A PROPOSED PROGRAMMING SYSTEM FOR KNUTH'S MIX COMPUTER

by

Max Neil Akers

United States
Naval Postgraduate School

THESIS

A PROPOSED PROGRAMMING SYSTEM

FOR

KNUTH'S MIX COMPUTER

BY

Max Neil Akers

June 1969

A Proposed Programming System

for

Knuth's MIX Computer

by

Max Neil Akers
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1958

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1969

ABSTRACT

A programming system using a hypothetical computer is proposed for use in teaching machine and assembly language programming courses. Major components such as monitor, assembler, interpreter, grader and diagnostics are described. The interpreter is programmed and documented for use on an IBM 360/67. The interpreter can be used for teaching machine language programming and can be incorporated into the proposed programming system.

TABLE OF CONTENTS

I. INTRODUCTION---------- 7

II. THE MIX PROGRAMMING SYSTEM---------- 9

A. CHOICE OF THE COMPUTER---------- 9

B. GENERAL OBJECTIVES---------- 9

C. SYSTEM COMPONENTS AND FUNCTIONS---------- 10

1. System Monitor---------- 12

2. Mix Simulator---------- 14

3. Mix Assembler---------- 15

4. Mix Grader---------- 16

5. Diagnostic Facilities---------- 17

III. THE MIX SIMULATOR---------- 19

A. INTERPRETATIVE ROUTINES---------- 19

B. DESCRIPTION AND STRUCTURE---------- 20

C. PROGRAMMING TECHNIQUES---------- 22

IV. CONCLUSIONS AND RECOMMENDATIONS---------- 23

APPENDIX A. USERS MANUAL---------- 24

APPENDIX B. PROGRAM FLOWCHARTS---------- 30

APPENDIX C. PROGRAM LISTING---------- 63

BIBLIOGRAPHY---------- 104

INITIAL DISTRIBUTION LIST---------- 105

FORM DD 1473 ----------107

LIST OF FIGURES

1. Global Organization of Mix Programming System---------- 11

I. INTRODUCTION

The development of programming skill is generally not the primary goal of a graduate or undergraduate program in computer science. However, it is generally recognized as an important secondary goal in which students should attain a reasonable level of proficiency. This is usually accomplished by requiring computer work in required courses and thesis work or involvement in special project courses. At least one widely recognized authority, D. W. Hamming, would require every computer science major to design, build, debug, and document a reasonably sized program such as a simulator or simplified compiler[1].

In attempting to teach an assembly language, professors are faced with a common problem, namely, the assembly language and software configuration of the computer facility available at their school is often too complex to be used effectively as a teaching aid. The problem is equally critical for the beginning student or novice programmer who is just beginning to develop the formal concepts of a stored program computer. The assembly language is sufficiently complex that he cannot even begin to understand the great complexities of the operating system which stands between him and the device which he has set out to master. Getting even the most simple form of input or output smacks of mysticism and usually evolves to the process of blind faith in an instructor provided recipe.

One technique available for handling this problem is to choose a computer, either hypothetical or actual, that can be used as an effective teaching aid. A program is written in one of the

programming languages available on the existing computer facility which will then accept and execute programs written in the machine or assembly language of the chosen instructional computer.

This thesis chooses a computer, proposes the design for a student programming system and provides a simulator for the IBM 360/67 which accepts and executes programs written in the chosen machine language.

II. THE MIX PROGRAMMING SYSTEM

A. CHOICE OF THE COMPUTER

Choosing an appropriate computer with its associated machine and assembly languages for instructional purposes is not a particularly difficult task but as usual there are compromises in any choice. An existing machine could have been chosen, but most existing machines have idiosyncrasies which are of no value for instruction in assembly languages. Although there may be other machines which are equally good, the hypothetical computer MIX [2] was chosen for the following reasons:

1. The basic structure of the machine and assembly language is simple and straight forward so that its operation may be easily learned.
2. The language is powerful enough to allow most algorithms to be briefly expressed.
3. MIX is not too difficult to simulate on most existing computers.
4. The MIX computer is simple enough in concept to permit the student to maintain some sense of contact with what is happening to his program.

B. GENERAL OBJECTIVES

In developing a MIX Programming System based on the MIX computer the basic philosophy is to remove advanced concepts such as CSECT, DSECT, and JCL from the tutorial realm of teaching basic courses in assembly language programming. Such a philosophy will allow the MIX Programming System to be used in introductory courses

while at the same time providing an environment in which a student in a hard core assembly language course can participate in a systematic development of those very concepts which have made possible today's sophisticated systems.

Functionally, the MIX Programming System must meet the following minimum requirements:

1. Accept source programs written in the MIX Assembly Language (MIXAL) and emit MIX machine language which can be interpretatively executed.

2. Accept source programs written in the MIX machine language and interpretatively execute them.

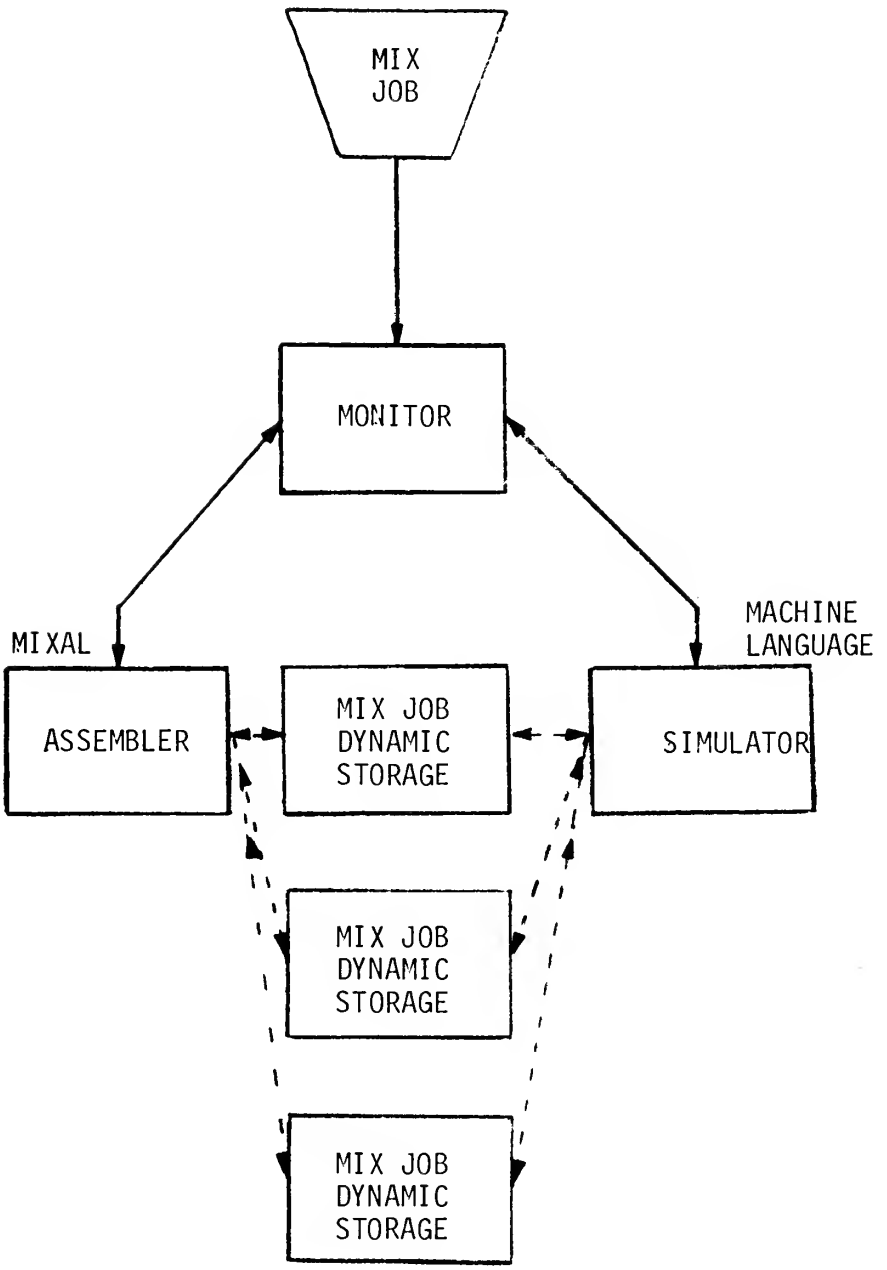
3. Provide the necessary program facilities to load and execute MIX programs.

4. Provide facilities for monitoring student use (or misuse) of the system.

C. SYSTEM COMPONENTS AND FUNCTIONS

The system readily lends itself to development in a modular form where each component module acts as a functional unit. Since the system is designed as a multiple user, re-enterant system each component will reside in core simultaneously. The global organization of the program is shown in Figure 1. The primary cycle of the system is through MONITOR then to either MIXAL for an assembly language program or to SIMULATOR for a machine language program. The exact nature of control is determined by the requests issued by the user through the MONITOR command language. The proposed component modules are discussed in the following paragraphs.

Figure 1. Global Organization of the Mix Programming System



1. System Monitor

The System Monitor provides for system control, calling the assembler, the simulator and other component modules when required. The essential feature of the System Monitor is that it does not obstruct the student's view of the MIX computer. The System Monitor then must be capable of operating in the environment of a modern computer operating system and preventing the occurrence of items such as hardware detected errors from influencing the processing of a MIX program. Occurrences of this nature should be reported back to the System Monitor where error recovery procedures can be conducted to allow normal processing of the MIX program.

