Subtract top two levels of stack.
34	100010	AND	AND top two levels of stack.
35	100011	OR	OR top two levels of stack.
36	100100	EOR	Exclusive OR top two levels of stack.
37	100101	NOT	Complement top level of stack.
38	100110	XTS	Index to stack. Puts contents of index register on top of stack.
39	100111	STX	Stack to index. Removes top of stack and places it in index register.
40	101000	ADX	Adds top of stack to index.
41	101001	SBX	Subtracts top of stack from index.
42	101010	RET	Transfers to address contained in top of stack.
43	101011	POP	Discard the top level of the stack.
44	101100	STOP	Stop run.

In the one address S-machine instruction format the first bit and second bit represent indirect and index bits. If the index bit is 1 the address of the one-address instructions is formed by adding the contents of the X register to the twenty-four-bit address in the instruction, and then performing indirect addressing with the resulting address if the indirect bit is also a 1. Indirect addressing is related to indexing. If this bit is on then a new thirty-two-bit word is fetched from emmory and it is also checked for indexing and indirect addressing.

The contents of the accumulator stack are actually stored in memory starting from the maximum available address and working down. The current top level of the stack is kept in the memory location contained in the STK register. If an item is to be pushed into the stack then STK is decreased by 4 and sent to the MAR as the address of the new top level. The bottom level is always in location $2^{12} - 4$. For instance, a stack machine provides an instruction SUB which combines the top two levels of the stack by subtraction and stores the result back into the second level, subsequently discarding the top level. Prior to the execution of the SUB instruction, the STK register contains the address of the top of the stack. The second level is in location numbered 4 larger as shown in Figure 3. In order to execute the SUB instruction the address in STK must be sent to the MAR register. A READ can be used to access the top level of the stack and place it in the MDR register. This execution should be as the following sequence;

```
MAR <—— STK. READ
MDR <—— STORAGE (MAR)
A   <—— MDR
STK <—— STK - 4
MAR <—— STK, READ
MDR <—— STORAGE (MAR)
MDR <—— MDR - A
MAR <—— STK, WRITE
STORAGE (MAR) <—— MDR
```

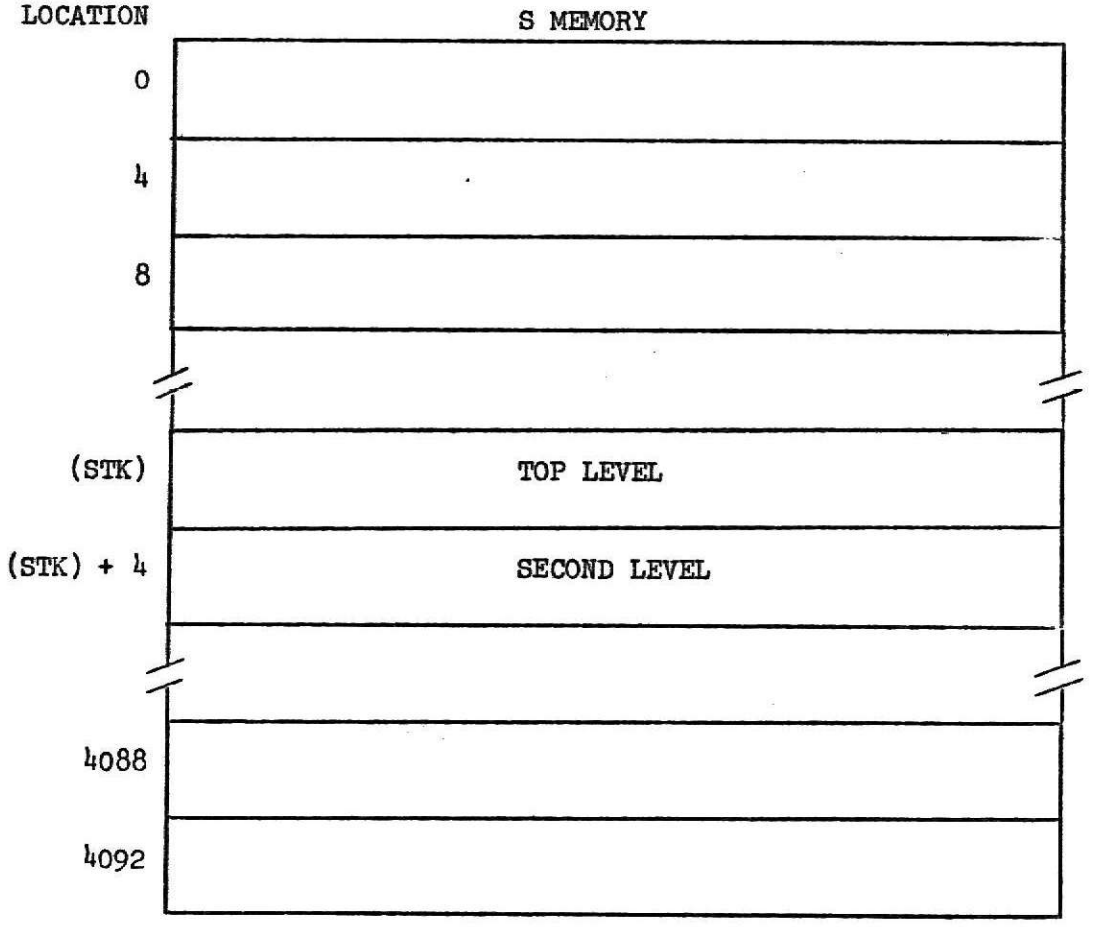

Figure 3. Stack Memory

Having fetched the next instruction from emmory into the MDR it is necessary to decode the instruction bits in order to determine their type. If it is a one-address instruction it is also necessary to form the effective address in the B register. Since this is a requirement for all one-address instructions it should be done before the remaining 5 bits of instruction code are examined. If the first bit of the instruction code is a 1 a zero-address instruction was fetched. If it is a 0 a one-address instruction was fetched. The decoding step is written as shown in Figure 4.

The S-machine consists of the S-instruction set. There are two types instructions; one-address and zero-address instructions. The code for one-address instructions is between zero and thirty one, whereas the code for zero-address instructions is between thirty two and sixty three. Each S instruction occupies a thirty-two bit word so that its location address is always a multiple of four bytes. There are eight registers used in the S-machine data flow. The purpose of the program written in the S-machine language was to test for the proper execution of S-machine instructions by simulation.

II. MICRO-PROGRAM

The second source module of the simulator is the micro-program module, which was written by means of one-address and four-address micro-instructions. The Macro assembler was used to assemble the micro-program. The one-address and four-address micro-instructions are shown in Tables III and IV, respectively. In these tables I1 represents the input bus 1 (IN-1), I2 the input bus 2 (IN-2), OUT the output bus and Z is either P,R, or W, meaning process only, process followed by READ, or process followed by WRITE, respectively.

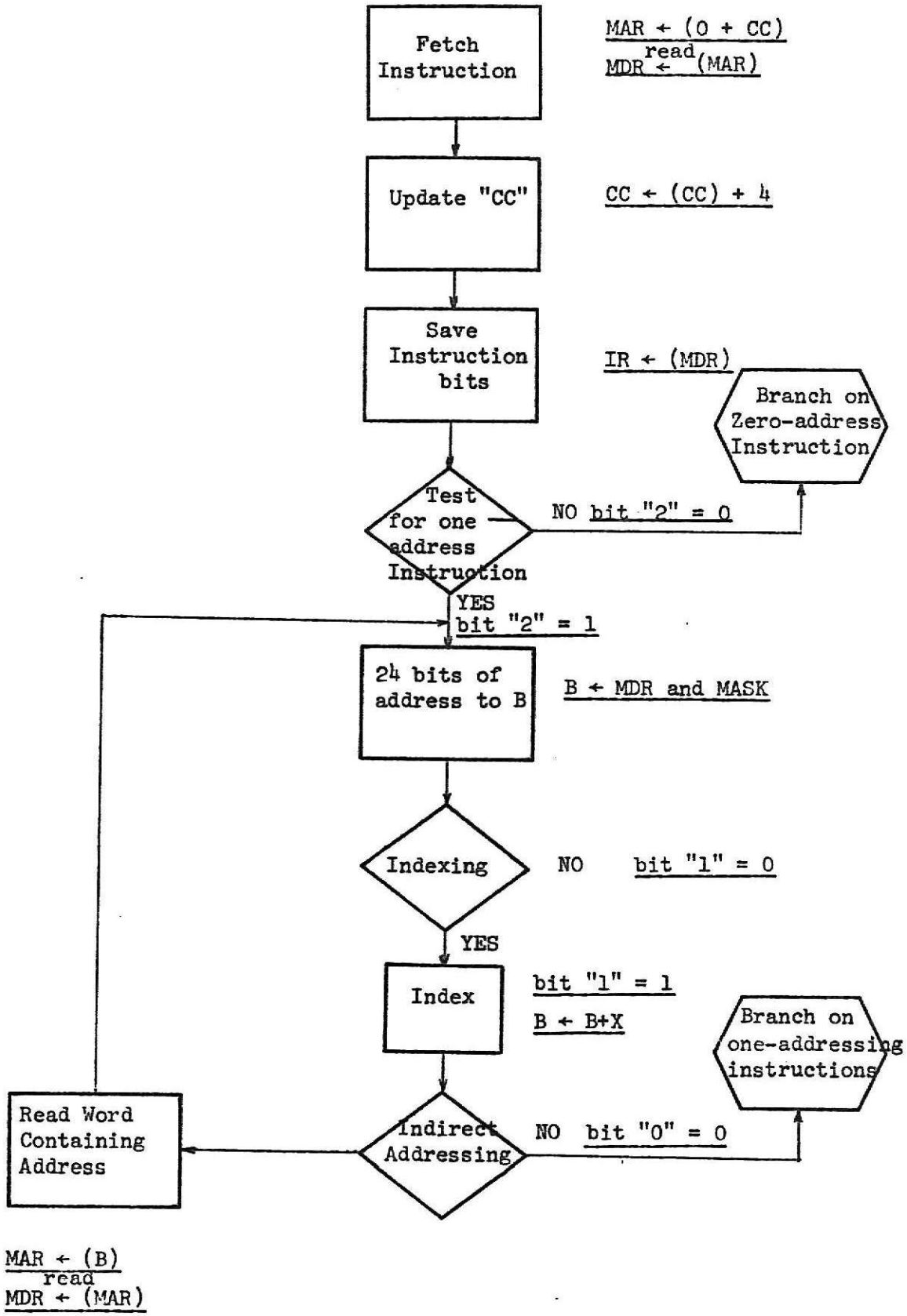

Figure 4. Flow-chart for decoding instruction

The micro machine is shown in Figure 4. It contains a memory in which the micro-program is stored. Each step in the micro-program is stored as a word in the micro memory. The micro memory can be read but not written.

The micro control counter serves as an adder for the micro machine. Each time when the simulator executes a micro instruction the micro control counter is incremented to the next micro-instruction.

The micro address register is only used for accessing micro instructions. It always contains the address of the next micro-program step to be executed. Each micro-instruction is fetched from the micro memory in turn and decoded from the memory data register to determine which gates to open or which conditions to test. If conditions are tested and are true, then a new address is gated from the data register to the address register in the micro machine.

There are two different types of micro-instruction formats, one for operation steps and one for testing steps. The way in which each of these is stored in the sixteen-bit-wide micro memory is illustrated in Figure 5 and Figure 6. If the left bit is a 0 then the word represents an operation. Each group of bits, called a control group, contains a coded representation of an action. The codes used for each of these control groups are shown in Tables V through IX. If, on the other hand, the top bit is a 1, the sixteen-bit word represents a test action. In this case the micro machine examines the indicated bit or bits in the S-machine data flow and performs a branch if the condition is met. This branch is effected by placing the bottom twelve bits of the micro word into the micro control counter. The conditions tested for each of the eight code values of bits 1 through 8 of the micro word, are shown in Table X.