Since the system is to allow multiple users, the System Monitor must perform the storage allocation function, allocating storage for each MIX job. The arrival and duration of each MIX job is unpredictable; thus, storage allocation must be performed dynamically. The dynamic storage allocation for each MIX job is simplified as the amount required for each MIX job will be constant. This dynamically assigned storage is used to provide work areas, constants and simulated MIX memory required by the re-enterant nature of the program components. In other words, all information relating to a MIX job is maintained in this area and is accessed indirectly through the use of base or index registers.

It will be necessary for the System Monitor to maintain a status summary for each job entering the system. All information relating to the status of a particular job can be maintained in a status vector established within the dynamically assigned storage for each job. The status vector will then be available to all components of the system.

The MIX Programming System makes a variety of facilities available to the user. In order to use these facilities, the user must be able to specify what actions the system is to perform. The Monitor command language is a set of specification cards or control cards required to affect the actions desired. The crucial area of the command language is card format and design. Simple errors in card preparation should not result in job termination nor should simple errors cause erratic job behavior. Default parameters in the command language are not to be permitted. This is essential so that the student can feel that MIX does only as he has directed. The command language should be sufficient to permit the System Monitor to perform the following tasks:

1. Identify the program.
2. Assemble and execute a MIX assembly language program.
3. Execute a machine language program.
4. Initiate a post-mortem dump.
5. Produce a machine language object deck from an assembly program.
6. Provide program listing options.
7. Maintain accounting information on each user.
8. Perform required grader functions.
9. Set execution time limit in MIX time units.
10. Recognize end of job or delimiter between jobs.

Finally, the System Monitor must perform resource allocation for each MIX job. Since more than one MIX program may be running simultaneously, I/O requirements created by individual MIX programs may be interleaved. The System Monitor must keep track of the I/O requirements associated with each MIX program. In effect, the System

Monitor must create a continuous input and output stream for each MIX program. The techniques available for implementation of resource allocation by the System Monitor are, to a large extent, dependent on the computer system environment in which the MIX Programming System will be run. The number and type of peripheral I/O devices which are attached to the MIX computer will influence the complexity of the resource allocation task. As a minimum, the MIX computer should have a printer and card reader, but disk and tape capabilities are desirable for hard core programming courses.

Reference 3 is an excellent summary of monitor and supervisory systems.

2. MIX Simulator

The MIX Simulator must provide for interpretative execution of MIX machine language. Considering the horror with which programming in machine language is considered, this may seem to be an unusual requirement. However, a simulator in some form must be present to execute the machine code generated by a MIX assembler and some experience in programming directly in machine code is at least desirable if not necessary. Programming in machine language is not as difficult as it may seem, particularly for smaller programs frequently used as student programming exercises. There is a certain amount of appeal in machine language programming. Every instruction executed is a creation supplied in a very deliberate and intentional manner. The programmer is in a position to understand precisely what his program did as compared to what it is doing. It is here that the MIX Programming System should provide an indication of computer status on demand from the user. The System should provide a printout of

switch positions and lights available at the computer console. This would include items such as the overflow switch and comparison indicator.

The Simulator supplies vital statistics to the Monitor. Included are MIX execution time, number of instructions executed and how the job terminated.

A MIX Simulator, described in Chapter III and documented in Appendices A, B, and C, has been provided for immediate implementation on IBM 360/67. This will permit an interested user to have immediate access to a tool for use in machine language programming while remaining components of the system are developed and implemented.

3. MIX Assembler

A MIX Assembler must accept the MIX Assembly Language (MIXAL) and emit executable code in the format which the Simulator assumes to be loaded in MIX memory. This format is described in Chapter III. The assembler must also provide for the conversion of this format to the format described in exercise 26, Section 1.3.1 of reference 2. This conversion is necessary to provide for outputting object decks in true MIX machine language code.

The MIX Assembler should perform the usual assembly functions of converting mnemonic operation code to machine language codes, converting labels to storage addresses, translating symbolic expressions in operand fields to their numeric equivalent, defining literals and constants and reserving storage. Hashing or randomizing techniques should be used for symbolic accesses. A good discussion of these techniques may be found in references 4 and 5.

4. MIX Grader

The role of a grader is to assist an instructor in achieving standardized evaluation of student programming exercises. Although this feature is not a necessary part of a programming system, it becomes extremely important in a student programming environment where several hundred student exercises may be run each week.

The basic requirements of a grader are:

1. It should grade exercises in MIX machine and assembly language.
2. It should require a minimal amount of active student involvement.
3. It should record every attempt at an exercise and provide summaries on demand by the instructor.
4. It should be reasonably well protected from manipulation by the clever student as well as from the student who blunders in.
5. It should permit student rosters and course identification to be specified.
6. It should provide the instructor with the ability to specify a limited number of intermediate answers and a final answer for an exercise.

It is to be remembered that the grader is to function as an assistant to the instructor and should not assume the role of being the sole grader of student exercises. It can provide the instructor with such information as successful assembly, successful execution, correct answers, number of runs, execution time and number of instructions for each student.

The decision to permit specifications of intermediate and final answers can add considerably to the complexity of the grader. The

evaluation of answers produced by a student program, if any, is an important part of the overall evaluation of a student program. If there are intermediate answers, the student programmer will be required to place his answers in some specified location. This poses some restraint on the student programmer but since he must put answers somewhere, it might as well be at some place accessible by the grader. One technique would be to require the student to store answers at some absolute core location in MIX core. Answers, if any, would always be at the same location for easy access without any specification required by either the instructor or the student. Evaluation of answers poses another artificial requirement on the student in that as each intermediate answer is derived, the student program must return control to the grader for evaluation of the answer. Care must be exercised to limit the extent of such artificial requirements on the student.

Some fringe benefits may result from the use of a grader. It may motivate the student to take more care in the preparation of each run he attempts and hence lead to the development of better programming habits. If an unlimited number of runs for each job is permitted without penalty, there is a tendency to let the computer system do all the checking for card punch, syntax and other similar errors. The student who knows that each run counts will attempt to make the most out of each run.

A good discussion of graders is given in references 6 and 7.

5. Diagnostic Facilities

The MIX Programming System can provide two essential debugging aids, the trace and the post-mortem dump. Since the MIX Simulator is an interpretative routine executing one instruction

at a time, implementation of a trace routine which prints out information at execution time is relatively simple. The trace should be directly controlled by the student programmer with simple commands such as TRACE ON and TRACE OFF. The trace can be made available for either assembly or machine language programs. Implementation can be accomplished by establishing a trace indicator in the simulator control cycle. The indicator can be set by various means. One means is to attach an indicator to each instruction that is to be traced. This means is suitable for the MIX Simulator described in Chapter III since there is an unused bit already available in the instruction as stored in IBM 360 core. Another means is to require the command language trace card to specify start and stop MIX locations for the trace. The trace indicator is then set by the simulator control cycle by testing the location counter. When on, the trace indicator causes a branch to a subroutine which prints out desired trace information. This information includes MIX register contents, comparison indicator state, overflow indicator state, the instruction just executed and the contents of the memory cell specified by the instruction just executed.

The post-mortem dump should permit the student to request a given area of MIX memory be produced in MIX character code. Readability of the dump is of primary concern. The MIX memory location followed by the contents of that location should be printed. The contents of the MIX location should be printed according to MIX character code and not in binary, octal, or hexadecimal. The dump should also print out the contents of MIX registers.

III. THE MIX SIMULATOR

A. INTERPRETATIVE ROUTINES

An interpretative routine (or interpreter) is a type of translator program that performs the instructions of another program. Each instruction of the source program is first translated and then executed. The interpreter consists of two basic parts:

1. A set of subroutines corresponding to the set of operations in the source language.
2. A control section that analyzes each source instruction and selects the appropriate execution subroutine.

Interpreters are used widely in programming systems. The source language for an interpreter is usually a machine-like language such as the assembly language for another computer. Examples of tasks which can be performed by interpretative routines are:

1. Translation and execution of the instructions of a proposed computer. This allows checking the format and structure of the proposed computer instruction codes.
2. Performance of debugging aids.
3. Representation of a complicated sequence of decisions and actions.
4. Communication between passes of a multipass system.

The combined translate/execute feature of interpreters is responsible for most of the important characteristics of interpretative routines. Some of these caracteristics are:

1. If looping is to be permitted in the source language, then all of the source program must be retained in memory.

2. Even though a source statement may have been translated and executed once, any looping or re-execution of that statement will require re-translation.

3. Interpreters tend to be slow since each instruction in a loop must be re-translated each time through the loop.

4. Diagnostics can be written in terms of the source language rather than the object code of the interpreter.

Interpretative routines are not widely used for production work for the simple reason that they make a high speed, expensive computer perform like its cheaper cousin in regards to execution time. Only in programs requiring short execution times (such as student exercises) can interpretative execution be justified because of its advantageous debugging and diagnostic characteristics.

B. DESCRIPTION AND STRUCTURE

The MIX Simulator is designed to interpretatively execute MIX machine language as a source language. In addition to the control cycle and associated subroutines, the MIX simulator has the following additional features:

1. The input machine language is checked for format and content errors. This feature enhances the use of the simulator when programs are written directly in MIX machine language.

2. The program is completely reenterant. In a multi-programming environment such as the IBM OS/360, this means that more than one MIX program can be in execution while only one copy of the MIX Simulator is currently stored in memory.

The detailed operational sequences of the MIX simulator control routine and associated subroutines are illustrated in Appendix B,

the Program Flowcharts, and in Appendix C, the Program Listing. The basic operational sequence is described below:

1. The program is initialized. Storage is dynamically allocated for work areas and simulated MIX memory.

2. Each input card is read and printed in two formats for the program listing. The input card is checked for errors. If errors are found on any input card, a switch is set to prevent further loading of information to simulated MIX memory and to prevent the program from executing. An error message is also printed.

3. When a transfer card is encountered, a switch is tested to see if input errors have been found and if there are no errors, the program enters the execute phase. If errors have been found, the job is terminated.

4. The transfer card provides the execute phase with the MIX location at which execution is to commence. The control cycle routine fetches the instruction to be executed and performs those functions which are common to all instructions. These functions include unpacking or breaking up the instruction into component parts suitable for manipulation by the program subroutines, computing the effective address, and branching to the appropriate operator subroutine.

5. The operator subroutine performs the functions of the MIX instruction. The operator subroutines may in turn call certain support routines. These support routines perform functions which are needed by some but not all operator subroutines. For example, the GETV subroutine is used by those instructions in which F is a field specification to fetch the specified field and return it to the calling routine.

6. The instruction fetch and execute cycle is continued until a program halt occurs or until the program is terminated by the occurrence of an execution error.

C. PROGRAMMING TECHNIQUES

Since a MIX program must remain in memory during execution, the format for storing a MIX word in simulated MIX core was chosen so as to minimize 360 core requirements. Since each MIX word contains 5 bytes plus sign and byte size for this computer is 6 bits; 31 bits are required. Each MIX word is stored in a 32 bit 360 fullword. The format is:

One alternate technique considered was to store each MIX word in a 6 byte 360 packed format. This would have required an additional 8000 bytes of 360 core to simulate a 4000 word MIX computer. The 32 bit full word format had the advantage of requiring less core and permitting greater use of register vice packed decimal processing of instructions.

Reenterability was achieved by using the reenterable forms of IBM 360 system macros and using the GETMAIN macro to assign storage. Communication with the assigned storage is established by the use of a DSECT which symbolically assigns all storage requirements for simulated MIX memory, MIX registers and other constants. In other words, all processing of MIX is done in storage which is assigned dynamically for each entering MIX program.

IV. CONCLUSIONS AND RECOMMENDATIONS

The MIX Simulator is completely adequate for use as a tutorial aid in teaching machine language programming, however, optimum use of the MIX Simulator could be achieved by incorporating it into a MIX Programming System which contains as a minimum a System Monitor, the MIX Simulator and a MIX Assembler. Some of the functions now performed by the MIX Simulator would be transferred to the System Monitor. These functions include editing MIX machine language source programs, converting from the format described by reference 2 to format stored in simulated MIX memory and loading the converted code to MIX memory.

As noted in Appendix A, the I/O operators are not true representatives of the I/O operators described in reference 2; these I/O operators permit a segment of code to initiate an I/O operation and then continue processing. They also permit checks to be made at some point to ensure that the I/O operation was complete before attempting to use a piece of data in a computation. This basic concept of overlapping I/O and processor operations should be made available to the student programmer.

Other minor changes to the MIX Simulator would be to provide pagination for the machine language program listing and program output and to include floating point arithmetic capability.

In conclusion, the MIX Simulator can be used alone as a valuable aid in teaching machine and assembly language programming when included in a MIX Programming System.

APPENDIX A

USERS MANUAL

The use of the MIX Simulator, as implemented under IBM OS/360, is described in this appendix.

A. USERS MANUAL

1. GENERAL INFORMATION

D.E. Knuth describes the MIX computer, the MIXAL assembly language and the MIX machine language [2]. This appendix provides that additional information needed to run MIX machine language programs on the NPGS MIX Simulator.

NPGS MIX Simulator differs from MIX as described in reference 2 in the following ways:

a. The byte size for NPGS MIX is 6 bits.

b. MIX words preceeding the transfer card are loaded according to the following format:

(1) Byte one and two: 2 byte address

(2) Byte three: Index register number

(3) Byte four: F specification

(4) Byte five: operation code

For example, a card which has "XXXXX101003005001636" punched in columns 1-20 would cause the following to be loaded:

Location 0100: +

3005	00	16	36
1-2	3	4	5

c. Only unit 16, a card reader, and unit 18, a printer, are provided.

d. All I/O devices take 250 MIX time units for any operation.

e. The operation codes for Jump Ready and Jump Busy are treated as NO operations and overlapping of I/O is not possible. On I/O the machine will wait until the transfer of information which

started with an IN/OUT instruction is completed. Therefore, a program may refer to the information in memory at any time since the memory reference will not be made until the I/O operation is completed.

2. INPUT EDIT CHECKING

The Mix Simulator includes a loader which reads MIX program cards, checks the input program cards for errors, issues input error messages and loads the MIX program to MIX memory if no errors have been discovered. Input errors which are checked and their associated error messages are as follows:

a. Message: WORD COUNT NOT CONSISTENT WITH NUMBER OF WORDS ON CARD.

   Cause: Column 6 denotes the number of consecutive words on this card to be loaded. The number in column 6 does not agree with the actual number of words on the card.

b. Message: INVALID WORD COUNT IN COLUMN 6, NOT IN RANGE 0-7.

   Cause: The number of consecutive words on a card may vary between 1-7 and may be 0 for the transfer card. Any number outside this range is invalid.

c. Message: INVALID LOCATION IN COLUMN 7-10, NOT IN RANGE 0-3999.

   Cause: Column 7-10 denotes the location at which the first word on the card is to be loaded. Any number beyond the range 0-3999 is outside the physical limits of MIX memory.

d. Message: INVALID CHARACTER OR SIGN PUNCHED ON CARD.

Cause: The information for each word is punched numerically as a decimal number with a minus (11-punch) overpunch over the least significant digit if the word is to be negative. Any other punches such as alphabetics or a minus overpunch in the wrong column is invalid.

e. Message: OP, F, OR I FIELD EXCEEDS BYTESIZE OP 63 OR AA FIELD EXCEEDS 2 BYTESIZE OF 4095.

Cause: MIX word not in accordance with paragraph 1 of this appendix.

If an error is discovered during program loading, the program will not be executed.

Only cards preceding the transfer card are checked for input errors. Any data cards following the transfer card are not edit checked. Care should be exercised to insure that a proper transfer card is present as the program will not execute if a transfer card is missing. The correct format for the transfer card is described in the answer to exercise 26, Section 1.3.1 of reference 2.

3. EXECUTION ERRORS

Once a MIX program has been loaded and execution commences, errors can occur which will cause execution to be terminated and a diagnostic error message printed. Each diagnostic message commences with the header "*****ERROR*****LOC____" where LOC is the MIX memory location at which the error occurred. Each diagnostic message is printed as a plain language message and is to a certain degree, self-explanatory. Each diagnostic message and possible causes are discussed below.

a. Message: ILLEGAL INDEX SPECIFIED.

Cause: The index number specified in a MIX instruction is not in the range 0-6.

b. Message: EFFECTIVE ADDRESS TOO LARGE.

Cause: The MIX location specified in a MIX instruction, after indexing, is not in the range -4095 to 4095. Note that in certain instructions the MIX location is treated as a signed number and as such is valid so long as the number will fit in two MIX bytes.

c. Message: MEMORY REFERENCE BEYOND LIMITS OF MIX CORE.

Cause: The MIX location specified in a MIX instruction, after indexing, is not in the range 0-3999. This error can occur only in instructions, such as LOAD or STORE instructions, where the MIX location is used to make a memory reference.

d. Message: ILLEGAL F SPECIFIED.

Cause: In the instructions where F is used to specify the field of a MIX word, F is not of the form 8L+R. In the instructions where F is used to specify a variant of the operation code, F is not valid for that operation code.

e. Message: ILLEGAL ATTEMPT TO LOAD INDEX REGISTER WITH VALUE GREATER THAN 4095.

Cause: Index registers hold two bytes plus sign. Any operation which attempts to place a number outside the range -4095 to 4095 is undefined.

f. Message: UNDEFINED I/O OPERATION.

Cause: Improper unit specified by F or M 0 in an IOC instruction.

4. PROGRAM TERMINATION

A MIX program may be terminated in one of four ways. They are:

a. Input errors are discovered while loading the program. The remainder of the program is listed and checked for input errors but load is terminated and execution will not occur. Upon reaching a valid transfer card, the message "EXECUTION NOT ATTEMPTED DUE TO INPUT ERRORS" is printed and the job terminated.

b. Errors are discovered during program execution. The diagnostic message is printed, execution time in MIX time units to point of termination is printed and the job is terminated.

c. If a valid transfer card is not present in the program or the MIX program attempts to read a card from an empty reader, execution is terminated with the message "CARD READER EMPTY OR VALID TRANSFER CARD NOT ENCOUNTERED."

d. Execution proceeds to completion after encountering a HALT instruction and the MIX execution time units are printed.

5. OVERFLOW

On several instructions, overflow is permissible and execution will continue with the remainder modulo word size. When overflow occurs, the warning message "LOC____: ARITHMETIC OVERFLOW, EXECUTION CONTINUES MODULO WORD SIZE" is printed.