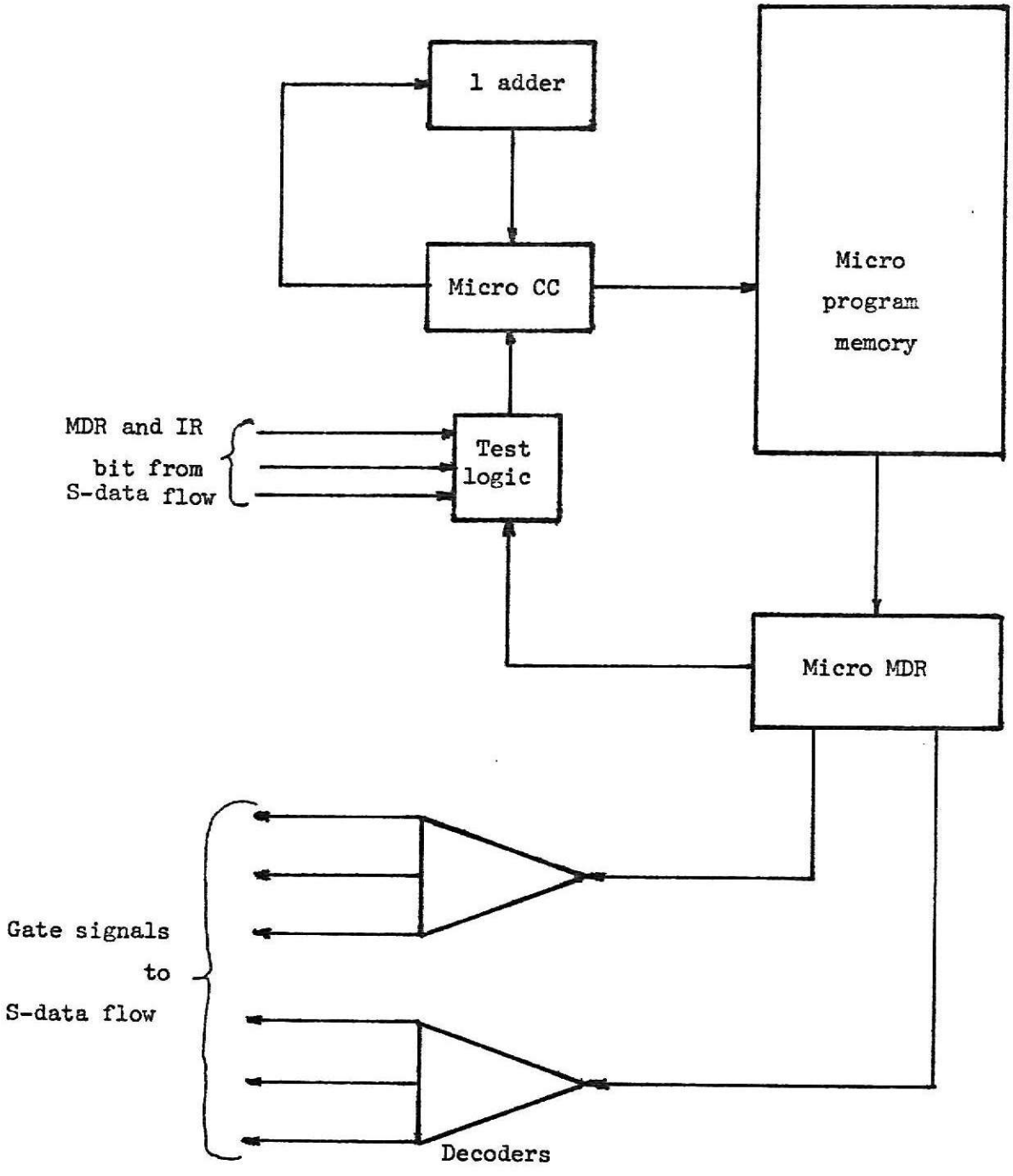

Figure 4. Micro machine

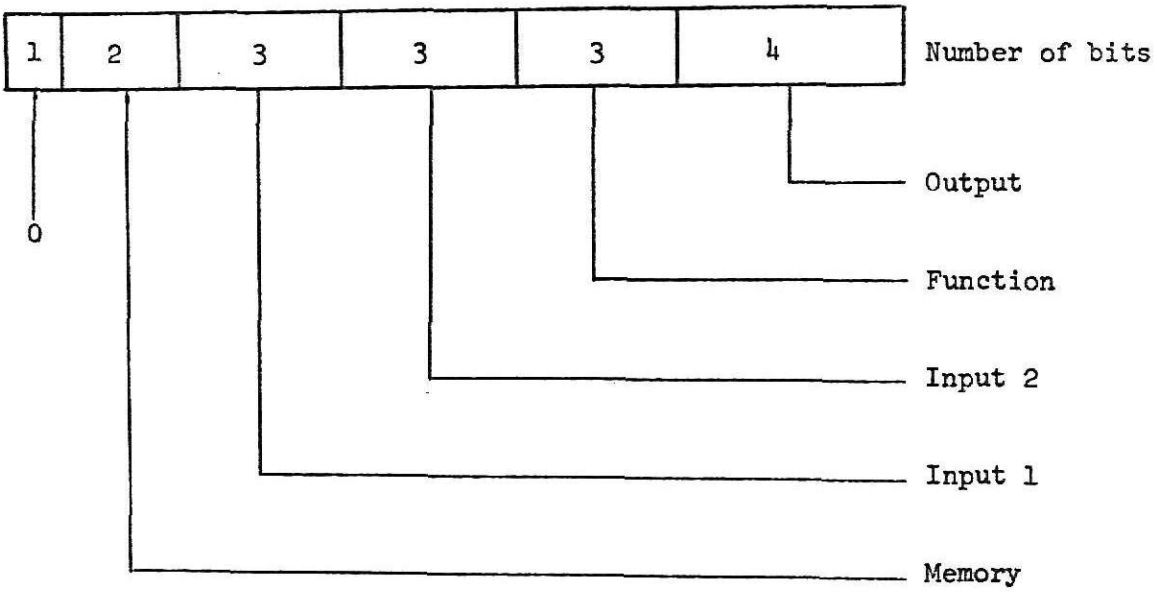

Figure 5. Micro instruction format for operation

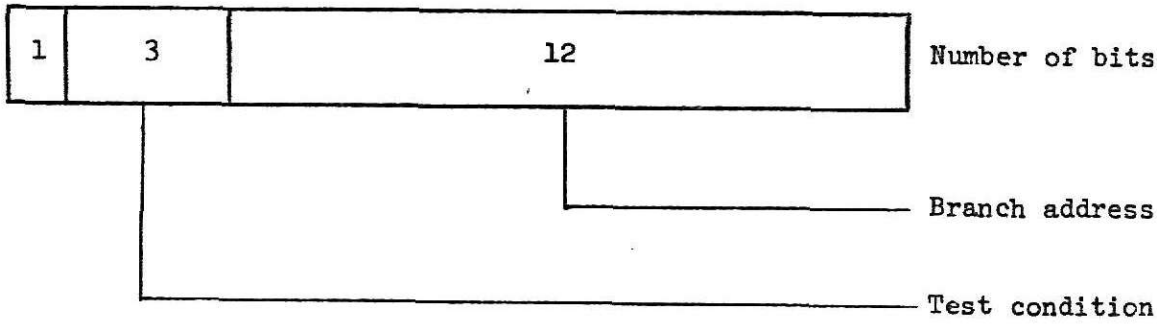

Figure 6. Micro instruction format for testing

TABLE III

ZERO-ADDRESS INSTRUCTION

Operation	Address	Comment
TO	J	Test MDR bit 0, branch to Micro location J if 0.
T1	J	Test MDR bit 1, branch to Micro location J if 0.
T2	J	Test MDR bit 2, branch to Micro location J if 0.
TMDR	J	Test MDR for all 0s, branch if true.
T1	J	Branch on 5 bits of IR.
TRM	J	Always

TABLE IV

FOUR-ADDRESS INSTRUCTION

Operation	Address Field
ADDM	I1, I2, OUT, Z
SUMB	I1, I2, OUT, Z
ANDM	I1, I2, OUT, Z
ORM	I1, I2, OUT, Z
NOT	I1, I2, OUT, Z
EOR	I1, I2, OUT, Z
LS	I1, I2, OUT, Z
RS	I1, I2, OUT, Z

TABLE V

MEMORY CONTROL (2 BITS)

Code Value	Action	Comment
0	NEXT	Go to fetch next micro instruction
1	READ	The READ or WRITE action is taken after rest of the micro instruction execution occurs
2	WRITE	
3	DUMP	Print out the contents of all register

TABLE VI

INPUT BUS 1 (3 BITS)

Code	Register or Number gated onto bus
0	0
1	MDR
2	X
3	-4
4	ZERO
5	ZERO
6	ZERO
7	-4

TABLE VII

INPUT BUS 2 (3 BITS)

Code	Register or Number Gated Onto Bus
0	0
1	1
2	FFFFFFFF (MASK)
3	A
4	CC
5	B
6	STK
7	-4

TABLE VIII

OUTPUT BUS (4 BITS)

Code	Out Bus Gated To
0	MAR
1	MDR
2	X
3	A
4	CC
5	B
6	STK
7	IR
8 to 15	ZERO

TABLE IX

FUNCTION (3 BITS)

Code	Output Function
0	IN1 $\div$ IN2 (Twos complement)
1	IN1 - IN2 (Twos complement)
2	IN1 "AND" IN2
3	IN1 "OR" IN2
4	NOT (IN1 + IN2)
5	IN1 "EXCLUSIVE OR" IN2
6	Half (IN1 + IN2) (RIGHT shift 1)
7	Double (IN1 + IN2) (Left shift 1)

TABLE X

TEST CONDITIONS (3 BITS)

Code	Branch If
0	ALWAYS
1	MDR BIT 0 0
2	MDR BIT 1 0
3	MDR BIT 2 0
4	MDR 0
5	ABEND1
6	ABEND2
7	The 5 bits in the IR register are added with the bottom 5 bits of the branch address and used for a branch.

In the one-address micro-instructions the first bit (should be 1) indicates that it is a test step, the next 3 bits indicate that MDR bit 0 is to be tested, while the final 12 bits are the address. In the four-address micro-instructions the addresses provided will have to have the appropriate values from the tables of codes. Hence the first bit should be 0, the next two bits are the Z part and the next three bits each should be I1, I2 and the value from the tables of codes. The last part, consisting of four bits, should be OUT.

In the micro-program (appendix C) statement numbers from 68 to 73 (including generated macro) are the 'FETCH' actions to fetch an instruction from S memory and then increment the instruction counter. The next statement numbers (from 74 to 91) are to 'INTERPRET' and test whether it was a one-address instruction or zero-address instruction, and if there was any indexing or indirect addressing to be done. The statement numbers from 93 to 120 and 122 to 150 are the zero-address and one-address jump-tables respectively. The last part of all operations was 'EXECUTION' to execute the micro-instructions.

III. THE SIMULATOR

The available memory in the S-machine was divided into sections so that the various pieces of the S-machine could be represented. Words were allocated to each of the registers in the data flow and a block of memory was allocated for storing the micro program. The remainder of the memory was used for the simulator. Figure 7 shows the allocation of memory for the S-machine.

The simulator program was written in the IBM OS/360 conventional usage assembler language. The micro control counter was kept in general purpose register TWO. the micro-instruction, which is assumed to occupy a sixteen-bit

halfword, was fetched into register THREE, and the sign bit was examined to see if it was a test micro instruction. If it was not the five control groups were extracted and placed in registers FOUR through EIGHT, each multiplied by four in order to the byte. The contents of input bus 1 and input bus 2 registers were placed in general purpose registers FOUR and FIVE prior to a branch to the program segment which handled the operation. If the sign bit was negative then the test control group was extracted and placed in register RIGHT. After addressing to the byte, it moved the branch address to register Five, and then jumped to the test condition program segment. The flow of the simulator is outlined in the following steps.

S1. Fetch next micro instruction.

S2. Increment micro control counter.

S3. Test sign bit. If negative go to S7. If positive go to S4.

S4. Extract function and register addresses.

S5. Check function group. If ADD go to A1. If SUB go to A2, If AND go to A3 If OR go to A4. If NOT go to A5. If EOR go to A6. If RS go to A7. If LS go to A8.

A1. OUT --- IN1 + IN2 go to S6.

A2. OUT --- IN1 - IN2 go to S6.

A3. OUT --- IN1 AND IN2 go to S6.

A4. OUT --- IN1 OR IN2 go to S6.

A5. OUT --- -(IN1 + IN2) go to S6.

A6. OUT --- IN1 EOR IN2 go to S6.

A7. OUT --- HALF(IN1 + IN2) go to S6.

A8. OUT --- DOUBLE(IN1 + IN2) go to S6.

S6. Check memory group. If zero go to S1. If READ go to B1.

If WRITE go to B2. Otherwise go to DUMP to print all register contents.

B1. Extract 12 bits from address and add address to start of MEMORY then fetch word from memory and store in MDR. go to S1.

B2. Form the address in a similar manner with READ and then copy the contents of the MDR register into the address location. go to S1.

S7. Extract condition and register addresses.

S8. Pick up the lower 12 bits of the micro word.

S9. Test bits 0 through 7 of MDR. If bit 0 go to S11. If bit 1 through 6 go to S10. If bit 7 go to S12.

S10. Test bit. If 1 go to S1. Otherwise go to S11.

S11. ADD address of start of memory to micro word then go to S1.

S12. ADD five bits of IR to micro word then go to S11.

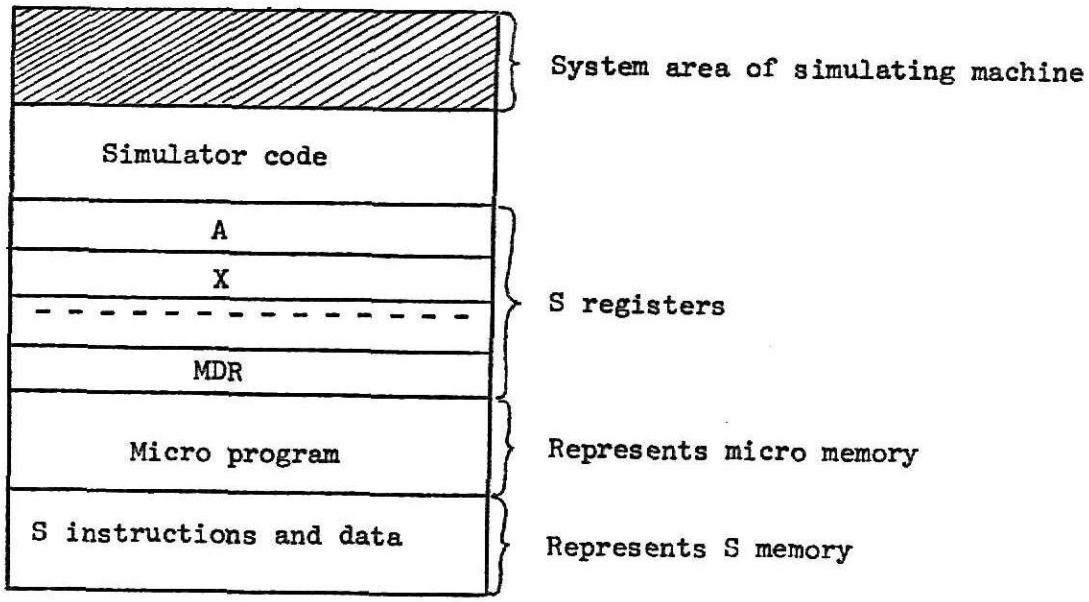

Figure 7. Memory use in simulator of S machine

SUMMARY

In this report the S-machine and its machine language have been described. The method of controlling the S-machine which required the writting of a program which was then used to control the data flow of the S-machine so that S-machine programs could be executed was also described. The S-machine instructions which were used to test the S-machine language were described by the micro-program and executed properly by the simulator program. The contents of all registers used in the data flow were printed by the tracing routine (appendix D). The results of this study showed that the microprogramming techniques have several advantages. Some of these advantages are flexibility, easy implementation of new OP codes and economical means of having large instruction sets on smaller type machine.

BIBLIOGRAPHY

1. WILKES, M. V. The Growth of Interest in Microprogramming Comp. Surveys (Sept. 1969). PP 139-145

2. ROSIN, R. F. Contemporary Concepts of Microprogramming and Emulation. Comp. Surveys (Dec. 1969), PP 197-212

3. GEAR, W. C. Computer Organization and Programming. New York: McGraw-Hill Book Co. 1969 PP 171-203

4. KNUTH, D. E. The Art of Computer Programming. Volume 1 Richard S. Varga and Michael A. Harrison, Editors. Addison-Wesley Publishing Company. 1968 PP 182-208 and PP 234-238

5. KENT, WILLIAM. Assembler-Language Macroprogramming: A Tutorial Oriented Toward the IBM 360. Comp. Surveys (Dec. 1969), PP 183-196

6. PAYNE, W. H. Machine, Assembly, and Systems Programming for the IBM 360. New York: Harper & Row, Publishers. 1969. PP 215-257

APPENDIX A

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

```
                                OS/360 ASSEMBLER

LEVEL=G       RELEASE=16JUL69       SYSTEM=MFT                  TIME=14:34:01       DAY=TUESDAY       DATE=26 MAY 70

ASSEMBLER OPTIONS=ESD,RLD,LIST,LOAD,NODECK,NORENT,NOTEST,EXTIME=5,NOBATCH,UTBUFF=3,FULLXREF,INSTSET=0,
                  LINECNT=46,NOEXECUTE,SPACE=MAX-2K.
```

SYMBOL	TYPE	ID	ADDR	LENGTH	LD ID
MEMORY	ER	01			
MCODE	ER	02			
SIMULATR	SD	03	000000	0005BD	
HEXCON2	ER	04			
PRINTBUF	ER	05			