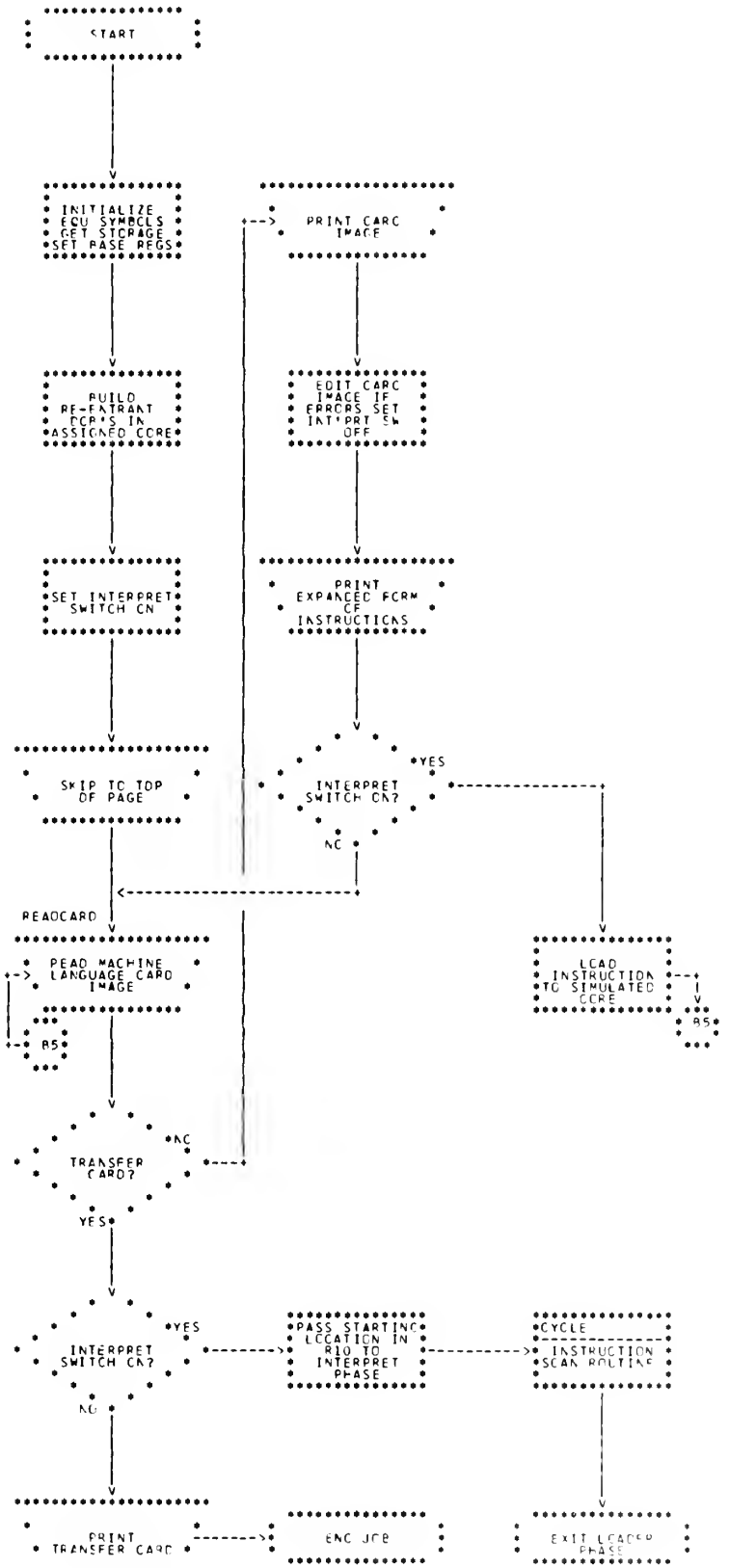
APPENDIX B

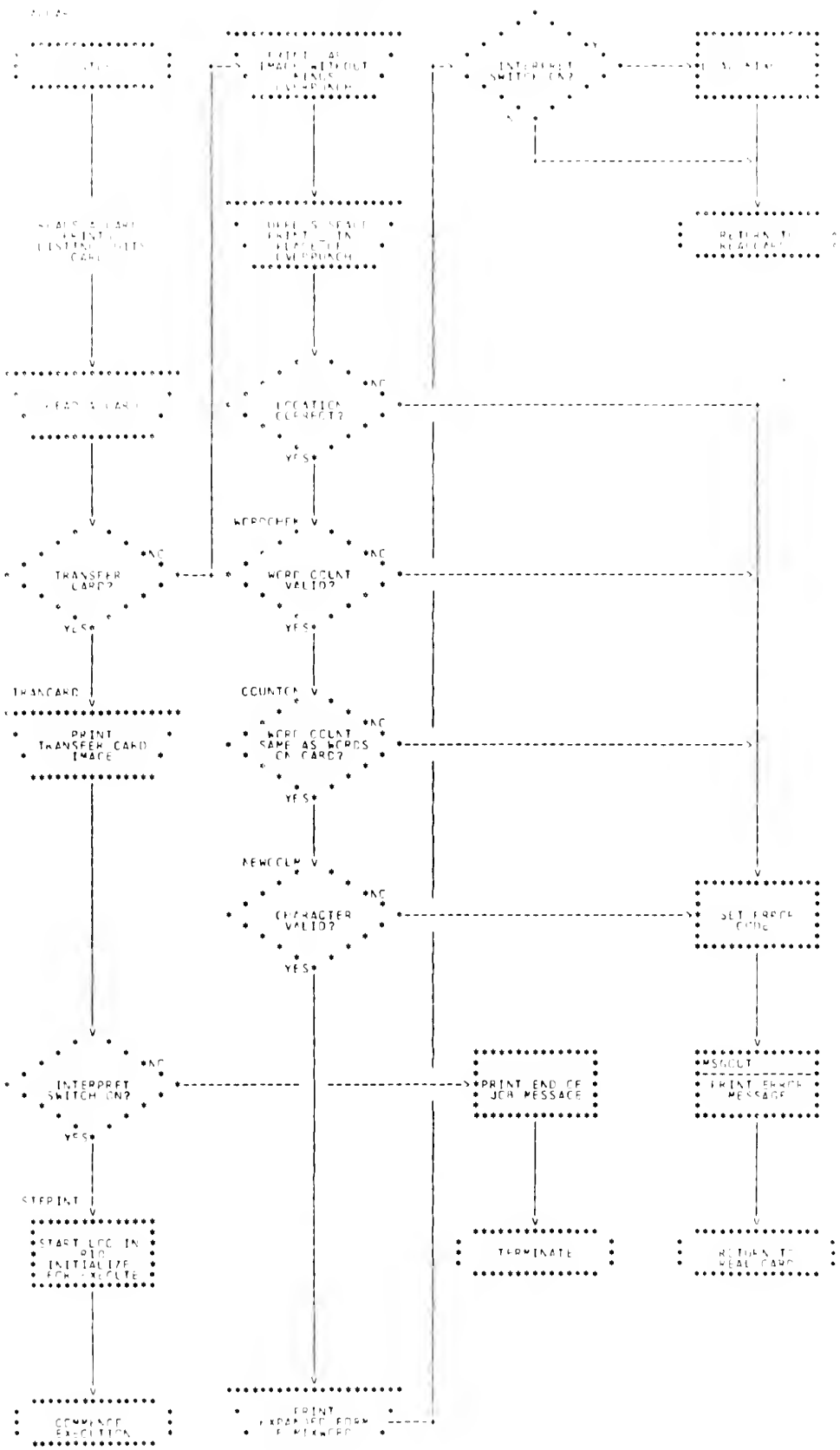
THE MIX SIMULATOR PROGRAM FLOWCHARTS

Flowcharts for the various subroutines of the Mix Simulator are provided.

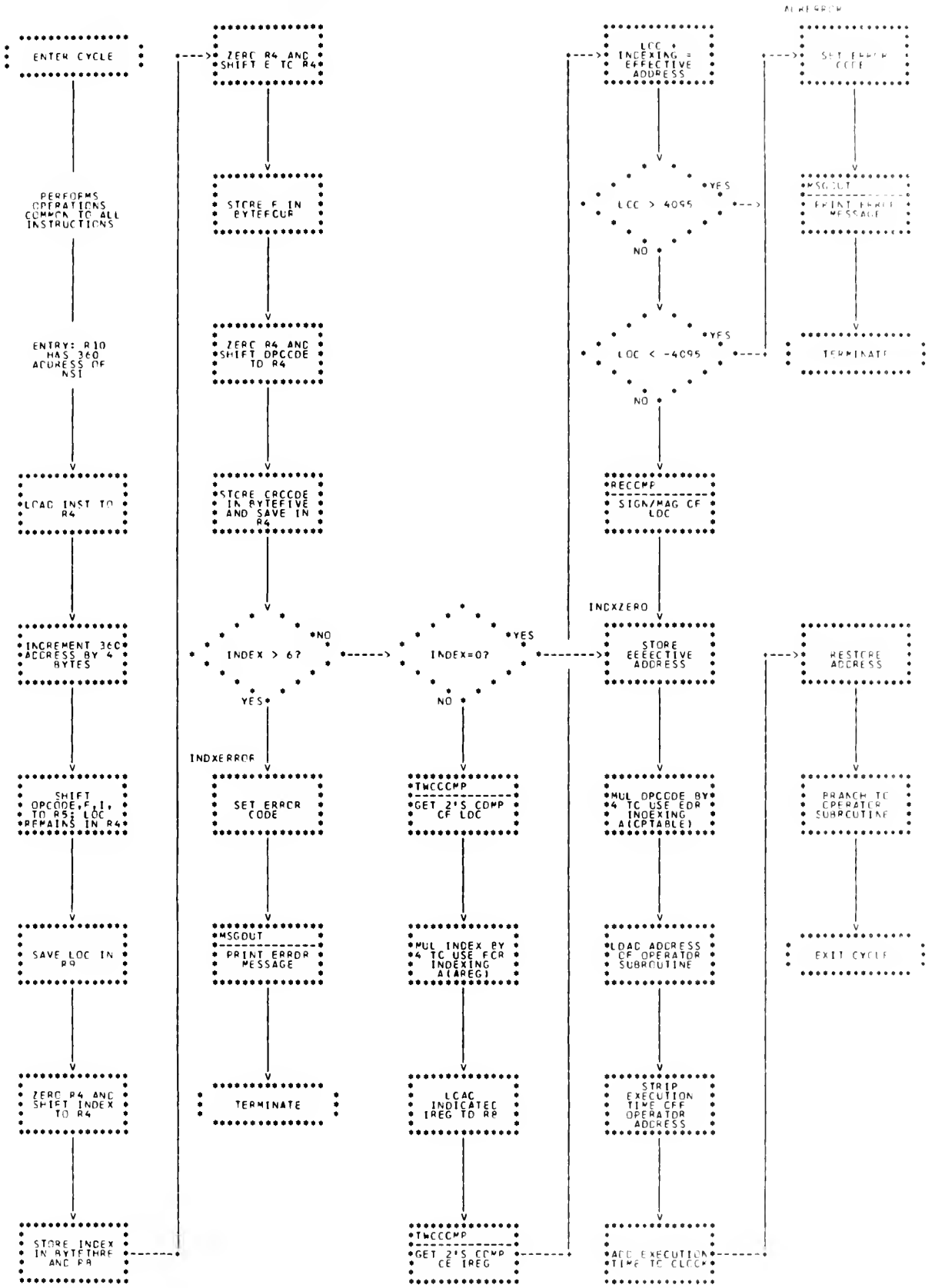
Title	Page
Loader Phase	31
Readcard	32
Scan Routine Cycle	33
GetV	34
Fetch	35
Fcheck	36
Abadr	37
Unpack	38
Pack	39
Two's Complement	40
Sign and Magnitude	41
Error	42
Input Error	43
Msgout	44
Add/Subtract	45
Multiply	46
Divide	47
Numeric	48
Shift(SLA, SRA)	49
Shift(SLC, SRC, SRAX,SLAX)	50
Move	51
LDA,LDAN,LDX, LDXN	52
IO Control	53
LDI	54
LDIN	55
Store	56
In	57
Out	58
Jump	59
Regjump	60
Regops	61
Compare	62