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    1          MACRO
                                    2 &A       BITS  &N,&R,&TAB
                                    3 &A       SR    8,8          CLEAR REGISTER 8
                                    4          SLDL  8,&N         MOVE N BITS FROM 9 TO 8
                                    5          SLL   8,2          QUADRUPLE 8
                                    6          A     8,&TAB       ADD ADDRESS OF TABLE
                                    7          L     &R,0(8)      FETCH TABLE ENTRY TO REGISTER R
                                    8          MEND
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     10          EXTRN MEMORY,MCODE  ADDRESS OF S MEMORY AND MCODE
000000                               11 SIMULATR CSECT
                                     12          SAVE  (14,12),,*
000000 47F0 F00E             0000E   13+         B     14(0,15) BRANCH AROUND ID
000004 08                            14+         DC    AL1(8)
000005 E2C9D4E4D3C1E3D9              15+         DC    CL8'SIMULATR' IDENTIFIER
00000D 00
00000E 90EC D00C             0000C   16+         STM   14,12,12(13) SAVE REGISTERS
000012 05C0                          17          BALR  12,0
000014                               18          USING *,12
000014 5810 C58C             005A0   19          L     1,=A(SAV1)     LINKAGE CEREMONY
000018 5010 D008             00008   20          ST    1,8(,13)
00001C 50D0 1004             00004   21          ST    13,4(,1)
000020 18D1                          22          LR    13,1
000022 5820 C590             005A4   23          L     2,=F'46'       INITIALIZE LINESIZE FOR TRACE PRINT.
000026 5020 C338             0034C   24          ST    2,LINESIZE
00002A 5820 C2C8             002DC   25 INITIAL  L     2,AMCODE       INITIALIZE MICRO-CONTROL COUNTER.
00002E 5020 C334             00348   26 NEXT     ST    2,MCC          STORE MICRO-CONTROL COUNTER.
000032 4832 0000             00000   27          LH    3,0(2)         LOAD NEXT MICROINSTRUCTION.
000036 5A20 C594             005A8   28          A     2,=F'2'        INCREMENT MICROCONTROL COUNTER
00003A 1293                          29          LTR   9,3            TEST SIGN BIT
00003C 4740 C094             000A8   30          BM    TEST           GO TO TEST TYPES IF NEGATIVE
000040 8D80 0011             00011   31          SLDL  8,17           REMOVE SIGN BITS
                                     32          BITS  2,6,TMEM       NEXT 2 BITS TO REG 6,
000044 1B88                          33+         SR    8,8 CLEAR REGISTER 8
000046 8D80 0002             00002   34+         SLDL  8,2 MOVE N BITS FROM 9 TO 8
00004A 8980 0002             00002   35+         SLL   8,2 QUADRUPLE 8
00004E 5A80 C1C8             001DC   36+         A     8,TMEM ADD ADDRESS OF TABLE
000052 5868 0000             00000   37+         L     6,0(8) FETCH TABLE ENTRY TO REGISTER R
                                     38 *                       3 BITS TO REG 4 INDIRECTED
                                     39          BITS  3,4,TIN1       INDIRECT THROUGH MEN TABLE.
000056 1B88                          40+         SR    8,8 CLEAR REGISTER 8
000058 8D80 0003             00003   41+         SLDL  8,3 MOVE N BITS FROM 9 TO 8
00005C 8980 0002             00002   42+         SLL   8,2 QUADRUPLE 8
000060 5A80 C1CC             001E0   43+         A     8,TIN1 ADD ADDRESS OF TABLE
000064 5848 0000             00000   44+         L     4,0(8) FETCH TABLE ENTRY TO REGISTER R
                                     45 *                       DITTO FOR REG 5 AND TABLE IN-2
                                     46          BITS  3,5,TIN2       THROUGH IN-1 TABLE.
000068 1B88                          47+         SR    8,8 CLEAR REGISTER 8
00006A 8D80 0003             00003   48+         SLDL  8,3 MOVE N BITS FROM 9 TO 8
00006E 8980 0002             00002   49+         SLL   8,2 QUADRUPLE 8
000072 5A80 C1D0             001E4   50+         A     8,TIN2 ADD ADDRESS OF TABLE
000076 5858 0000             00000   51+         L     5,0(8) FETCH TABLE ENTRY TO REGISTER R
                                     52          BITS  3,7,TFUN       DITTO FOR REG 6 AND TABLE FUN.
00007A 1B88                          53+         SR    8,8 CLEAR REGISTER 8
00007C 8D80 0003             00003   54+         SLDL  8,3 MOVE N BITS FROM 9 TO 8
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
000080 8980 0002            00002    55+         SLL   8,2 QUADRUPLE 8
000084 5A80 C1D4            001E8    56+         A     8,TFUN ADD ADDRESS OF TABLE
000088 5878 0000            00000    57+         L     7,0(8) FETCH TABLE ENTRY TO REGISTER R
                                     58 *                       4 BITS TO REG 8 INDIRECTED
                                     59          BITS  4,8,TOUT     THROUGH TABLE OUT.
00008C 1B88                          60+         SR    8,8 CLEAR REGISTER 8
00008E 8D80 0004            00004    61+         SLDL  8,4 MOVE N BITS FROM 9 TO 8
000092 8980 0002            00002    62+         SLL   8,2 QUADRUPLE 8
000096 5A80 C1D8            001EC    63+         A     8,TOUT ADD ADDRESS OF TABLE
00009A 5888 0000            00000    64+         L     8,0(8) FETCH TABLE ENTRY TO REGISTER R
00009E 5844 0000            00000    65          L     4,0(4)       IN-1 OPERAND TO R4.
0000A2 5855 0000            00000    66          L     5,0(5)       IN-2 OPERAND TO R5.
0000A6 07F7                          67          BR    7            BRANCH TO FUNCTION PROGRAM.
0000A8 8D80 0011            00011    68 TEST     SLDL  8,17         REMOVE SIGN BITS.
                                     69 *                       3 BITS TO REG 4 INDIRECTED THROUGH
                                     70          BITS  3,4,TTEST    TEST CONDITIONS TABLE.
0000AC 1B88                          71+         SR    8,8 CLEAR REGISTER 8
0000AE 8D80 0003            00003    72+         SLDL  8,3 MOVE N BITS FROM 9 TO 8
0000B2 8980 0002            00002    73+         SLL   8,2 QUADRUPLE 8
0000B6 5A80 C1DC            001F0    74+         A     8,TTEST ADD ADDRESS OF TABLE
0000BA 5848 0000            00000    75+         L     4,0(8) FETCH TABLE ENTRY TO REGISTER R
0000BE 1B88                          76          SR    8,8          CLEAR REGISTER 8.
0000C0 8D80 000C            0000C    77          SLDL  8,12
0000C4 1858                          78          LR    5,8
0000C6 07F4                          79          BR    4            BRANCH TO TEST CONDITION PROGRAM.
0000C8 1A45                          80 ADDM     AR    4,5          FUNCTION PROGRAM FOR ADDM.
0000CA 5048 0000            00000    81          ST    4,0(8)
0000CE 07F6                          82          BR    6
0000D0 1845                          83 SUBM     SR    4,5
0000D2 5048 0000            00000    84          ST    4,0(8)
0000D6 07F6                          85          BR    6
0000D8 1445                          86 ANDM     NR    4,5
0000DA 5048 0000            00000    87          ST    4,0(8)
0000DE 07F6                          88          BR    6
0000E0 1645                          89 ORM      OR    4,5
0000E2 5048 0000            00000    90          ST    4,0(8)
0000E6 07F6                          91          BR    6
0000E8 1A45                          92 NOT      AR    4,5
0000EA 1344                          93          LCR   4,4
0000EC 5048 0000            00000    94          ST    4,0(8)
0000F0 07F6                          95          BR    6
0000F2 1745                          96 EOR      XR    4,5
0000F4 5048 0000            00000    97          ST    4,0(8)
0000F8 07F6                          98          BR    6
0000FA 1A45                          99 RS       AR    4,5
0000FC 8840 0001            00001   100          SRL   4,1
```

```
  LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
000100 5048 0000              00000   101          ST    4,0(8)
000104 07F6                           102          BR    6
000106 1A45                           103 LS       AR    4,5
000108 8940 0001              00001   104          SLL   4,1
00010C 5048 0000              00000   105          ST    4,0(8)
000110 07F6                           106          BR    6
000112 58A0 C32C              00340   107 RMEM     L     10,MAR          MEMORY CONTROL GROUP PROGRAM FOR READ.
000116 54A0 C33C              00350   108          N     10,MASK12       EXTRACT 12 BITS ADDRESS.
00011A 5AA0 C2C4              002D8   109          A     10,AMEMORY      ADD ADDRESS OF START OF MEMORY.
00011E 58AA 0000              00000   110          L     10,0(10)        FETCH WORD FROM MEMORY.
000122 50A0 C330              00344   111          ST    10,MDR          PLACE IN MDR.
000126 47F0 C12A              0013E   112          B     DUMP
00012A 58A0 C32C              00340   113 WMEM     L     10,MAR
00012E 54A0 C33C              00350   114          N     10,MASK12
000132 5AA0 C2C4              002D8   115          A     10,AMEMORY
000136 58B0 C330              00344   116          L     11,MDR
00013A 50B0 A000              00000   117          ST    11,0(,10)
00013E 58F0 C598              005AC   118 DUMP     L     15,=A(TRACE)
000142 05EF                           119          BALR  14,15
000144 47F0 C01A              0002E   120          B     NEXT
000148 5860 C330              00344   121 ST0      L     6,MDR           TEST PROGRAM FOR ST0.
00014C 5460 C340              00354   122          N     6,MASK8         TEST BIT 0 OF MDR.
000150 1266                           123          LTR   6,6
000152 4770 C12A              0013E   124          BNZ   DUMP
000156 47F0 C186              0019A   125          B     AMCC            BRANCH ADDRESS TO MICRO CONTROL COUNTER.
00015A 5860 C330              00344   126 ST1      L     6,MDR           TEST PROGRAM FOR ST1.
00015E 5460 C344              00358   127          N     6,MASK4         TEST BIT 1 OF MDR.
000162 1266                           128          LTR   6,6
000164 4770 C12A              0013E   129          BNZ   DUMP
000168 47F0 C186              0019A   130          B     AMCC
00016C 5860 C330              00344   131 ST2      L     6,MDR           TEST PROGRAM FOR ST2.
000170 5460 C348              0035C   132          N     6,MASK2         TEST BIT 2 OF MDR.
000174 1266                           133          LTR   6,6
000176 4770 C12A              0013E   134          BNZ   DUMP
00017A 47F0 C186              0019A   135          B     AMCC
00017E 5860 C330              00344   136 STMDR    L     6,MDR           TEST PROGRAM FOR STMDR.
000182 1266                           137          LTR   6,6             TEST ALL BITS OF MDR.
000184 4770 C12A              0013E   138          BNZ   DUMP
000188 47F0 C186              0019A   139          B     AMCC
00018C 5860 C328              00334   140 STI      L     6,IR
000190 8960 0003              00003   141          SLL   6,3
000194 8860 001B              0001B   142          SRL   6,27
000198 1A56                           143          AR    5,6
00019A 1875                           144 AMCC     LR    7,5
00019C 8970 0001              00001   145          SLL   7,1
0001A0 5A70 C2C8              002DC   146          A     7,AMCODE
```

```
  LOC   OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
0001A4 1827                        147          LR    2,7
0001A6 47F0 C12A             0013E 148          B     DUMP
0001AA 58D0 D004             00004 149          L     13,4(,13)
                                   150          RETURN (14,12),T,RC=0
0001AE 98EC D00C             0000C 151+         LM    14,12,12(13) RESTORE THE REGISTERS
0001B2 92FF D00C       0000C       152+         MVI   12(13),X'FF' SET RETURN INDICATION
0001B6 41F0 0000             00000 153+         LA    15,0(0,0) LOAD RETURN CODE
0001BA 07FE                        154+         BR    14 RETURN
                                   155 ABEND1   ABEND 1,DUMP
0001BC                             156+         CNOP  0,4
0001BC 47F0 C1B0             001C4 157+ABEND1   B     *+8 BRANCH AROUND CONSTANT
0001C0 80                          158+         DC    AL1(128) DUMP/STEP CODE
0001C1 000001                      159+         DC    AL3(1) COMPLETION CODE
0001C4 5810 C1AC             001C0 160+         L     1,*-4 LOAD CODES INTO REG 1
0001C8 0A0D                        161+         SVC   13 LINK TO ABEND ROUTINE
                                   162 ABEND2   ABEND 2,DUMP
0001CA 0700                        163+         CNOP  0,4
0001CC 47F0 C1C0             001D4 164+ABEND2   B     *+8 BRANCH AROUND CONSTANT
0001D0 80                          165+         DC    AL1(128) DUMP/STEP CODE
0001D1 000002                      166+         DC    AL3(2) COMPLETION CODE
0001D4 5810 C1BC             001D0 167+         L     1,*-4 LOAD CODES INTO REG 1
0001D8 0A0D                        168+         SVC   13 LINK TO ABEND ROUTINE
                                   169 *
0001DA 0000
0001DC 000001F4                    170 TMEM     DC    A(MEM)          ADDRESSES OF TABLES FOR CONTROL GROUP.
0001E0 00000204                    171 TIN1     DC    A(IN1)
0001E4 00000224                    172 TIN2     DC    A(IN2)
0001E8 00000244                    173 TFUN     DC    A(FUN)
0001EC 00000264                    174 TOUT     DC    A(OUT)
0001F0 000002A4                    175 TTEST    DC    A(TESTM)
                                   176 *
0001F4 0000013E                    177 MEM      DC    A(DUMP)
0001F8 00000112                    178          DC    A(RMEM)
0001FC 0000012A                    179          DC    A(WMEM)
000200 0000013E                    180          DC    A(DUMP)
                                   181 *
000204 000002C4                    182 IN1      DC    A(ZERO)         TABLE OF ADDRESSES OF
000208 00000344                    183          DC    A(MDR)          S REGISTERS OR CONSTANTS.
00020C 00000338                    184          DC    A(X)
000210 000002D0                    185          DC    A(MFOUR)
000214 000002C4                    186          DC    A(ZERO)
000218 000002C4                    187          DC    A(ZERO)
00021C 000002C4                    188          DC    A(ZERO)
000220 000002CC                    189          DC    A(FOUR)
                                   190 *
000224 000002C4                    191 IN2      DC    A(ZERO)
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000228 000002C8                   192          DC    A(ONE)
00022C 000002D4                   193          DC    A(MASK)
000230 00000328                   194          DC    A(A)
000234 00000330                   195          DC    A(CC)
000238 0000032C                   196          DC    A(B)
00023C 00000334                   197          DC    A(STK)
000240 000002CC                   198          DC    A(FOUR)
                                  199 *
000244 000000C8                   200 FUN      DC    A(ADDM)       TABLE OF ADDRESSES OF PROGRAMS
000248 000000D0                   201          DC    A(SUBM)       FOR FUNCTION.
00024C 000000D8                   202          DC    A(ANDM)
000250 000000E0                   203          DC    A(ORM)
000254 000C00E8                   204          DC    A(NOT)
000258 000C00F2                   205          DC    A(EOR)
00025C 000000FA                   206          DC    A(RS)
000260 00000106                   207          DC    A(LS)
                                  208 *
000264 00000340                   209 OUT      DC    A(MAR)        ADDRESSES OF REGISTERS FOR OUTPUT.
000268 00000344                   210          DC    A(MDR)
00026C 00000338                   211          DC    A(X)
000270 00000328                   212          DC    A(A)
000274 00000330                   213          DC    A(CC)
000278 0000032C                   214          DC    A(B)
00027C 00000334                   215          DC    A(STK)
000280 0000033C                   216          DC    A(IR)
000284 000002C4                   217          DC    A(ZERO)
000288 000002C4                   218          DC    A(ZERO)
00028C 000002C4                   219          DC    A(ZERO)
000290 000002C4                   220          DC    A(ZERO)
000294 000002C4                   221          DC    A(ZERO)
000298 000002C4                   222          DC    A(ZERO)
00029C 000002C4                   223          DC    A(ZERO)
0002A0 000002C4                   224          DC    A(ZERO)
                                  225 *
0002A4 0000019A                   226 TESTM    DC    A(AMCC)       TABLE OF BRANCH ADDRESS FOR TEST GROUP.
0002A8 00000148                   227          DC    A(ST0)
0002AC 0000015A                   228          DC    A(ST1)
0002B0 0000016C                   229          DC    A(ST2)
0002B4 0000017E                   230          DC    A(STMDR)
0002B8 000001BC                   231          DC    A(ABEND1)
0002BC 000001CC                   232          DC    A(ABEND2)
0002C0 0000018C                   233          DC    A(STI)
                                  234 *
0002C4 00000000                   235 ZERO     DC    F'0'
0002C8 00000001                   236 ONE      DC    F'1'
0002CC 00000004                   237 FOUR     DC    F'4'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
0002D0 FFFFFFFC                     238 MFOUR    DC    F'-4'
0002D4 00FFFFFF                     239 MASK     DC    X'00FFFFFF'   24 ONE BITS (HEXADECIMAL CONSTANT)
0002D8 00000000                     240 AMEMORY  DC    A(MEMORY)     ADRESS OF MEMORY LOCATION 0
0002DC 00000000                     241 AMCODE   DC    A(MCODE)      ADRESS OF MICRO MEMORY LOCATION 0
0002E0 0000000000000000             242 SAV1     DC    18A(0)
000328                              243 A        DS    F        DEFINITIONS OF S REGISTERS.
00032C                              244 B        DS    F
000330                              245 CC       DS    F
000334 00000FFC                     246 STK      DC    A(MEMORY+4096-4)
000338                              247 X        DS    F
00033C                              248 IR       DS    F
000340                              249 MAR      DS    F
000344                              250 MDR      DS    F
000348                              251 MCC      DS    F
00034C                              252 LINESIZE DS    F
000350                              253          DS    0F
000350 00003FFC                     254 MASK12   DC    X'00003FFC'
000354 80000000                     255 MASK8    DC    X'80000000'
000358 40000000                     256 MASK4    DC    X'40000000'
00035C 20000000                     257 MASK2    DC    X'20000000'
                                    258 *
                                    259 *        TRACE ALL REGISTERS.
                                    260 *
                                    261 TRACE    PRINT SA=SVTRACE,CSECT=NO
000360 47F0 F00A            0000A   262+TRACE    B     10(0,15) BRANCH AROUND ID
000364 05                           263+         DC    AL1(5)
000365 E3D9C1C3C5                   264+         DC    CL5'TRACE' IDENTIFIER
00036A 90EC D00C            0000C   265+         STM   14,12,12(13) SAVE REGISTERS
00036E 05C0                         266+         BALR  12,0 CREATE BASE REGISTER
000370                              267+         USING *,12
000370 1830                         268+         LR    3,0 SAVE -0-
000372 1821                         269+         LR    2,1 SAVE PARAMETER REGISTER
000374 4110 C160            004D0   270+         LA    1,SVTRACE GET ADDRESS OF SAVE AREA
000378 501D 0008            00008   271+         ST    1,8(13) STORE FORWARD POINTER
00037C 50D1 0004            00004   272+         ST    13,4(1) STORE BACKWARD POINTER
000380 18D1                         273+         LR    13,1 SET UP REG FOR SAVE AREA
000382 1812                         274+         LR    1,2 RESTORE PARAMETER REGISTER
000384 1803                         275+         LR    0,3 RESTORE -0-
000386 9240 C1A8      00518         276          MVI   OUTT,C' '
00038A D284 C1A9 C1A8 00519 00518   277          MVC   OUTT+1(133),OUTT
                                    278          HXOUT OUTT+21,A
000390                              279+         CNOP  0,4
000390 4510 C030            003A0   280+         BAL   1,*+16
000394 0000052D                     281+         DC    A(OUTT+21) BYTE ADDRESS
000398 00000328                     282+         DC    A(A) FULL WORD ADDRESS
00039C 00000000                     283+         DC    V(HEXCON2) HEXCON2  -  FULL WORD TO BYTE
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
0003A0 58F0 C02C          0039C   284+         L     15,*-4
0003A4 05EF                       285+         BALR  14,15
                                  286          HXOUT OUTT+31,B
0003A6 0700                       287+         CNOP  0,4
0003A8 4510 C048          003B8   288+         BAL   1,*+16
0003AC 00000537                   289+         DC    A(OUTT+31) BYTE ADDRESS
0003B0 0000032C                   290+         DC    A(B) FULL WORD ADDRESS
0003B4 00000000                   291+         DC    V(HEXCON2) HEXCON2  -  FULL WORD TO BYTE
0003B8 58F0 C044          003B4   292+         L     15,*-4
0003BC 05EF                       293+         BALR  14,15
                                  294          HXOUT OUTT+41,CC
0003BE 0700                       295+         CNOP  0,4
0003C0 4510 C060          003D0   296+         BAL   1,*+16
0003C4 00000541                   297+         DC    A(OUTT+41) BYTE ADDRESS
0003C8 00000330                   298+         DC    A(CC) FULL WORD ADDRESS
0003CC 00000000                   299+         DC    V(HEXCON2) HEXCON2  -  FULL WORD TO BYTE
0003D0 58F0 C05C          003CC   300+         L     15,*-4
0003D4 05EF                       301+         BALR  14,15
                                  302          HXOUT OUTT+51,STK
0003D6 0700                       303+         CNOP  0,4
0003D8 4510 C078          003E8   304+         BAL   1,*+16
0003DC 0000054B                   305+         DC    A(OUTT+51) BYTE ADDRESS
0003E0 00000334                   306+         DC    A(STK) FULL WORD ADDRESS
0003E4 00000000                   307+         DC    V(HEXCON2) HEXCON2  -  FULL WORD TO BYTE
0003E8 58F0 C074          003E4   308+         L     15,*-4
0003EC 05EF                       309+         BALR  14,15
                                  310          HXOUT OUTT+61,X
0003EE 0700                       311+         CNOP  0,4
0003F0 4510 C090          00400   312+         BAL   1,*+16
0003F4 00000555                   313+         DC    A(OUTT+61) BYTE ADDRESS
0003F8 00000338                   314+         DC    A(X) FULL WORD ADDRESS
0003FC 00000000                   315+         DC    V(HEXCON2) HEXCON2  -  FULL WORD TO BYTE
0004C0 58F0 C08C          003FC   316+         L     15,*-4
000404 05EF                       317+         BALR  14,15
                                  318          HXOUT OUTT+71,IR
000406 0700                       319+         CNOP  0,4
0004C8 4510 C0A8          00418   320+         BAL   1,*+16
00040C 0000055F                   321+         DC    A(OUTT+71) BYTE ADDRESS
000410 0000033C                   322+         DC    A(IR) FULL WORD ADDRESS
000414 00000000                   323+         DC    V(HEXCON2) HEXCON2  -  FULL WORD TO BYTE
000418 58F0 C0A4          00414   324+         L     15,*-4
00041C 05EF                       325+         BALR  14,15
                                  326          HXOUT OUTT+81,MAR
00041E 0700                       327+         CNOP  0,4
000420 4510 C0C0          00430   328+         BAL   1,*+16
000424 00000569                   329+         DC    A(OUTT+81) BYTE ADDRESS
```

```
000428 00000340                      330+         DC    A(MAR) FULL WORD ADDRESS
00042C 00000000                      331+         DC    V(HEXCON2) HEXCON2  -  FULL WORD TO BYTE
000430 58F0 C0BC             0042C   332+         L     15,*-4
000434 05EF                          333+         BALR  14,15
                                     334          HXOUT OUTT+91,MDR
000436 0700                          335+         CNOP  0,4
000438 4510 C0D8             00448   336+         BAL   1,*+16
00043C 00000573                      337+         DC    A(OUTT+91) BYTE ADDRESS
000440 00000344                      338+         DC    A(MDR) FULL WORD ADDRESS
000444 00000000                      339+         DC    V(HEXCON2) HEXCON2  -  FULL WORD TO BYTE
000448 58F0 C0D4             00444   340+         L     15,*-4
00044C 05EF                          341+         BALR  14,15
                                     342          HXOUT OUTT+101,MCC
00044E 0700                          343+         CNOP  0,4
000450 4510 C0F0             00460   344+         BAL   1,*+16
000454 0000057D                      345+         DC    A(OUTT+101) BYTE ADDRESS
000458 00000348                      346+         DC    A(MCC) FULL WORD ADDRESS
00045C 00000000                      347+         DC    V(HEXCON2) HEXCON2  -  FULL WORD TO BYTE
000460 58F0 C0EC             0045C   348+         L     15,*-4
000464 05EF                          349+         BALR  14,15
                                     350          PUTST OUTT
000466 0700                          351+         CNOP  0,4
000468 4510 C104             00474   352+         BAL   1,*+12
00046C 00000518                      353+         DC    A(OUTT)
000470 000C0000                      354+         DC    V(PRINTBUF)
000474 58F0 C100             00470   355+         L     15,*-4
000478 05EF                          356+         BALR  14,15
00047A 58A0 C240             005B0   357          L     10,=A(LINESIZE)
00047E 589A 0000             00000   358          L     9,0(10)
000482 5890 C244             005B4   359          S     9,=F'1'
000486 509A 0000             00000   360          ST    9,0(10)
00048A 5990 C248             005B8   361          C     9,=F'0'
00048E 4770 C14E             004BE   362          BNE   RETURN
000492 9240 C1A8       00518         363          MVI   OUTT,C' '
000496 D283 C1A9 C1A8 00519 00518    364          MVC   OUTT+1(132),OUTT
00049C D200 C1A8 C24C 00518 005BC    365          MVC   OUTT(1),=C'1'
                                     366          PUTST OUTT
0004A2 0700                          367+         CNOP  0,4
0004A4 4510 C140             004B0   368+         BAL   1,*+12
0004A8 00000518                      369+         DC    A(OUTT)
0004AC 00000000                      370+         DC    V(PRINTBUF)
0004B0 58F0 C13C             004AC   371+         L     15,*-4
0004B4 05EF                          372+         BALR  14,15
0004B6 5890 C234             005A4   373          L     9,=F'46'
0004BA 509A 0000             00000   374          ST    9,0(10)
                                     375 RETURN   TERME
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

0004BE 58DD 0004             00004   376+RETURN   L _    13,4(13) RESTORE REGISTER -13-
0004C2 98EC D00C             0000C   377+         LM     14,12,12(13) RESTORE THE REGISTERS
0004C6 92FF D00C       0000C         378+         MVI    12(13),X'FF' SET RETURN INDICATION
0004CA 41F0 0000             00000   379+         LA     15,0(0,0) LOAD RETURN CODE
0004CE 07FE                          380+         BR     14 RETURN
000400 0000C00000000000              381 SVTRACE  DC     18A(0)
000518                               382 OUTT     DS     133C
                                     383          END
0005A0 000002E0                      384                 =A(SAV1)
0005A4 0000002E                      385                 =F'46'
0005A8 00000002                      386                 =F'2'
0005AC 00000360                      387                 =A(TRACE)
0005B0 0000034C                      388                 =A(LINESIZE)
0005B4 00000001                      389                 =F'1'
0005B8 00000000                      390                 =F'0'
0005BC F1                            391                 =C'1'
```

26 MAY 70

SYMBOL	LEN	VALUE	DEFN	REFERENCES													
A	4	000328	243	194	212	282											
ABEND1	4	0001BC	157	231													
ABEND2	4	0001CC	164	232													
ADDM	2	0000C8	80	200													
AMCC	2	00019A	144	125	130	135	139	226									
AMCODE	4	0002DC	241	25	146												
AMEMORY	4	0002D8	240	109	115												
ANDM	2	0000D8	86	202													
B	4	00032C	244	196	214	290											
CC	4	000330	245	195	213	298											
DUMP	4	00013E	118	112	124	129	134	138	148	177	180						
EOR	2	0000F2	96	205													
FOUR	4	0002CC	237	189	198												
FUN	4	000244	200	173													
INITIAL	4	00002A	25														
IN1	4	000204	182	171													
IN2	4	000224	191	172													
IR	4	00033C	248	140	216	322											
LINESIZE	4	00034C	252	24	357	388											
LS	2	000106	103	207													
MAR	4	000340	249	107	113	209	330										
MASK	4	0002D4	239	193													
MASK12	4	000350	254	108	114												
MASK2	4	00035C	257	132													
MASK4	4	000358	256	127													
MASK8	4	000354	255	122													
MCC	4	000348	251	26	346												
MCODE	1	000000	10	241													
MDR	4	000344	250	111	116	121	126	131	136	183	210	338					
MEM	4	0001F4	177	170													
MEMORY	1	000000	10	240	246												
MFOUR	4	0002D0	238	185													
NEXT	4	00002E	26	120													
NOT	2	0000E8	92	204													
ONE	4	0002C8	236	192													
ORM	2	0000E0	89	203													
OUT	4	000264	209	174													
OUTT	1	000518	382	276	277	277	281	289	297	305	313	321	329	337	345	353	363
				364	364	365	369										
RETURN	4	00048E	376	362													
RMEM	4	000112	107	178													
RS	2	0000FA	99	206													
SAV1	4	0002E0	242	19	384												
SIMULATR	1	000000	11														
STI	4	00018C	140	233													
STK	4	000334	246	197	215	306											

SYMBOL	LEN	VALUE	DEFN	REFERENCES												
STMDR	4	00017E	136	230												
STO	4	000148	121	227												
ST1	4	00015A	126	228												
ST2	4	00016C	131	229												
SUBM	2	0000D0	83	201												
SVTRACE	4	0004D0	381	270												
TEST	4	0000A8	68	30												
TESTM	4	0002A4	226	175												
TFUN	4	0001E8	173	56												
TIN1	4	0001E0	171	43												
TIN2	4	0001E4	172	50												
TMEM	4	0001DC	170	36												
TOUT	4	0001EC	174	63												
TRACE	4	000360	262	118	387											
TTEST	4	0001F0	175	74												
WMEM	4	00012A	113	179												
X	4	000338	247	184	211	314										
ZERO	4	0002C4	235	182	186	187	188	191	217	218	219	220	221	222	223	224

POS.ID	REL.ID	FLAGS	ADDRESS
03	01	0C	0002D8
03	01	0C	000334
03	02	0C	0002DC
03	03	0C	0001DC
03	03	0C	0001E0
03	03	0C	0001E4
03	03	0C	0001E8
03	03	0C	0001EC
03	03	0C	0001F0
03	03	0C	0001F4
03	03	0C	0001F8
03	03	0C	0001FC
03	03	0C	000200
03	03	0C	000204
03	03	0C	000208
03	03	0C	00020C
03	03	0C	000210
03	03	0C	000214
03	03	0C	000218
03	03	0C	00021C
03	03	0C	000220
03	03	0C	000224
03	03	0C	000228
03	03	0C	00022C
03	03	0C	000230
03	03	0C	000234
03	03	0C	000238
03	03	0C	00023C
03	03	0C	000240
03	03	0C	000244
03	03	0C	000248
03	03	0C	00024C
03	03	0C	000250
03	03	0C	000254
03	03	0C	000258
03	03	0C	00025C
03	03	0C	000260
03	03	0C	000264
03	03	0C	000268
03	03	0C	00026C
03	03	0C	000270
03	03	0C	000274
03	03	0C	000278
03	03	0C	00027C
03	03	0C	000280
03	03	0C	000284

POS.ID	REL.ID	FLAGS	ADDRESS
03	03	0C	000288
03	03	0C	00028C
03	03	0C	000290
03	03	0C	000294
03	03	0C	000298
03	03	0C	00029C
03	03	0C	0002A0
03	03	0C	0002A4
03	03	0C	0002A8
03	03	0C	0002AC
03	03	0C	0002B0
03	03	0C	0002B4
03	03	0C	0002B8
03	03	0C	0002BC
03	03	0C	0002C0
03	03	0C	000394
03	03	0C	000398
03	03	0C	0003AC
03	03	0C	0003B0
03	03	0C	0003C4
03	03	0C	0003C8
03	03	0C	0003DC
03	03	0C	0003E0
03	03	0C	0003F4
03	03	0C	0003F8
03	03	0C	00040C
03	03	0C	000410
03	03	0C	000424
03	03	0C	000428
03	03	0C	00043C
03	03	0C	000440
03	03	0C	000454
03	03	0C	000458
03	03	0C	00046C
03	03	0C	0004A8
03	03	0C	0005A0
03	03	0C	0005AC
03	03	0C	0005B0
03	04	1C	00039C
03	04	1C	0003B4
03	04	1C	0003CC
03	04	1C	0003E4
03	04	1C	0003FC
03	04	1C	000414
03	04	1C	00042C
03	04	1C	000444

POS.ID	REL.ID	FLAGS	ADDRESS
03	04	1C	00045C
03	05	1C	000470
03	05	1C	0004AC

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

APPENDIX B