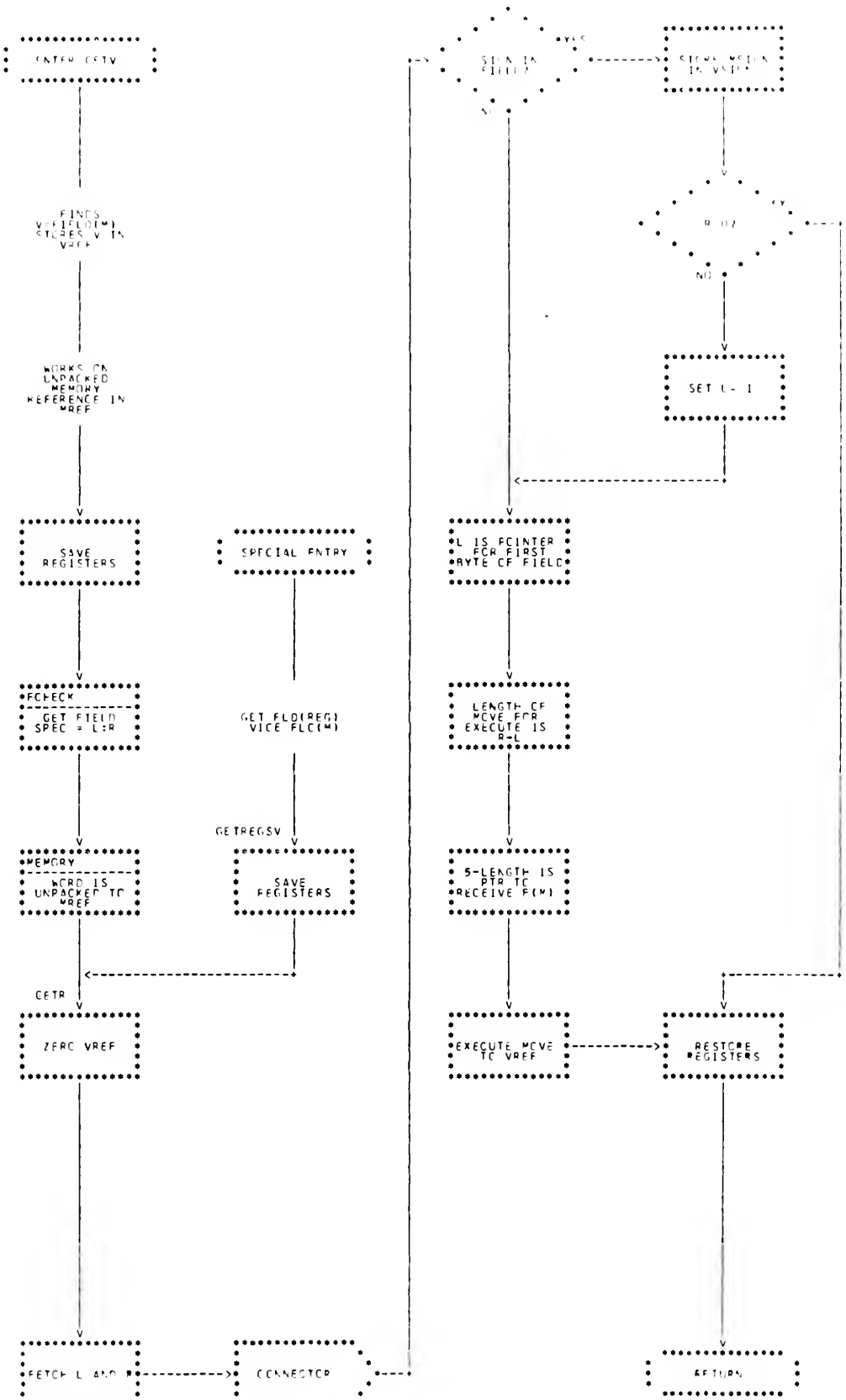
BASIC SYSTEM FLOWCHART LOADER PHASE



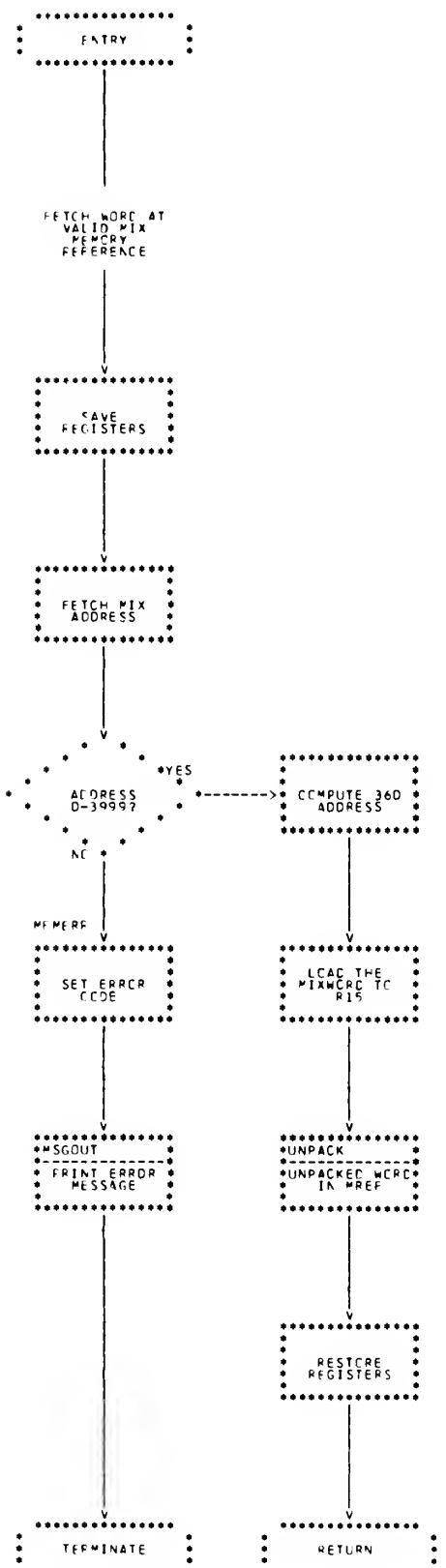

INTERPRET SWITCH CARD
NO
YES
TRANSFER CARD?
TRANCARD
PRINT TRANSFER CARD IMAGE
INTERPRET SWITCH CARD?
STEPINT
START LOC IN OLD INITIALIZE FOR CYCLE
COMMENCE EXECUTION
LOCATION CORRECT?
WORD COUNT VALID?
COUNTCK
WORD COUNT SAME AS WORDS ON CARD?
NEWCCLK
CHARACTER VALID?
PRINT EXPANDED FORM OF MISWORD
SET ERROR CODE
MSGOUT
PRINT ERROR MESSAGE
PRINT END OF JOB MESSAGE
TERMINATE
RETURN TO REAL CARD

FLOWCHART INSTRUCTION SCAN ROUTINE CYCLE

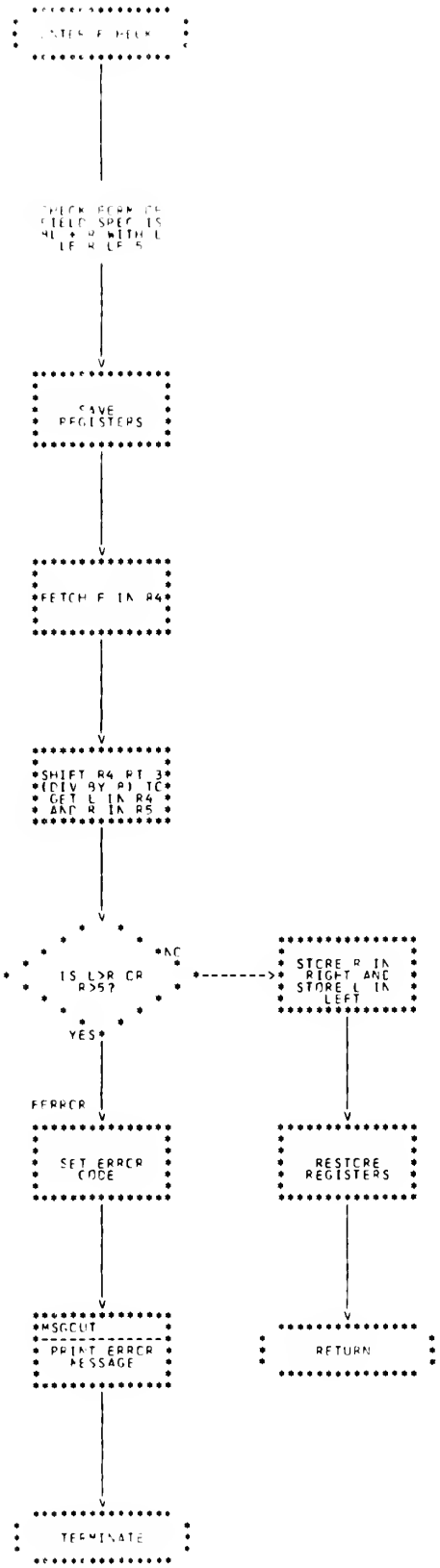



ENTER CFTV
FINDS V=FIELD(M) STORES V IN VREF
WORKS ON UNPACKED MEMORY REFERENCE IN VREF
SAVE REGISTERS
FCHECK
GET FIELD SPEC = L:R
MEMORY
WORD IS UNPACKED TO VREF
SPECIAL ENTRY
GET FLD(REG) VICE FLD(M)
GETREGSV
SAVE REGISTERS
GETR
ZERO VREF
FETCH L AND R
CONNECTOR
SIGN IN FIELD?
YES
NO
STORE V=SIGN IN VSIGN
R OR Z
YES
NO
SET L= 1
L IS POINTER FOR FIRST BYTE OF FIELD
LENGTH OF MOVE FOR EXECUTE IS R+L
S-LENGTH IS PTR TO RECEIVE F(M)
EXECUTE MOVE TO VREF
RESTORE REGISTERS
RETURN