```
                                        OS/360 ASSEMBLER

LEVEL=G       RELEASE=16JUL69       SYSTEM=MFT                    TIME=14:35:36       DAY=TUESDAY       DATE=26 MAY 70

ASSEMBLER OPTIONS=ESD,RLD,LIST,LOAD,NODECK,NORENT,NOTEST,EXTIME=5,NOBATCH,UTBUFF=3,FULLXREF,INSTSET=0,
                  LINECNT=46,NOEXECUTE,SPACE=MAX-2K.
```

EXTERNAL SYMBOL DICTIONARY PAGE 1

26 MAY 70

SYMBOL	TYPE	ID	ADDR	LENGTH	LD ID
MEMORY	SD	01	000000	00103C	

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    1          MACRO
                                    2          SMOM  &OP,&B,&A,&X,&I
                                    3          LCLA  &II,&XX
                                    4          AIF   ('&I' EQ '').IBLNK
                                    5 &II      SETA  1
                                    6 .IBLNK   AIF   ('&X' EQ '').XBLNK
                                    7 &XX      SETA  1
                                    8 .XBLNK   ANOP
                                    9 &B       DC    AL.1(&II),AL.1(&XX),AL.6(&OP),AL3(&A-MEMORY)
                                   10          MEND
                                   11          MACRO
                                   12 &B       LOAD  &A,&X,&I
                                   13          LCLA  &OP
                                   14 &OP      SETA  0
                                   15          SMOM  &OP,&B,&A,&X,&I
                                   16          MEND
                                   17          MACRO
                                   18 &B       LDI   &A,&X,&I
                                   19          LCLA  &OP
                                   20 &OP      SETA  1
                                   21          SMOM  &OP,&B,&A,&X,&I
                                   22          MEND
                                   23          MACRO
                                   24 &B       STORE &A,&X,&I
                                   25          LCLA  &OP
                                   26 &OP      SETA  2
                                   27          SMOM  &OP,&B,&A,&X,&I
                                   28          MEND
                                   29          MACRO
                                   30 &B       TRA   &A,&X,&I
                                   31          LCLA  &OP
                                   32 &OP      SETA  3
                                   33          SMOM  &OP,&B,&A,&X,&I
                                   34          MEND
                                   35          MACRO
                                   36 &B       TPL   &A,&X,&I
                                   37          LCLA  &OP
                                   38 &OP      SETA  4
                                   39          SMOM  &OP,&B,&A,&X,&I
                                   40          MEND
                                   41          MACRO
                                   42 &B       TMI   &A,&X,&I
                                   43          LCLA  &OP
                                   44 &OP      SETA  5
                                   45          SMOM  &OP,&B,&A,&X,&I
                                   46          MEND
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                  47          MACRO
                                  48 &B       TZE   &A,&X,&I
                                  49          LCLA  &OP
                                  50 &OP      SETA  6
                                  51          SMOM  &OP,&B,&A,&X,&I
                                  52          MEND
                                  53          MACRO
                                  54 &B       TNZ   &A,&X,&I
                                  55          LCLA  &OP
                                  56 &OP      SETA  7
                                  57          SMOM  &OP,&B,&A,&X,&I
                                  58          MEND
                                  59          MACRO
                                  60 &B       ENTER &A,&X,&I
                                  61          LCLA  &OP
                                  62 &OP      SETA  8
                                  63          SMOM  &OP,&B,&A,&X,&I
                                  64          MEND
                                  65          MACRO
                                  66 &B       LDX   &A,&X,&I
                                  67          LCLA  &OP
                                  68 &OP      SETA  9
                                  69          SMOM  &OP,&B,&A,&X,&I
                                  70          MEND
                                  71          MACRO
                                  72 &B       LDXI  &A,&X,&I
                                  73          LCLA  &OP
                                  74 &OP      SETA  10
                                  75          SMOM  &OP,&B,&A,&X,&I
                                  76          MEND
                                  77          MACRO
                                  78 &B       LOOP  &A,&J,&I
                                  79          LCLA  &OP
                                  80 &OP      SETA  11
                                  81          SMOM  &OP,&B,&A,&X,&I
                                  82          MEND
                                  83          MACRO
                                  84 &B       LS    &A,&X,&I
                                  85          LCLA  &OP
                                  86 &OP      SETA  12
                                  87          SMOM  &OP,&B,&A,&X,&I
                                  88          MEND
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                  93          SMDM  &OP,&B,&A,&X,&I
                                  94          MEND
                                  95          MACRO
                                  96 &A       ADD
                                  97 &A       DC    AL.2(0),AL.6(32),AL3(0)
                                  98          MEND
                                  99          MACRO
                                 100 &A       SUB
                                 101 &A       DC    AL.2(0),AL.6(33),AL3(0)
                                 102          MEND
                                 103          MACRO
                                 104 &A       AND
                                 105 &A       DC    AL.2(0),AL.6(34),AL3(0)
                                 106          MEND
                                 107          MACRO
                                 108 &A       ORS
                                 109 &A       DC    AL.2(0),AL.6(35),AL3(0)
                                 110          MEND
                                 111          MACRO
                                 112 &A       EOR
                                 113 &A       DC    AL.2(0),AL.6(36),AL3(0)
                                 114          MEND
                                 115          MACRO
                                 116 &A       NOT
                                 117 &A       DC    AL.2(0),AL.6(37),AL3(0)
                                 118          MEND
                                 119          MACRO
                                 120 &A       XTS
                                 121 &A       DC    AL.2(0),AL.6(38),AL3(0)
                                 122          MEND
                                 123          MACRO
                                 124 &A       STX
                                 125 &A       DC    AL.2(0),AL.6(39),AL3(0)
                                 126          MEND
                                 127          MACRO
                                 128 &A       ADX
                                 129 &A       DC    AL.2(0),AL.6(40),AL3(0)
                                 130          MEND
                                 131          MACRO
                                 132 &A       SBX
                                 133 &A       DC    AL.2(0),AL.6(41),AL3(0)
                                 134          MEND
                                 135          MACRO
                                 136 &A       RET
                                 137 &A       DC    AL.2(0),AL.6(42),AL3(0)
                                 138          MEND
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                  139           MACRO
                                  140 &A        POP
                                  141 &A        DC     AL.2(0),AL.6(43),AL3(0)
                                  142           MEND
                                  143           MACRO
                                  144 &A        STOP
                                  145 &A        DC     AL.2(0),AL.6(44),AL3(0)
                                  146           MEND
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     148           ENTRY MEMORY
000000                               149 MEMORY    CSECT
                                     150           LOAD  NN
000000 00000030                      151+          DC    AL.1(0),AL.1(0),AL.6(0),AL3(NN-MEMORY)
                                     152           LOAD  TWO
000004 00000034                      153+          DC    AL.1(0),AL.1(0),AL.6(0),AL3(TWO-MEMORY)
                                     154           TRA   B
000008 0300001C                      155+          DC    AL.1(0),AL.1(0),AL.6(3),AL3(B-MEMORY)
                                     156 A         LOAD  TWO
00000C 00000034                      157+A         DC    AL.1(0),AL.1(0),AL.6(0),AL3(TWO-MEMORY)
                                     158           ADD
000010 20000000                      159+          DC    AL.2(0),AL.6(32),AL3(0)
                                     160           STORE RESULT
000014 0200003C                      161+          DC    AL.1(0),AL.1(0),AL.6(2),AL3(RESULT-MEMORY)
                                     162           TRA   C
000018 0300002C                      163+          DC    AL.1(0),AL.1(0),AL.6(3),AL3(C-MEMORY)
                                     164 B         STORE RESULT
00001C 0200003C                      165+B         DC    AL.1(0),AL.1(0),AL.6(2),AL3(RESULT-MEMORY)
                                     166           LOAD  ONE
000020 00000038                      167+          DC    AL.1(0),AL.1(0),AL.6(0),AL3(ONE-MEMORY)
                                     168           SUB
000024 21000000                      169+          DC    AL.2(0),AL.6(33),AL3(0)
                                     170           TNZ   A
000028 0700000C                      171+          DC    AL.1(0),AL.1(0),AL.6(7),AL3(A-MEMORY)
                                     172 C         STOP
00002C 2C000000                      173+C         DC    AL.2(0),AL.6(44),AL3(0)
000030 00000005                      174 NN        DC    F'5'
000034 00000002                      175 TWO       DC    F'2'
000038 00000001                      176 ONE       DC    F'1'
00003C                               177 RESULT    DS    F
000040 0000000000000000              178           DC    ((4096-4)/4)F'0'
                                     179           END
```

SYMBOL	LEN	VALUE	DEFN	REFERENCES								
A	1	00000C	157	171								
B	1	00001C	165	155								
C	1	00002C	173	163								
MEMORY	1	000000	149	148	151	153	155	157	161	163	165	167	171
NN	4	000030	174	151								
ONE	4	000038	176	167								
RESULT	4	000028	177	161	165							
TWO	4	000034	175	153	157							

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

APPENDIX C

```
                                OS/360 ASSEMBLER

LEVEL=G       RELEASE=16JUL69       SYSTEM=MFT                    TIME=14:36:34       DAY=TUESDAY       DATE=26 MAY 70

ASSEMBLER OPTIONS=ESD,RLD,LIST,LOAD,NODECK,NORENT,NOTEST,EXTIME=5,NOBATCH,UTBUFF=3,FULLXREF,INSTSET=0,
                  LINECNT=46,NOEXECUTE,SPACE=MAX-2K.
```

```
SYMBOL    TYPE  ID   ADDR    LENGTH  LD ID

          PC    01  000000  000000
ICODE     SD    02  000000  000106
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                   1          MACRO
                                   2 &B       TO    &A       TEST MDR BIT0,BRANCH TO MICRO LOCATION J IF 0.
                                   3 &B       DC    AL.4(9),AL.12((&A-MCODE)/2)
                                   4          MEND
                                   5          MACRO
                                   6 &B       T1    &A       TEST MDR BIT1,BRANCH TO MICRO LOCATION J IF 0.
                                   7 &B       DC    AL.4(10),AL.12((&A-MCODE)/2)
                                   8          MEND
                                   9          MACRO
                                  10 &B       T2    &A       TEST MDR BIT2,BRANCH TO MICRO LOCATION J IF 0.
                                  11 &B       DC    AL.4(11),AL.12((&A-MCODE)/2)
                                  12          MEND
                                  13          MACRO
                                  14 &B       TMDR  &A       TEST MDR FOR ALL OS,BRANCH IF TRUE.
                                  15 &B       DC    AL.4(12),AL.12((&A-MCODE)/2)
                                  16          MEND
                                  17          MACRO
                                  18 &B       TI    &A       BRANCH ON 5 BITS OF IR
                                  19 &B       DC    AL.4(15),AL.12((&A-MCODE)/2)
                                  20          MEND
                                  21          MACRO
                                  22 &B       TRM   &A       ALWAYS BRANCH
                                  23 &B       DC    AL.4(8),AL.12((&A-MCODE)/2)
                                  24          MEND
                                  25          MACRO
                                  26 &B       TDUMP
                                  27 &B       DC    AL.4(13),AL.12(0)
                                  28          MEND
                                  29 *        THE FOUR-ADDRESS MICRO INSTRUCTIONS
                                  30          MACRO
                                  31 &A       ADDM  &I1,&I2,&OUT,&Z
                                  32 &A       DC    AL.1(0),AL.2(&Z),AL.3(&I1),AL.3(&I2),AL.3(0),AL.4(&OUT)
                                  33          MEND
                                  34          MACRO
                                  35 &A       SUBM  &I1,&I2,&OUT,&Z
                                  36 &A       DC    AL.1(0),AL.2(&Z),AL.3(&I1),AL.3(&I2),AL.3(1),AL.4(&OUT)
                                  37          MEND
                                  38          MACRO
                                  39 &A       ANDM  &I1,&I2,&OUT,&Z
                                  40 &A       DC    AL.1(0),AL.2(&Z),AL.3(&I1),AL.3(&I2),AL.3(2),AL.4(&OUT)
                                  41          MEND
                                  42          MACRO
                                  43 &A       ORM   &I1,&I2,&OUT,&Z
                                  44 &A       DC    AL.1(0),AL.2(&Z),AL.3(&I1),AL.3(&I2),AL.3(3),AL.4(&OUT)
                                  45          MEND
                                  46          MACRO
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                  47 &A       NOT  _ &I1,&I2,&OUT,&Z
                                  48 &A       DC    AL.1(0),AL.2(&Z),AL.3(&I1),AL.3(&I2),AL.3(4),AL.4(&OUT)
                                  49          MEND
                                  50          MACRO
                                  51 &A       EOR   &I1,&I2,&OUT,&Z
                                  52 &A       DC    AL.1(0),AL.2(&Z),AL.3(&I1),AL.3(&I2),AL.3(5),AL.4(&OUT)
                                  53          MEND
                                  54          MACRO
                                  55 &A       RS    &I1,&I2,&OUT,&Z
                                  56 &A       DC    AL.1(0),AL.2(&Z),AL.3(&I1),AL.3(&I2),AL.3(6),AL.4(&OUT)
                                  57          MEND
                                  58          MACRO
                                  59 &A       LS    &I1,&I2,&OUT,&Z
                                  60 &A       DC    AL.1(0),AL.2(&Z),AL.3(&I1),AL.3(&I2),AL.3(7),AL.4(&OUT)
                                  61          MEND
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000000                            63          DS   0F                START ON FULL WORD BOUNDARY.
                                  64          ENTRY MCODE            MCODE ADDRESS FOR SIMULATOR.
000000                            65 MCODE    CSECT
                                  66          ADDM  ZERO,ZERO,CC,P   FIRST MICRO INSTRUCTION.
000000 0004                       67+         DC    AL.1(0),AL.2(P),AL.3(ZERO),AL.3(ZERO),AL.3(0),AL.4(CC)
                                  68 FETCH    ADDM  0,CC,MAR,R       FETCH NEXT INSTRUCTION FROM S MEMORY.
000002 2200                       69+FETCH    DC    AL.1(0),AL.2(R),AL.3(0),AL.3(CC),AL.3(0),AL.4(MAR)
                                  70          ADDM  FOUR,CC,CC,P     INCREMENT INSTRUCTION COUNTER.
000004 1E04                       71+         DC    AL.1(0),AL.2(P),AL.3(FOUR),AL.3(CC),AL.3(0),AL.4(CC)
                                  72          ADDM  MDR,0,IR,P       MOVE INSTRUCTION BITS TO IR.
000006 0407                       73+         DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(0),AL.3(0),AL.4(IR)
                                  74          T2    ONEADR           BRANCH IF ONE-ADDRESS INSTRUCTION.
000008 B006                       75+         DC    AL.4(11),AL.12((ONEADR-MCODE)/2)
                                  76          TI    LOC32            TEST FOR 32 ZERO-ADDRESS INSTRUCTION.
00000A F00D                       77+         DC    AL.4(15),AL.12((LOC32-MCODE)/2)
                                  78 ONEADR   ANDM  MDR,MASK,B,P     MOVE 24 ADDRESS BITS TO B REGISTER.
00000C 0525                       79+ONEADR   DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(MASK),AL.3(2),AL.4(B)
                                  80          T1    NOINDX           TEST FOR NO INDEXING.
00000E A009                       81+         DC    AL.4(10),AL.12((NOINDX-MCODE)/2)
                                  82          ADDM  X,B,B,P          INDEX ADDRESS IN B.
000010 0A85                       83+         DC    AL.1(0),AL.2(P),AL.3(X),AL.3(B),AL.3(0),AL.4(B)
                                  84 NOINDX   TO    NOINDR           TEST FOR NO INDIRECT ADDRESSING.
000012 900C                       85+NOINDX   DC    AL.4(9),AL.12((NOINDR-MCODE)/2)
                                  86          ADDM  0,B,MAR,R        ADDRESS TO MAR AND READ NEXT ADDRESS.
000014 2280                       87+         DC    AL.1(0),AL.2(R),AL.3(0),AL.3(B),AL.3(0),AL.4(MAR)
                                  88          TRM   ONEADR           RETURN TO TEST FOR MORE INDEXING.
000016 8006                       89+         DC    AL.4(8),AL.12((ONEADR-MCODE)/2)
                                  90 NOINDR   TI    LOC64            TEST FOR 32 ONE-ADDRESS INSTRUCTIONS.
000018 F01B                       91+NOINDR   DC    AL.4(15),AL.12((LOC64-MCODE)/2)
                                  92 *                               TRANSFER TO MICRO CODE FOR THE
                                  93 LOC32    TRM   SADD             S ADDITION INSTRUCTION.
00001A 802A                       94+LOC32    DC    AL.4(8),AL.12((SADD-MCODE)/2)
                                  95          TRM   SSUB             SIMILAR FOR SUBTRACTION.
00001C 802E                       96+         DC    AL.4(8),AL.12((SSUB-MCODE)/2)
                                  97          TRM   SAND             ETC.FOR EACH S INSTRUCTION.
00001E 8031                       98+         DC    AL.4(8),AL.12((SAND-MCODE)/2)
                                  99          TRM   SOR
000020 8034                      100+         DC    AL.4(8),AL.12((SOR-MCODE)/2)
                                 101          TRM   SEOR
000022 8037                      102+         DC    AL.4(8),AL.12((SEOR-MCODE)/2)
                                 103          TRM   SNOT
000024 803A                      104+         DC    AL.4(8),AL.12((SNOT-MCODE)/2)
                                 105          TRM   SXTS
000026 803C                      106+         DC    AL.4(8),AL.12((SXTS-MCODE)/2)
                                 107          TRM   SSTX
000028 8041                      108+         DC    AL.4(8),AL.12((SSTX-MCODE)/2)
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                  109          TRM   SADX
00002A 8045                       110+         DC    AL.4(8),AL.12((SADX-MCODE)/2)
                                  111          TRM   SSBX
00002C 8047                       112+         DC    AL.4(8),AL.12((SSBX-MCODE)/2)
                                  113          TRM   SRET
00002E 8049                       114+         DC    AL.4(8),AL.12((SRET-MCODE)/2)
                                  115          TRM   SPOP
000030 8042                       116+         DC    AL.4(8),AL.12((SPOP-MCODE)/2)
                                  117          TRM   STOP
000032 807A                       118+         DC    AL.4(8),AL.12((STOP-MCODE)/2)
                                  119          TRM   ILLORD
000034 8082                       120+         DC    AL.4(8),AL.12((ILLORD-MCODE)/2)
                                  121 *                            TRANSFER TO MICRO CODE FOR
                                  122 LOC64    TRM   SLOAD        ONE-ADDRESS INSTRUCTIONS.
000036 804B                       123+LOC64    DC    AL.4(8),AL.12((SLOAD-MCODE)/2)
                                  124          TRM   SLDI
000038 804D                       125+         DC    AL.4(8),AL.12((SLDI-MCODE)/2)
                                  126          TRM   SSTORE
00003A 8051                       127+         DC    AL.4(8),AL.12((SSTORE-MCODE)/2)
                                  128          TRM   STRA
00003C 8054                       129+         DC    AL.4(8),AL.12((STRA-MCODE)/2)
                                  130          TRM   STPL
00003E 8056                       131+         DC    AL.4(8),AL.12((STPL-MCODE)/2)
                                  132          TRM   STMI
000040 8059                       133+         DC    AL.4(8),AL.12((STMI-MCODE)/2)
                                  134          TRM   STZE
000042 805C                       135+         DC    AL.4(8),AL.12((STZE-MCODE)/2)
                                  136          TRM   STNZ
000044 805F                       137+         DC    AL.4(8),AL.12((STNZ-MCODE)/2)
                                  138          TRM   SENTER
000046 8062                       139+         DC    AL.4(8),AL.12((SENTER-MCODE)/2)
                                  140          TRM   SLDX
000048 8066                       141+         DC    AL.4(8),AL.12((SLDX-MCODE)/2)
                                  142          TRM   SLDXI
00004A 8068                       143+         DC    AL.4(8),AL.12((SLDXI-MCODE)/2)
                                  144          TRM   SLOOP
00004C 806A                       145+         DC    AL.4(8),AL.12((SLOOP-MCODE)/2)
                                  146          TRM   SLS
00004E 806E                       147+         DC    AL.4(8),AL.12((SLS-MCODE)/2)
                                  148          TRM   SRS
000050 8078                       149+         DC    AL.4(8),AL.12((SRS-MCODE)/2)
                                  150          TRM   ILLORD
000052 8082                       151+         DC    AL.4(8),AL.12((ILLORD-MCODE)/2)
                                  152 SADD     ADDM  FOUR,STK,MAR,R     GET SECOND LEVEL OF STACK.
000054 3F00                       153+SADD     DC    AL.1(0),AL.2(R),AL.3(FOUR),AL.3(STK),AL.3(0),AL.4(MAR)
                                  154          ADDM  MOR,A,A,P          PERFORM ADDITION.
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000056 0583                      155+          DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(A),AL.3(0),AL.4(A)
                                 156 DECR      ADDM  FOUR,STK,STK,P      CHANGE STACK POINTER.
000058 1F06                      157+DECR      DC    AL.1(0),AL.2(P),AL.3(FOUR),AL.3(STK),AL.3(0),AL.4(STK)
                                 158           TRM   FETCH               RETURN TO FETCH OF NEXT INSTRUCTION.
00005A 8001                      159+          DC    AL.4(8),AL.12((FETCH-MCODE)/2)
                                 160 SSUB      ADDM  FOUR,STK,MAR,R
00005C 3F00                      161+SSUB      DC    AL.1(0),AL.2(R),AL.3(FOUR),AL.3(STK),AL.3(0),AL.4(MAR)
                                 162           SUBM  MDR,A,A,P
00005E 0593                      163+          DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(A),AL.3(1),AL.4(A)
                                 164           TRM   DECR
000060 802C                      165+          DC    AL.4(8),AL.12((DECR-MCODE)/2)
                                 166 SAND      ADDM  FOUR,STK,MAR,R
000062 3F00                      167+SAND      DC    AL.1(0),AL.2(R),AL.3(FOUR),AL.3(STK),AL.3(0),AL.4(MAR)
                                 168           ANDM  MDR,A,A,P
000064 05A3                      169+          DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(A),AL.3(2),AL.4(A)
                                 170           TRM   DECR
000066 802C                      171+          DC    AL.4(8),AL.12((DECR-MCODE)/2)
                                 172 SOR       ADDM  FOUR,STK,MAR,R
000068 3F00                      173+SOR       DC    AL.1(0),AL.2(R),AL.3(FOUR),AL.3(STK),AL.3(0),AL.4(MAR)
                                 174           ORM   MDR,A,A,P
00006A 05B3                      175+          DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(A),AL.3(3),AL.4(A)
                                 176           TRM   DECR
00006C 802C                      177+          DC    AL.4(8),AL.12((DECR-MCODE)/2)
                                 178 SEOR      ADDM  FOUR,STK,MAR,R
00006E 3F00                      179+SEOR      DC    AL.1(0),AL.2(R),AL.3(FOUR),AL.3(STK),AL.3(0),AL.4(MAR)
                                 180           EOR   MDR,A,A,P
000070 05D3                      181+          DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(A),AL.3(5),AL.4(A)
                                 182           TRM   DECR
000072 802C                      183+          DC    AL.4(8),AL.12((DECR-MCODE)/2)
                                 184 SNOT      NOT   0,A,A,P             COMPLEMENT TOP OF STACK.
000074 01C3                      185+SNOT      DC    AL.1(0),AL.2(P),AL.3(0),AL.3(A),AL.3(4),AL.4(A)
                                 186           TRM   FETCH
000076 8001                      187+          DC    AL.4(8),AL.12((FETCH-MCODE)/2)
                                 188 SXTS      ADDM  0,STK,MAR,P
000078 0300                      189+SXTS      DC    AL.1(0),AL.2(P),AL.3(0),AL.3(STK),AL.3(0),AL.4(MAR)
                                 190           ADDM  0,A,MDR,W           WRITE TOP LEVEL INTO MEMORY.
00007A 4181                      191+          DC    AL.1(0),AL.2(W),AL.3(0),AL.3(A),AL.3(0),AL.4(MDR)
                                 192           ADDM  X,0,A,P             COPY X TO A AS NEW TOP OF STACK.
00007C 0803                      193+          DC    AL.1(0),AL.2(P),AL.3(X),AL.3(0),AL.3(0),AL.4(A)
                                 194 INCR      ADDM  MFOUR,STK,STK,P     CHANGE STACK POINTER.
00007E 0F06                      195+INCR      DC    AL.1(0),AL.2(P),AL.3(MFOUR),AL.3(STK),AL.3(0),AL.4(STK)
                                 196           TRM   FETCH
000080 8001                      197+          DC    AL.4(8),AL.12((FETCH-MCODE)/2)
                                 198 SSTX      ADDM  0,A,X,P             TOP OF STACK TO INDEX.
000082 0182                      199+SSTX      DC    AL.1(0),AL.2(P),AL.3(0),AL.3(A),AL.3(0),AL.4(X)
                                 200 SPOP      ADDM  FOUR,STK,MAR,R      GET SECOND LEVEL OF STACK.
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000084 3F00                      201+SPOP     DC    AL.1(0),AL.2(R),AL.3(FOUR),AL.3(STK),AL.3(0),AL.4(MAR)
                                 202          ADDM  MDR,0,A,P           AND COPY TO REGISTER.
000086 0403                      203+         DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(0),AL.3(0),AL.4(A)
                                 204          TRM   DECR                CHANGE STACK POINTER.
000088 802C                      205+         DC    AL.4(8),AL.12((DECR-MCODE)/2)
                                 206 SADX     ADDM  X,A,X,P             ADD STACK TO INDEX.
00008A 0982                      207+SADX     DC    AL.1(0),AL.2(P),AL.3(X),AL.3(A),AL.3(0),AL.4(X)
                                 208          TRM   SPOP                AND POP.
00008C 8042                      209+         DC    AL.4(8),AL.12((SPOP-MCODE)/2)
                                 210 SSBX     SUBM  X,A,X,P             SUBTRACT STACK FROM INDEX.
00008E 0992                      211+SSBX     DC    AL.1(0),AL.2(P),AL.3(X),AL.3(A),AL.3(1),AL.4(X)
                                 212          TRM   SPOP                AND POP.
000090 8042                      213+         DC    AL.4(8),AL.12((SPOP-MCODE)/2)
                                 214 SRET     ADDM  0,A,CC,P            GATE STACK TO CC.
000092 0184                      215+SRET     DC    AL.1(0),AL.2(P),AL.3(0),AL.3(A),AL.3(0),AL.4(CC)
                                 216          TRM   SPOP                AND POP.
000094 8042                      217+         DC    AL.4(8),AL.12((SPOP-MCODE)/2)
                                 218 SLOAD    ADDM  0,B,MAR,R           GET OPERAND.
000096 2280                      219+SLOAD    DC    AL.1(0),AL.2(R),AL.3(0),AL.3(B),AL.3(0),AL.4(MAR)
                                 220          ADDM  MDR,0,B,P           PLACE IN B.
000098 0405                      221+         DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(0),AL.3(0),AL.4(B)
                                 222 SLDI     ADDM  0,STK,MAR,P         STACK ADDRESS TO MEMORY.
00009A 0300                      223+SLDI     DC    AL.1(0),AL.2(P),AL.3(0),AL.3(STK),AL.3(0),AL.4(MAR)
                                 224          ADDM  0,A,MDR,W           WRITE TOP OF STACK.
00009C 4181                      225+         DC    AL.1(0),AL.2(W),AL.3(0),AL.3(A),AL.3(0),AL.4(MDR)
                                 226          ADDM  0,B,A,P             NEW TOP OF STACK TO A.
00009E 0283                      227+         DC    AL.1(0),AL.2(P),AL.3(0),AL.3(B),AL.3(0),AL.4(A)
                                 228          TRM   INCR                AND CHANGE STACK POINTER.
0000A0 803F                      229+         DC    AL.4(8),AL.12((INCR-MCODE)/2)
                                 230 SSTORE   ADDM  0,B,MAR,P           ADDRESS TO MEMORY.
0000A2 0280                      231+SSTORE   DC    AL.1(0),AL.2(P),AL.3(0),AL.3(B),AL.3(0),AL.4(MAR)
                                 232          ADDM  0,A,MDR,W           WRITE TOP OF STACK IN STORE LOCATION.
0000A4 4181                      233+         DC    AL.1(0),AL.2(W),AL.3(0),AL.3(A),AL.3(0),AL.4(MDR)
                                 234          TRM   SPOP                AND POP.
0000A6 8042                      235+         DC    AL.4(8),AL.12((SPOP-MCODE)/2)
                                 236 STRA     ADDM  0,B,CC,P            BRANCH ADDRESS TO CC.
0000A8 0284                      237+STRA     DC    AL.1(0),AL.2(P),AL.3(0),AL.3(B),AL.3(0),AL.4(CC)
                                 238          TRM   FETCH
0000AA 8001                      239+         DC    AL.4(8),AL.12((FETCH-MCODE)/2)
                                 240 STPL     ADDM  0,A,MDR,P           A TO MDR FOR TESTING.
0000AC 0181                      241+STPL     DC    AL.1(0),AL.2(P),AL.3(0),AL.3(A),AL.3(0),AL.4(MDR)
                                 242          TO    STRA                TRANSFER IF POSITIVE.
0000AE 9054                      243+         DC    AL.4(9),AL.12((STRA-MCODE)/2)
                                 244          TRM   FETCH
0000B0 8001                      245+         DC    AL.4(8),AL.12((FETCH-MCODE)/2)
                                 246 STMI     ADDM  0,A,MDR,P           A TO MDR FOR TESTING.
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
0000B2 0181                      247+STMI     DC    AL.1(0),AL.2(P),AL.3(0),AL.3(A),AL.3(0),AL.4(MDR)
                                 248          TO    FETCH
0000B4 9001                      249+         DC    AL.4(9),AL.12((FETCH-MCODE)/2)
                                 250          TRM   STRA                TRANSFER IF NEGATIVE.
0000B6 8054                      251+         DC    AL.4(8),AL.12((STRA-MCODE)/2)
                                 252 STZE     ADDM  0,A,MDR,P           A TO MDR FOR TESTING.
0000B8 0181                      253+STZE     DC    AL.1(0),AL.2(P),AL.3(0),AL.3(A),AL.3(0),AL.4(MDR)
                                 254          TMDR  STRA                TRANSFER IF 0.
0000BA C054                      255+         DC    AL.4(12),AL.12((STRA-MCODE)/2)
                                 256          TRM   FETCH
0000BC 8001                      257+         DC    AL.4(8),AL.12((FETCH-MCODE)/2)
                                 258 STNZ     ADDM  0,A,MDR,P
0000BE 0181                      259+STNZ     DC    AL.1(0),AL.2(P),AL.3(0),AL.3(A),AL.3(0),AL.4(MDR)
                                 260          TMDR  FETCH
0000C0 C001                      261+         DC    AL.4(12),AL.12((FETCH-MCODE)/2)
                                 262          TRM   STRA                TRANSFER IF NON-ZERO.
0000C2 8054                      263+         DC    AL.4(8),AL.12((STRA-MCODE)/2)
                                 264 SENTER   ADDM  0,B,MDR,P
0000C4 0281                      265+SENTER   DC    AL.1(0),AL.2(P),AL.3(0),AL.3(B),AL.3(0),AL.4(MDR)
                                 266          ADDM  0,CC,B,P
0000C6 0205                      267+         DC    AL.1(0),AL.2(P),AL.3(0),AL.3(CC),AL.3(0),AL.4(B)
                                 268          ADDM  MDR,0,CC,P
0000C8 0404                      269+         DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(0),AL.3(0),AL.4(CC)
                                 270          TRM   SLDI                NOW PUT B ON TOP OF STACK.
0000CA 804D                      271+         DC    AL.4(8),AL.12((SLDI-MCODE)/2)
                                 272 SLDX     ADDM  0,B,MAR,R           FETCH WORD FROM MEMORY.
0000CC 2280                      273+SLDX     DC    AL.1(0),AL.2(R),AL.3(0),AL.3(B),AL.3(0),AL.4(MAR)
                                 274          ADDM  MDR,0,B,P           AND PLACE IN B. MEMORY.
0000CE 0405                      275+         DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(0),AL.3(0),AL.4(B)
                                 276 SLDXI    ADDM  0,B,X,P             PLACE WORD IN INDEX.
0000D0 0282                      277+SLDXI    DC    AL.1(0),AL.2(P),AL.3(0),AL.3(B),AL.3(0),AL.4(X)
                                 278          TRM   FETCH
0000D2 8001                      279+         DC    AL.4(8),AL.12((FETCH-MCODE)/2)
                                 280 SLOOP    ADDM  X,FOUR,X,P          INCREMENT X BY 4.
0000D4 0882                      281+SLOOP    DC    AL.1(0),AL.2(P),AL.3(X),AL.3(FOUR),AL.3(0),AL.4(X)
                                 282          ADDM  X,0,MDR,P           MOVE X TO MDR FOR TESTING.
0000D6 0801                      283+         DC    AL.1(0),AL.2(P),AL.3(X),AL.3(0),AL.3(0),AL.4(MDR)
                                 284          TMDR  FETCH               IF 0,NO TRANSFER.
0000D8 C001                      285+         DC    AL.4(12),AL.12((FETCH-MCODE)/2)
                                 286          TRM   STRA                ELSE TRANSFER.
0000DA 8054                      287+         DC    AL.4(8),AL.12((STRA-MCODE)/2)
                                 288 SLS      ADDM  0,B,MDR,P           MOVE B TO MDR FOR TESTING.
0000DC 0281                      289+SLS      DC    AL.1(0),AL.2(P),AL.3(0),AL.3(B),AL.3(0),AL.4(MDR)
                                 290 JLS      TO    LEFT                LEFT SHIFT IF B IS POSITIVE.
0000DE 9074                      291+JLS      DC    AL.4(9),AL.12((LEFT-MCODE)/2)
                                 292 REPRS    RS    0,A,A,P             SHIFT ONE PLACE.
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
0000E0 01E3                      293+REPRS    DC    AL.1(0),AL.2(P),AL.3(0),AL.3(A),AL.3(6),AL.4(A)
                                 294          ADDM  MDR,ONE,MDR,P       DECREASE COUNT.
0000E2 0481                      295+         DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(ONE),AL.3(0),AL.4(MDR)
                                 296          TMDR  FETCH               FINISHED SHIFTING.
0000E4 C001                      297+         DC    AL.4(12),AL.12((FETCH-MCODE)/2)
                                 298          TRM   REPRS               REPEAT SHIFTING.
0000E6 8070                      299+         DC    AL.4(8),AL.12((REPRS-MCODE)/2)
                                 300 LEFT     TMDR  FETCH               ZERO SHIFTS LEFT.
0000E8 C001                      301+LEFT     DC    AL.4(12),AL.12((FETCH-MCODE)/2)
                                 302          LS    0,A,A,P             SHIFT ONE PLACE.
0000EA 01F3                      303+         DC    AL.1(0),AL.2(P),AL.3(0),AL.3(A),AL.3(7),AL.4(A)
                                 304          SUBM  MDR,ONE,MDR,P       DECREASE COUNT.
0000EC 0491                      305+         DC    AL.1(0),AL.2(P),AL.3(MDR),AL.3(ONE),AL.3(1),AL.4(MDR)
                                 306          TRM   LEFT
0000EE 8074                      307+         DC    AL.4(8),AL.12((LEFT-MCODE)/2)
                                 308 SRS      SUBM  0,B,MDR,P           MINUS COUNT TO MDR.
0000F0 0291                      309+SRS      DC    AL.1(0),AL.2(P),AL.3(0),AL.3(B),AL.3(1),AL.4(MDR)
                                 310          TRM   JLS                 JOIN LEFT-SHIFT PROGRAM.
0000F2 806F                      311+         DC    AL.4(8),AL.12((JLS-MCODE)/2)
                                 312 STOP     ADDM  0,0,A,P
0000F4 0003                      313+STOP     DC    AL.1(0),AL.2(P),AL.3(0),AL.3(0),AL.3(0),AL.4(A)
                                 314          ADDM  0,0,B,P
0000F6 0005                      315+         DC    AL.1(0),AL.2(P),AL.3(0),AL.3(0),AL.3(0),AL.4(B)
                                 316          ADDM  0,0,CC,P
0000F8 0004                      317+         DC    AL.1(0),AL.2(P),AL.3(0),AL.3(0),AL.3(0),AL.4(CC)
                                 318          ADDM  0,0,STK,P
0000FA 0006                      319+         DC    AL.1(0),AL.2(P),AL.3(0),AL.3(0),AL.3(0),AL.4(STK)
                                 320          ADDM  0,0,X,P
0000FC 0002                      321+         DC    AL.1(0),AL.2(P),AL.3(0),AL.3(0),AL.3(0),AL.4(X)
                                 322          ADDM  0,0,IR,P
0000FE 0007                      323+         DC    AL.1(0),AL.2(P),AL.3(0),AL.3(0),AL.3(0),AL.4(IR)
                                 324          ADDM  0,0,MDR,P
000100 0001                      325+         DC    AL.1(0),AL.2(P),AL.3(0),AL.3(0),AL.3(0),AL.4(MDR)
                                 326          ADDM  0,0,MAR,P
000102 0000                      327+         DC    AL.1(0),AL.2(P),AL.3(0),AL.3(0),AL.3(0),AL.4(MAR)
000104                           328          DS    0H
                                 329 ILLORD   TDUMP
000104 D000                      330+ILLORD   DC    AL.4(13),AL.12(0)
                                 331 *********EQU DEFINITIONS OF S-REGISTER NAMES AS NUMBERS.
000000                           332          CSECT
                                 333 *        OUTPUT BUS
000000                           334 MAR      EQU   0
000001                           335 MDR      EQU   1
000002                           336 X        EQU   2
000003                           337 A        EQU   3
000004                           338 CC       EQU   4
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000005                            339 B        EQU  5
000006                            340 STK      EQU  6
000007                            341 IR       EQU  7
                                  342 *        MEMORY CONTROL
000000                            343 P        EQU  0
000001                            344 R        EQU  1
000002                            345 W        EQU  2
000000                            346 ZERO     EQU  0
000007                            347 FOUR     EQU  7
000002                            348 MASK     EQU  2
000003                            349 MFOUR    EQU  3
000001                            350 ONE      EQU  1
                                  351          END
```