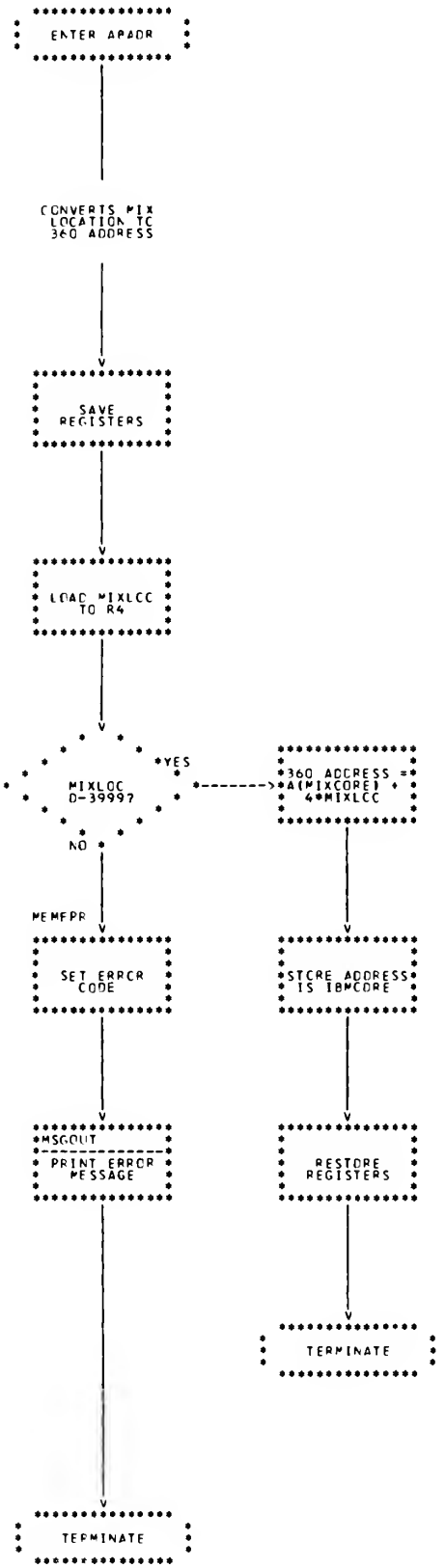
MIX MEMORY FETCH SUBROUTINE



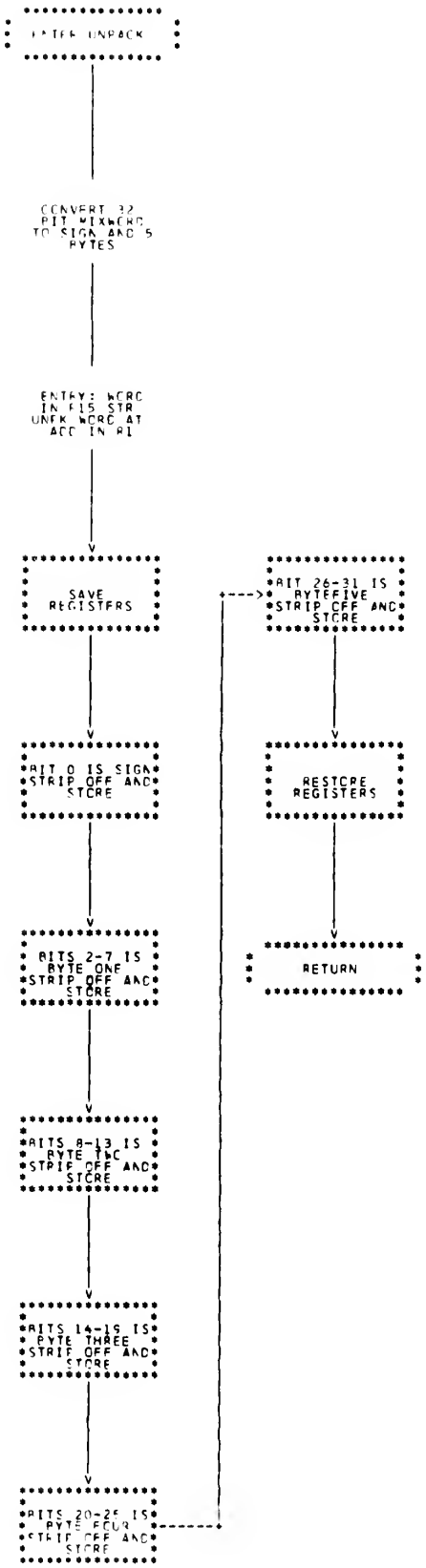
FLOWCHART FOR CHECK SUBROUTINE



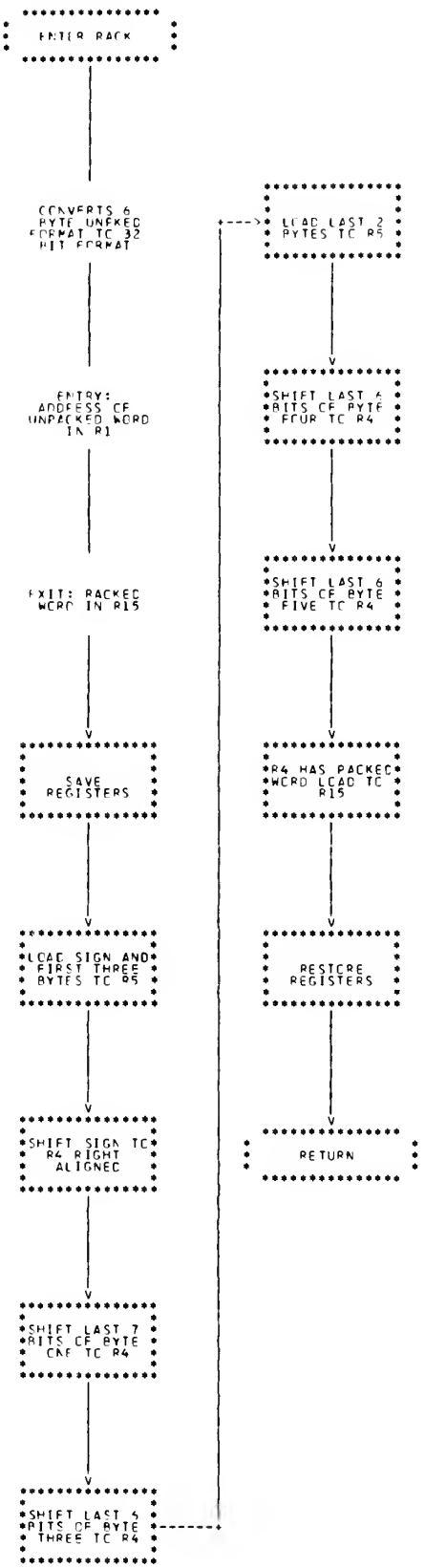
FLOWCHART SUBROUTINE ARADR

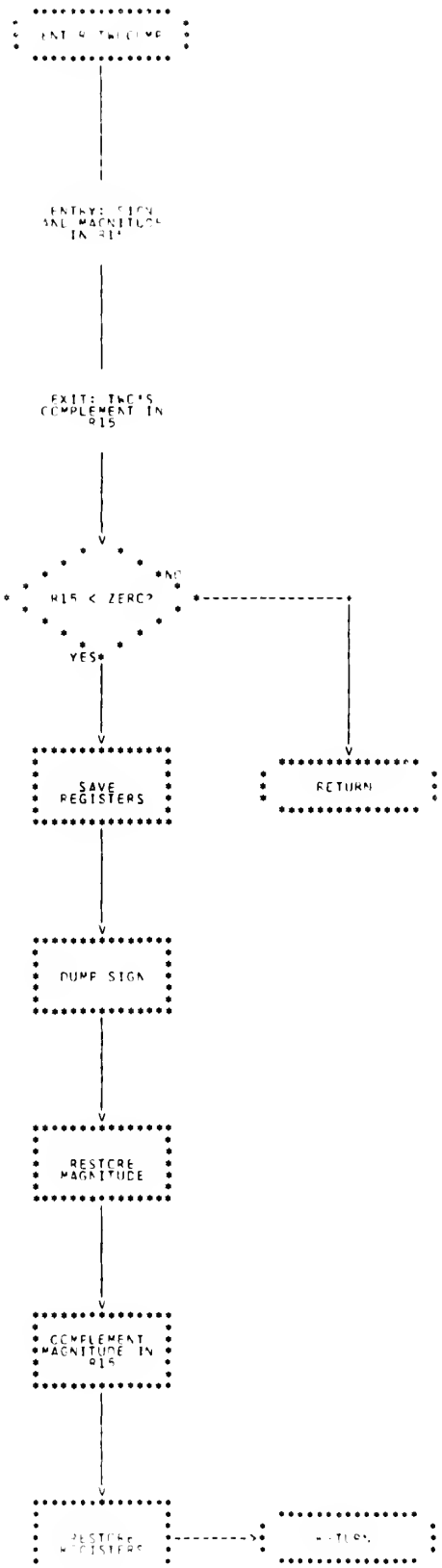


FLOWCHART SUBROUTINE UNPACK

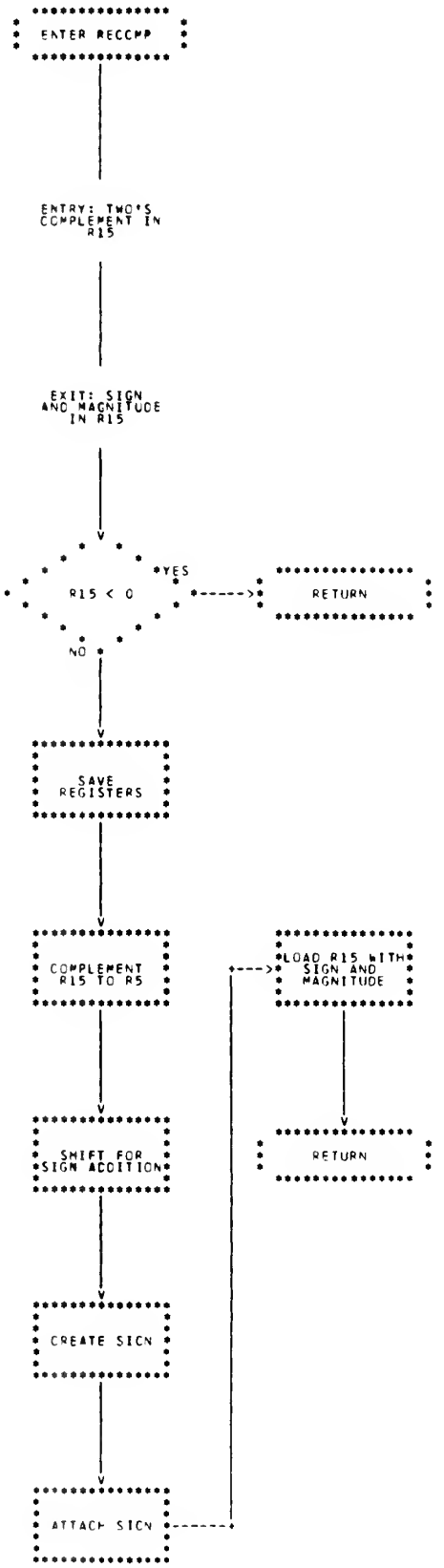


FLOWCHART SUBROUTINE PACK

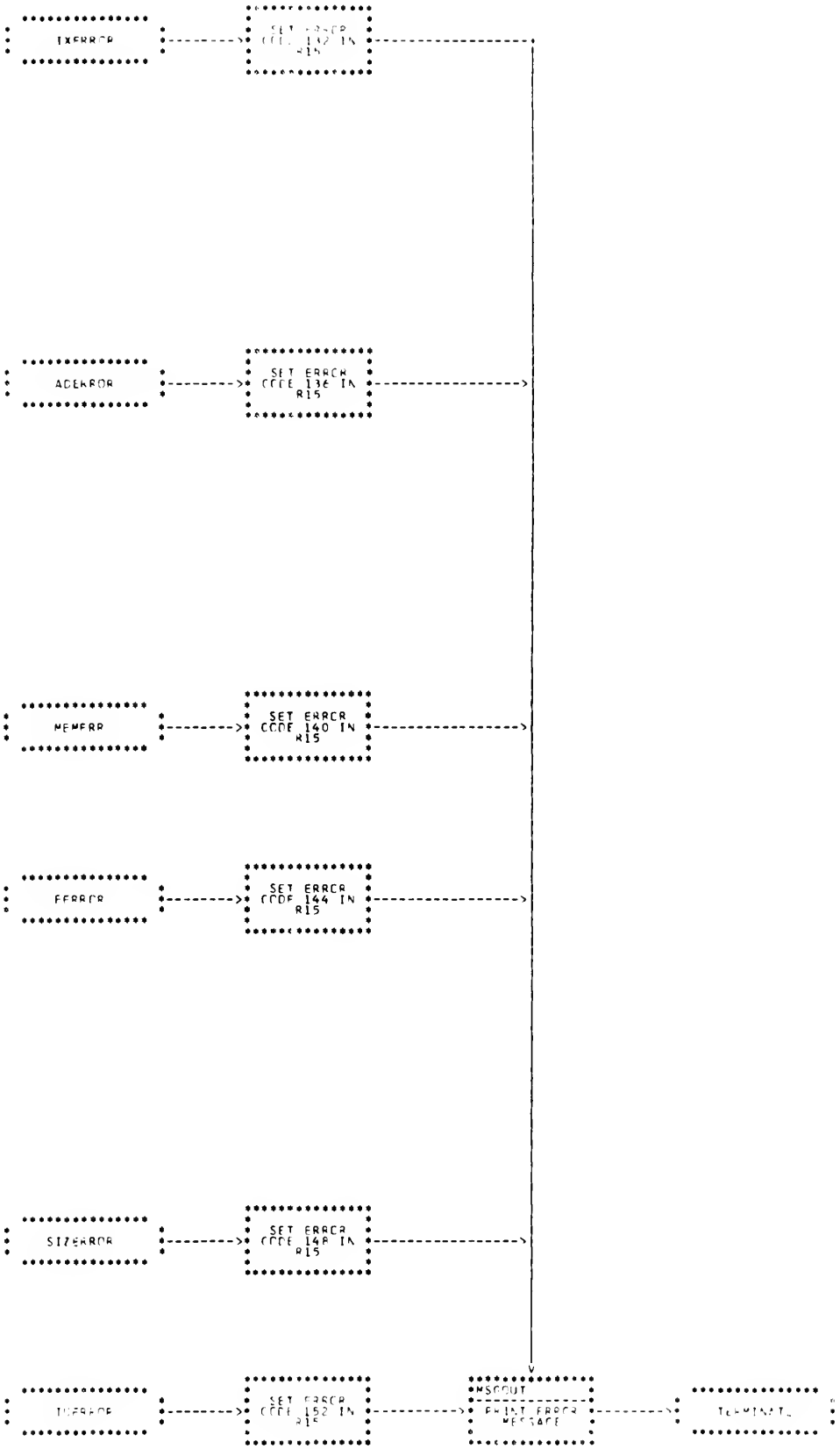



ENTRY: SIGN AND MAGNITUDE IN R15
EXIT: TWO'S COMPLEMENT IN R15
R15 < ZERO?
NO
YES
SAVE REGISTERS
RETURN
DUMP SIGN
RESTORE MAGNITUDE
COMPLEMENT MAGNITUDE IN R15