26 MAY 70

SYMBOL	LEN	VALUE	DEFN	REFERENCES													
A	1	000003	337	155	155	163	163	169	169	175	175	181	181	185	185	191	193
				199	203	207	211	215	225	227	233	241	247	253	259	293	293
				303	303	313											
B	1	000005	339	79	83	83	87	219	221	227	231	237	265	267	273	275	277
				289	309	315											
CC	1	000004	338	67	69	71	71	215	237	267	269	317					
DECR	1	000058	157	165	171	177	183	205									
FETCH	1	000002	69	159	187	197	239	245	249	257	261	279	285	297	301		
FOUR	1	000007	347	71	153	157	161	167	173	179	201	281					
ILLORD	1	000104	330	120	151												
INCR	1	00007E	195	229													
IR	1	000007	341	73	323												
JLS	1	0000DE	291	311													
LEFT	1	0000E8	301	291	307												
LOC32	1	00001A	94	77													
LOC64	1	000036	123	91													
MAR	1	000000	334	69	87	153	161	167	173	179	189	201	219	223	231	273	327
MASK	1	000002	348	79													
MCODE	1	000000	65	64	75	77	81	85	89	91	94	96	98	100	102	104	106
				108	110	112	114	116	118	120	123	125	127	129	131	133	135
				137	139	141	143	145	147	149	151	159	165	171	177	183	187
				197	205	209	213	217	229	235	239	243	245	249	251	255	257
				261	263	271	279	285	287	291	297	299	301	307	311		
MDR	1	000001	335	73	79	155	163	169	175	181	191	203	221	225	233	241	247
				253	259	265	269	275	283	289	295	295	305	305	309	325	
MFOUR	1	000003	349	195													
NOINDR	1	000018	91	85													
NOINCX	1	000012	85	81													
ONE	1	000001	350	295	305												
ONEADR	1	00000C	79	75	89												
P	1	000000	343	67	71	73	79	83	155	157	163	169	175	181	185	189	193
				195	199	203	207	211	215	221	223	227	231	237	241	247	253
				259	265	267	269	275	277	281	283	289	293	295	303	305	309
				313	315	317	319	321	323	325	327						
R	1	000001	344	69	87	153	161	167	173	179	201	219	273				
REPRS	1	0000E0	293	299													
SADD	1	000054	153	94													
SADX	1	00008A	207	110													
SAND	1	000062	167	98													
SENTER	1	0000C4	265	139													
SEOR	1	00006E	179	102													
SLDI	1	00009A	223	125	271												
SLDX	1	0000CC	273	141													
SLDXI	1	0000D0	277	143													
SLOAD	1	000096	219	123													
SLOOP	1	000004	281	145													

SYMBOL	LEN	VALUE	DEFN	REFERENCES											
SLS	1	0000DC	289	147											
SNOT	1	000074	185	104											
SOR	1	000068	173	100											
SPOP	1	000084	201	116	209	213	217	235							
SRET	1	000092	215	114											
SRS	1	0000F0	309	149											
SSBX	1	00008E	211	112											
SSTORE	1	0000A2	231	127											
SSTX	1	000082	199	108											
SSUB	1	00005C	161	96											
STK	1	000006	340	153	157	157	161	167	173	179	189	195	195	201	223	319
STMI	1	0000B2	247	133											
STNZ	1	0000BE	259	137											
STOP	1	0000F4	313	118											
STPL	1	0000AC	241	131											
STRA	1	0000A8	237	129	243	251	255	263	287						
STZE	1	0000B8	253	135											
SXTS	1	000078	189	106											
W	1	000002	345	191	225	233									
X	1	000002	336	83	193	199	207	207	211	211	277	281	281	283	321	
ZERO	1	000000	346	67	67											

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

APPENDIX D

OS/360 ASSEMBLER

LEVEL=G RELEASE=16JUL69 SYSTEM=MFT TIME=14:37:47 DAY=TUESDAY DATE=26 MAY 70

```
ASSEMBLER OPTIONS=ESD,RLD,LIST,LOAD,NODECK,NORENT,NOTEST,EXTIME=5,NOBATCH,UTBUFF=3,FULLXREF,INSTSET=0,
                  LINECNT=46,NOEXECUTE,SPACE=MAX-2K.
```

SYMBOL	TYPE	ID	ADDR	LENGTH	LD ID
BEGINLPX	SD	01	000000	000044	
SIMULATR	ER	02			

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     1 BEGINLPX PRIME
000000                               2+BEGINLPX START 0
000000 47F0 F00E             0000E   3+         B     14(0,15) BRANCH AROUND ID
000004 08                            4+         DC    AL1(8)
000005 C2C5C7C9D5D3D7E7              5+         DC    CL8'BEGINLPX' IDENTIFIER
00000D 00
00000E 90EC D00C             0000C   6+         STM   14,12,12(13) SAVE REGISTERS
000012 05C0                          7+         BALR  12,0 CREATE BASE REGISTER
000014                               8+         USING *,12
000014 1830                          9+         LR    3,0 SAVE -0-
000016 1821                         10+         LR    2,1 SAVE PARAMETER REGISTER
000018                              11+         CNOP  0,4
000018 4510 C00C             00020  12+         BAL   1,*+8 BRANCH AROUND LENGTH
00001C 00000048                     13+         DC    A(72) LENGTH
000020 5801 0000             00000  14+         L     0,0(1,0) LOAD LENGTH
000024 0A0A                         15+         SVC   10 ISSUE GETMAIN SVC
000026 501D 0008             00008  16+         ST    1,8(13) STORE FORWARD POINTER
00002A 50D1 0004             00004  17+         ST    13,4(1) STORE BACKWARD POINTER
00002E 18D1                         18+         LR    13,1 SET UP REGISTER FOR SAVE AREA
000030 1812                         19+         LR    1,2 RESTORE PARAMETER REGISTER
000032 1803                         20+         LR    0,3 RESTORE -0-
000034 58F0 C02C             00040  21          L     15,=V(SIMULATR)
000038 05EF                         22          BALR  14,15
00003A 47F0 0000             00000  23          B     0
                                    24          END
000040 00000000                     25                =V(SIMULATR)
```

26 MAY 70

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BEGINLPX	1	000000	2	

26 MAY 70

POS.ID	REL.ID	FLAGS	ADDRESS
01	02	1C	000040

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

```
F128-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,MAP,LET
          VARIABLE OPTIONS USED - SIZE=(131072,28672)              DEFAULT OPTION(S) USED
IEW0000      INCLUDE  SYSLIB(DRIVER1)
IEW0000      ENTRY  DRIVER1


                                                  MODULE MAP

  CONTROL SECTION                        ENTRY

    NAME     ORIGIN  LENGTH              NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION

  SIMULATR       00     5BD
  MEMORY        5C0    103C
  MCCDE        1600     106
  BEGINLPX     1708      44
  DRIVER1      1750     2D8
  HEXCON1 *    1A28     160
  HEXCON2 *    1B88     150
  PRINTBUF*    1CD8     2D8
 $PRIVATE      1FB0      00


 ENTRY ADDRESS      1750
 TOTAL LENGTH       1FB0

****GO         DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET
```

A	B	CC	STK	X	IR	MAR	MDR	MCC
00000026	27003900	00000000	0000F5DC	2F003010	26270039	00072627	00092627	0000F620
00000026	27003900	00000000	0000F5DC	2F003010	26270039	00000000	00000030	0000F622
00000026	27003900	00000004	0000F5DC	2F003010	26270039	00000000	00000030	0000F624
0C000026	27003900	00000004	0000F5DC	2F003010	00000030	00000000	00000030	0000F626
00000026	27003900	00000004	0000F5DC	2F003010	00000030	00000000	00000030	0000F628
0C000026	00000030	00000004	0000F5DC	2F003010	00000030	00000000	00000030	0000F62C
0C000026	00000030	00000004	0000F5DC	2F003010	00000030	00000000	00000030	0000F62E
00000026	00000030	00000004	0000F5DC	2F003010	00000030	00000000	00000030	0000F632
0C000026	00000030	00000004	0000F5DC	2F003010	00000030	00000000	00000030	0000F638
0C000026	00000030	00000004	0000F5DC	2F003010	00000030	00000000	00000030	0000F656
00000026	00000030	00000004	0000F5DC	2F003010	00000030	00000030	00000005	0000F6B6
0C000026	00000005	00000004	0000F5DC	2F003010	00000030	00000030	00000005	0000F6B8
0C000026	00000005	00000004	0000F5DC	2F003010	00000030	0000F5DC	00000005	0000F6BA
00000026	00000005	00000004	0000F5DC	2F003010	00000030	0000F5DC	00000026	0000F6BC
00000005	00000005	00000004	0000F5DC	2F003010	00000030	0000F5DC	00000026	0000F6BE
00000005	00000005	00000004	0000F5DC	2F003010	00000030	0000F5DC	00000026	0000F6C0
00000005	00000005	00000004	0000F5D8	2F003010	00000030	0000F5DC	00000026	0000F69E
0C000005	00000005	00000004	0000F5D8	2F003010	00000030	0000F5DC	00000026	0000F6A0
00000005	00000005	00000004	0000F5D8	2F003010	00000030	00000004	00000034	0000F622
00000005	00000005	00000008	0000F5D8	2F003010	00000030	00000004	00000034	0000F624
00000005	00000005	00000008	0000F5D8	2F003010	00000034	00000004	00000034	0000F626
00000005	00000005	00000008	0000F5D8	2F003010	00000034	00000004	00000034	0000F628
0C000005	00000034	00000008	0000F5D8	2F003010	00000034	00000004	00000034	0000F62C
00000005	00000034	00000008	0000F5D8	2F003010	00000034	00000004	00000034	0000F62E
00000005	00000034	00000008	0000F5D8	2F003010	00000034	00000004	00000034	0000F632
00000005	00000034	00000008	0000F5D8	2F003010	00000034	00000004	00000034	0000F638
00000005	00000034	00000008	0000F5D8	2F003010	00000034	00000004	00000034	0000F656
0C000005	00000034	00000008	0000F5D8	2F003010	00000034	00000034	00000002	0000F6B6
00000005	00000002	00000008	0000F5D8	2F003010	00000034	00000034	00000002	0000F6B8
00000005	00000002	00000008	0000F5D8	2F003010	00000034	0000F5D8	00000002	0000F6BA
00000005	00000002	00000008	0000F5D8	2F003010	00000034	0000F5D8	00000005	0000F6BC
00000002	00000002	00000008	0000F5D8	2F003010	00000034	0000F5D8	00000005	0000F6BE
00000002	00000002	00000008	0000F5D8	2F003010	00000034	0000F5D8	00000005	0000F6C0
00000002	00000002	00000008	0000F5D4	2F003010	00000034	0000F5D8	00000005	0000F69E
0C000002	00000002	00000008	0000F5D4	2F003010	00000034	0000F5D8	00000005	0000F6A0
0C000002	00000002	0000000C	0000F5D4	2F003010	00000034	00000008	0300001C	0000F622
00000002	00000002	0000000C	0000F5D4	2F003010	00000034	00000008	0300001C	0000F624
00000002	00000002	0000000C	0000F5D4	2F003010	0300001C	00000008	0300001C	0000F626
00000002	00000002	0000000C	0000F5D4	2F003010	0300001C	00000008	0300001C	0000F628
00000002	0000001C	0000000C	0000F5D4	2F003010	0300001C	00000008	0300001C	0000F62C
00000002	0000001C	0000000C	0000F5D4	2F003010	0300001C	00000008	0300001C	0000F62E
00000002	0000001C	0000000C	0000F5D4	2F003010	03C0001C	00000008	0300001C	0000F632
00000002	0000001C	0000000C	0000F5D4	2F003010	0300001C	00000008	0300001C	0000F638
00000002	0000001C	0000000C	0000F5D4	2F003010	0300001C	00000008	0300001C	0000F65C
00000002	0000001C	0000001C	0000F5D4	2F003010	0300001C	00000008	0300001C	0000F6C8
00000002	0000001C	0000001C	0000F5D4	2F003010	0300001C	00000008	0300001C	0000F6CA

```
00000002 0000001C 0000001C 0000F5D4 2F003010 0300001C 0000001C 0200003C 0000F622
00000002 0000001C 00000020 0000F5D4 2F003010 0300001C 0000001C 0200003C 0000F624
00000002 0000001C 00000020 0000F5D4 2F003010 0200003C 0000001C 0200003C 0000F626
00000002 0000001C 00000020 0000F5D4 2F003010 0200003C 0000001C 0200003C 0000F628
00000002 0000003C 00000020 000QF5D4 2F003010 0200003C 0000001C 0200003C 0000F62C
00000002 0000003C 00000020 0000F5D4 2F003010 0200003C 0000001C 0200003C 0000F62E
00000002 0000003C 00000020 0000F5D4 2F003010 0200003C 0000001C 0200003C 0000F632
00000002 0000003C 00000020 0000F5D4 2F003010 0200003C 0000001C 0200003C 0000F638
00000002 0000003C 00000020 0000F5D4 2F003010 0200003C 0000001C 0200003C 0000F65A
00000002 0000003C 000C0020 0000F5D4 2F003010 0200003C 0000003C 0200003C 0000F6C2
00000002 0000003C 00000020 0000F5D4 2F003010 0200003C 0000003C 00000002 0000F6C4
00000002 0000003C 00000020 0000F5D4 2F003010 0200003C 0000003C 00000002 0000F6C6
00000002 0000003C 00000020 0000F5D4 2F003010 0200003C 0000F5D8 00000005 0000F6A4
00000005 0000003C 00000020 C000F5D4 2F003010 0200003C 0000F5D8 00000005 0000F6A6
00000005 0000003C 00000020 0000F5D4 2F003010 0200003C 0000F5D8 00000005 0000F6A8
00000005 0000003C 00000020 0000F5D8 2F003010 0200003C 0000F5D8 00000005 0000F678
00000005 0000003C 00000020 0000F5D8 2F003010 0200003C 0000F5D8 00000005 0000F67A
00000005 0000003C 00000020 0000F5D8 2F003010 0200003C 00000020 00000038 00C0F622
00000005 0000003C 00000024 0000F5D8 2F003010 0200003C 00000020 00000038 0000F624
00000005 0000003C 00000024 0000F5D8 2F003010 00000038 00000020 00000038 0000F626
00000005 0000003C 00000024 0000F5D8 2F003010 00000038 00000020 00000038 0000F628
0C000005 00000038 00000024 0000F5D8 2F003010 00000038 00000020 00000038 0000F62C
0C000005 00000038 000C0024 0000F5D8 2F003010 00000038 00000020 00000038 0000F62E
00000005 00000038 00000024 0000F5D8 2F003010 00000038 00000020 00000038 0000F632
00000005 00000038 00000024 0000F5D8 2F003010 00000038 00000020 00000038 0000F638
0C000005 00000038 00000024 0000F5D8 2F003010 00000038 00000020 00000038 0000F656
00000005 00000038 00000024 0000F5D8 2F003010 00000038 00000038 00000001 0000F6B6
00000005 00000001 00000024 0000F5D8 2F003010 00000038 00000038 00C00001 0000F6B8
0C000005 00000001 00000024 0000F5D8 2F003010 00000038 0000F5D8 00000001 0000F6BA
00000005 00000001 00000024 0000F5D8 2F003010 00000038 0000F5D8 00000005 0000F6BC
00000001 00000001 00000024 0000F5D8 2F003010 00000038 0000F5D8 00000005 0000F6BE
0C000001 00000001 00000024 0000F5D8 2F003010 00000038 0000F5D8 00000005 0000F6C0
00000001 00000001 00000024 0000F5D4 2F003010 00000038 0000F5D8 00000005 0C00F69E
CC000001 00000001 00000024 0000F5D4 2F003010 00000038 0000F5D8 00000005 0000F6A0
0C000001 00000001 00000024 0000F5D4 2F003010 00000038 00000024 21000000 0000F622
00000001 00000001 00000028 0000F5D4 2F003010 00000038 00000024 21000000 0000F624
00000001 00000001 00000028 0000F5D4 2F003010 21000000 00000024 21000000 0000F626
00000001 00000001 00000028 0000F5D4 2F003010 21000000 00000024 21000000 0000F628
00000001 00000001 00000028 0000F5D4 2F003010 21000000 00000024 21000000 0000F62A
00000001 00000001 00000028 0000F5D4 2F003010 21000000 00000024 21000000 0000F63C
00000001 00000001 00000028 0000F5D4 2F003010 21000000 0000F5D8 00000005 0000F67C
0C000004 00000001 00C00028 0000F5D4 2F003010 21000000 0000F5D8 00000005 0000F67E
00000004 00000001 00000028 0000F5D4 2F003010 21000000 0000F5D8 00000005 0000F680
00000004 00000001 00000028 0000F5D8 2F003010 21000000 0000F5D8 00000005 0000F678
00000004 00000001 00C00028 0000F5D8 2F003010 21000000 0000F5D8 0C000005 0000F67A
00000004 00000001 00000028 0000F5D8 2F003010 21000000 00000028 0700000C 0000F622
```

```
00000004 00000001 0000002C 0000F5D8 2F003010 21000000 00000028 0700000C 0000F624
00000004 00000001 0000002C 0000F5D8 2F003010 0700000C 00000028 0700000C 0000F626
00000004 00000001 0000002C 0000F5D8 2F003010 0700000C 00000028 0700000C 0000F628
00000004 0000000C 0000002C 0000F5D8 2F003010 0700000C 00000028 0700000C 0000F62C
00000004 0000000C 0000002C 0000F5D8 2F003010 0700000C 00000028 0700000C 0000F62E
00000004 0000000C 0000002C 000ðF5D8 2F003010 0700000C 00000028 0700000C 0000F632
00000004 0000000C 0000002C 0000F5D8 2F003010 0700000C 00000028 0700000C 0000F638
CC000004 0000000C 0000002C 0000F5D8 2F003010 0700000C 00000028 0700000C 0000F664
00000004 0000000C 0000002C 0000F5D8 2F003010 0700000C 00000028 00000004 0000F6DE
CC000004 0000000C 0000002C 0000F5D8 2F003010 0700000C 00000028 00000004 0000F6E0
00000004 0000000C 0000002C 0000F5D8 2F003010 0700000C 00000028 00000004 0000F6E2
00000004 0000000C 0000000C 0000F5D8 2F003010 0700000C 00000028 00000004 0000F6C8
00000004 0000000C 0000000C 0000F5D8 2F003010 0700000C 00000028 00000004 0000F6CA
00000004 0000000C 0000000C 0000F5D8 2F003010 0700000C 0000000C 00000034 0000F622
00000004 0000000C 00000010 0000F5D8 2F003010 0700000C 0000000C 00000034 0000F624
0C000004 0000000C 00000010 0000F5D8 2F003010 00000034 0000000C 00000034 0000F626
0C000004 0000000C 000C0010 0000F5D8 2F003010 00000034 0000000C 00000034 0000F628
00000004 00000034 00000010 0000F5D8 2F003010 00000034 0000000C 00000034 0000F62C
00000004 00000034 00000010 0000F5D8 2F003010 00000034 0000000C 00000034 0000F62E
00000004 00000034 00000010 0000F5D8 2F003010 00000034 0000000C 00000034 0000F632
0C000004 00000034 00000010 0000F5D8 2F003010 00000034 0000000C 00000034 0000F638
00000004 00000034 00000010 0000F5D8 2F003010 00000034 0000000C 00000034 0000F656
00000004 00000034 00000010 0000F5D8 2F003010 00000034 00000034 00000002 0000F6B6
00000004 00000002 00000010 0000F5D8 2F003010 00000034 00000034 00000002 0000F6B8
00000004 00000002 00000010 0000F5D8 2F003010 00000034 0000F5D8 00000002 0000F6BA
CC000004 00000002 00000010 0000F5D8 2F003010 00000034 0000F5D8 00000004 0000F6BC
00000002 00000002 000C0010 0000F5D8 2F003010 00000034 0000F5D8 00000004 0000F6BE
00000002 00000002 00000010 0000F5D8 2F003010 00000034 0000F5D8 00000004 0000F6C0
00000002 00000002 00000010 0000F5D4 2F003010 00000034 0000F5D8 00000004 0000F69E
00000002 00000002 00000010 0000F5D4 2F003010 00000034 0000F5D8 00000004 0000F6A0
00000002 00000002 00000010 0000F5D4 2F003010 00000034 00000010 20000000 0C00F622
0C000002 00000002 00000014 0000F5D4 2F003010 00000034 00000010 20000000 0000F624
00000002 00000002 00000014 0000F5D4 2F003010 20000000 00000010 20000000 0000F626
00000002 00000002 00000014 0000F5D4 2F003010 20000000 00000010 20000000 0000F628
00000002 00000002 00000014 0000F5D4 2F003010 20000000 00000010 20000000 0000F62A
00000002 00000002 00000014 0000F5D4 2F003010 20000000 00000010 20000000 0000F63A
00000002 00000002 00000014 0000F5D4 2F003010 20000000 0000F5D8 00000004 0000F674
00000006 00000002 00000014 0000F5D4 2F003010 20000000 0000F5D8 00000004 0000F676
0C000006 00000002 00000014 0000F5D8 2F003010 20000000 0000F5D8 00000004 0000F678
00000006 00000002 00000014 0000F5D8 2F003010 20000000 0000F5D8 00000004 0000F67A
00000006 00000002 00000014 0000F5D8 2F003010 20000000 00000014 0200003C 0000F622
0C000006 00000002 00000018 0000F5D8 2F003010 20000000 00000014 0200003C 0000F624
00000006 00000002 00000018 0000F5D8 2F003010 0200003C 00000014 0200003C 0000F626
00000006 00000002 00000018 0000F5D8 2F003010 0200003C 00000014 0200003C 0000F628
00000006 0000003C 00000018 0000F5D8 2F003010 0200003C 00000014 0200003C 0000F62C
00000006 0000003C 000C0018 0000F5D8 2F003010 0200003C 00000014 0200003C 0000F62E
```

```
00000006 0000003C 00000018 0000F5D8 2F003010 0200003C 00000014 0200003C 0000F632
00000006 0000003C 00000018 0000F5D8 2F003010 0200003C 00000014 0200003C 0000F638
0C000006 0000003C 00000018 0000F5D8 2F003010 0200003C 00000014 0200003C 0000F65A
00000006 0000003C 00000018 0000F5D8 2F003010 0200003C 0000003C 0200003C 0000F6C2
00000006 0000003C 00000018 0000F5D8 2F003010 0200003C 0000003C 00000006 0000F6C4
0C000006 0000003C 00000018 0000F5D8 2F003010 0200003C 0000003C 00000006 0000F6C6
0C000006 0000003C 00000018 0000F5D8 2F003010 0200003C 0000F5DC 00000026 0000F6A4
00000026 0000003C 00000018 0000F5D8 2F003010 0200003C 0000F5DC 00000026 0000F6A6
00000026 0000003C 00000018 0000F5D8 2F003010 0200003C 0000F5DC 00000026 0000F6A8
0C000026 0000003C 00000018 0000F5DC 2F003010 0200003C 0000F5DC 00000026 0000F67B
00000026 0000003C 00000018 0000F5DC 2F003010 0200003C 0C00F5DC 00000026 0000F67A
00000026 0000003C 00000018 0000F5DC 2F003010 0200003C 00000018 0300002C 0000F622
CC000026 0000003C 0000001C 0000F5DC 2F003010 0200003C 00000018 0300002C 0000F624
CC000026 0000003C 0000001C 0000F5DC 2F003010 0300002C 00000018 0300002C 0000F626
00000026 0000003C 0000001C 0000F5DC 2F003010 0300002C 00000018 0300002C 0000F628
00000026 0000002C 0000001C 0000F5DC 2F003010 0300002C 00000018 0300002C 0000F62C
00000026 0000002C 0000001C 0000F5DC 2F003010 0300002C 00000018 0300002C 0000F62E
00000026 0000002C 0000001C 0000F5DC 2F003010 0300002C 00000018 0300002C 0000F632
0C000026 0000002C 0000001C 0000F5DC 2F003010 0300002C 00000018 0300002C 0000F638
00000026 0000002C 0000001C 0000F5DC 2F003010 03000C2C 00000018 0300002C 0000F65C
00000026 0000002C 0000002C 0000F5DC 2F003010 0300002C 00000018 0300002C 0000F6C8
00000026 0000002C 0000002C 0000F5DC 2F003010 0300002C 00000018 0300002C 0000F6CA
00000026 0000002C 0000002C 0000F5DC 2F003010 0300002C 0000002C 2C000000 0000F622
00000026 0000002C 00000030 0000F5DC 2F003010 0300002C 0000002C 2C000000 0000F624
00000026 0000002C 00000030 0000F5DC 2F003010 2C000000 0000002C 2C000000 0000F626
00000026 0000002C 00000030 0000F5DC 2F003010 2C000000 0000002C 2C000000 0000F628
00000026 0000002C 00000030 0000F5DC 2F003010 2C000000 0000002C 2C000000 0000F62A
00000026 0000002C 00000030 0000F5DC 2F003010 2C000000 0000002C 2C000000 0000F652
00000000 0000002C 00000030 0000F5DC 2F003010 2C000000 0000002C 2C000000 0000F714
00000000 00000000 00000030 0000F5DC 2F003010 2C000000 0000002C 2C000000 0000F716
00000000 00000000 00000000 0000F5DC 2F003010 2C000000 0000002C 2C000000 0000F718
00000000 00000000 00000000 00000000 2F003010 2C000000 0000002C 2C000000 0000F71A
00000000 00000000 00000000 00000000 00000000 2C000000 0000002C 2C000000 0000F71C
00000000 00000000 00000000 00000000 00000000 00000000 0000002C 2C000000 0000F71E
00000000 00000000 00000000 00000000 00000000 00000000 0000002C 00000000 0000F720
0C000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000F722
```

SIMULATION OF THE MICRO-PROGRAM OF THE S-MACHINE

by

JOE MING-HWA TIAO

B. A., Toyo University, Tokyo, Japan, 1964

M. A., Aoyama University, Tokyo, Japan, 1967

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Statistics and Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1970

It was the purpose of this study to introduce a portion of the S-machine language and to demonstrate how microprogramming techniques can be used to simulate this computer system.

The objectives of this study were:

(1) To implement and test a computer program which would simulate the S-machine language.

(2) To write and test programs in the assembler language of the S-machine, which had been implemented by using the macro facilities of the OS/360 assembler language.

(3) To use the interpretation of the micro-program to simulate the execution of the S-machine instructions.

(4) To write a simulator program which would act as an interpreter, to execute the micro-program, where the micro-program describes the action of the S-machine instructions.

The simulation of the S-machine was accomplished by means of a computer program divided into three separate segments; 1) the S-machine instruction segment 2) the micro-program segment and 3) the simulator segment. The macro-facilities of the IBM OS/360 assembler were utilized to assemble the three segments and when the segments were linked together at run time they simulated the S-machine.

The results of this study showed that when each S-machine instruction was executed the simulator interpreted the micro-program properly and the output was made by the macro assembler to trace and print out the contents of all registers in the data flow of the S-machine.