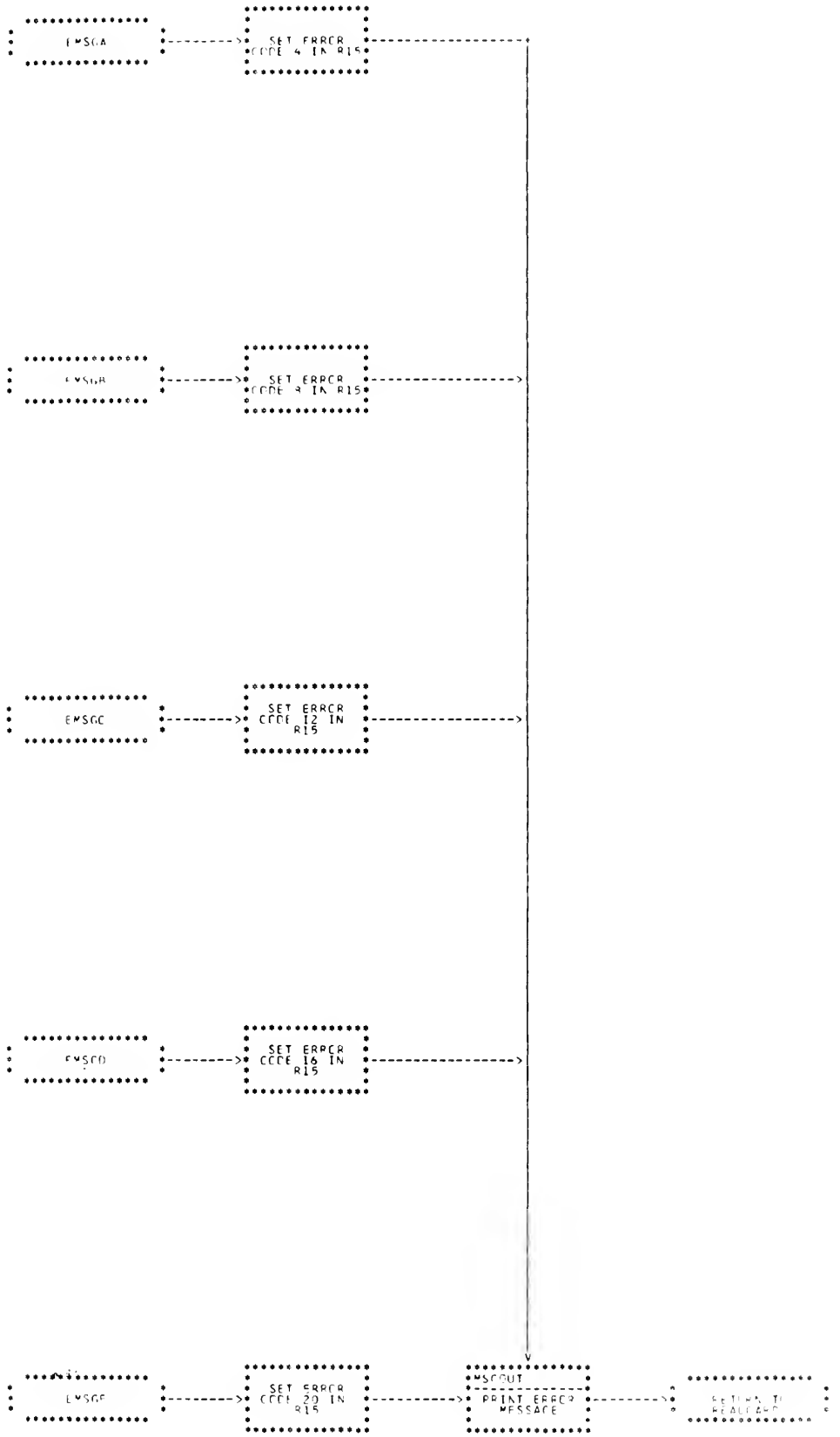
FLOWCHART SIGN AND MAGNITUDE SUBROUTINE

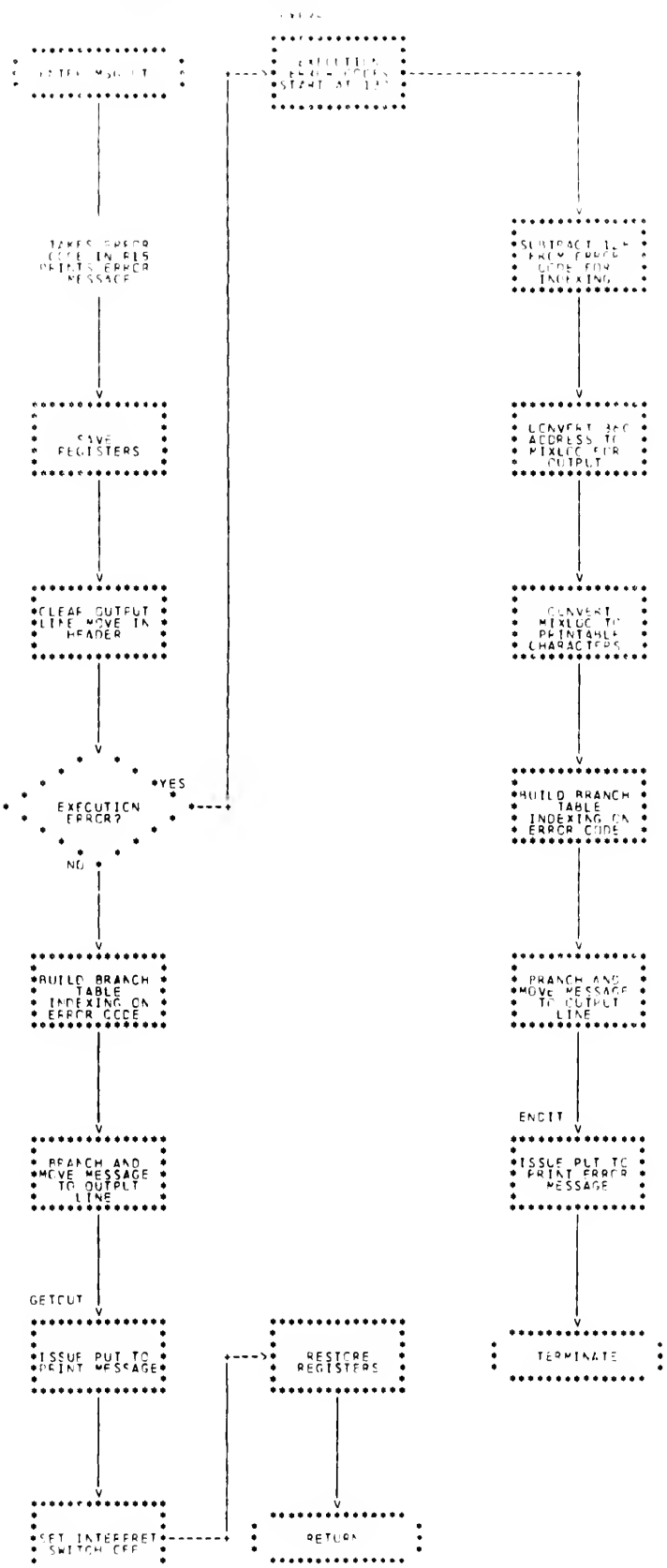


FLOWCHART EXECUTION ERROR ROUTINES

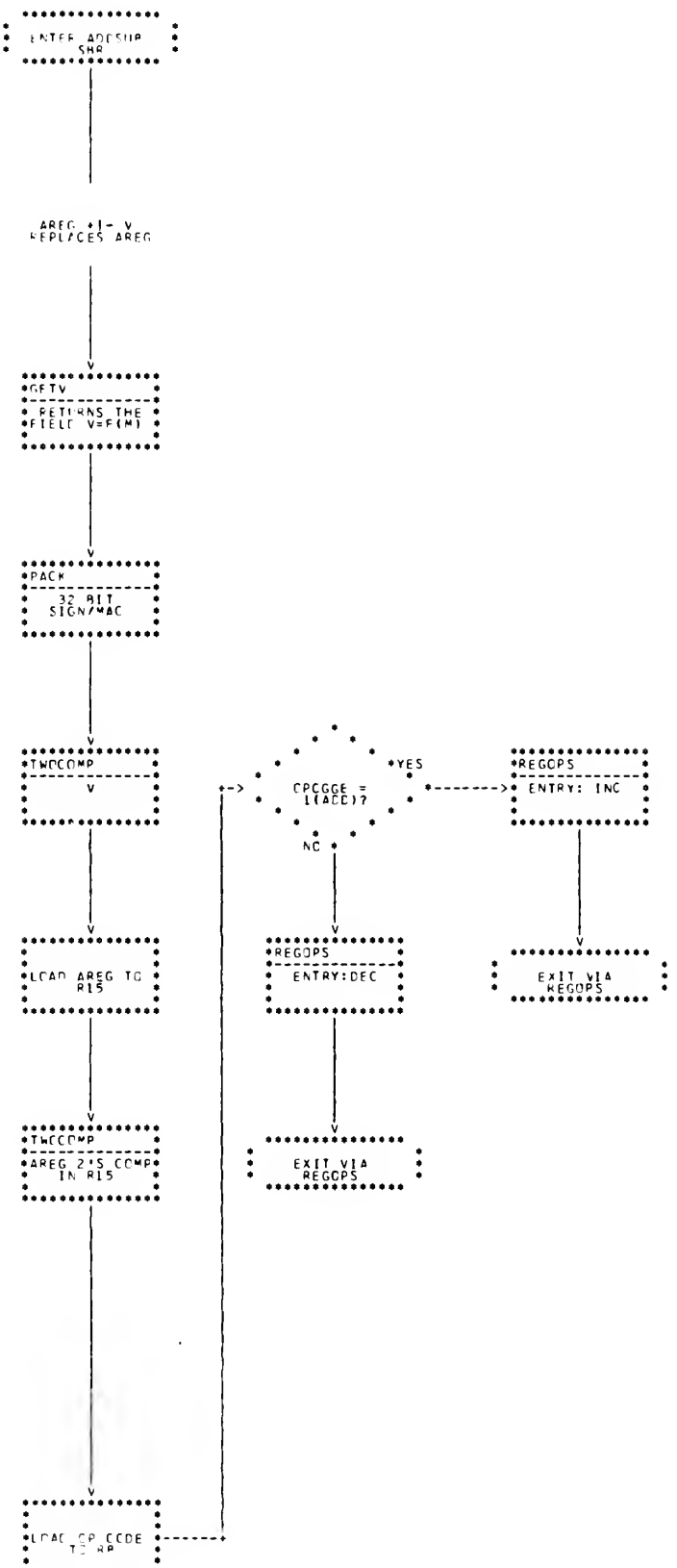


FLOWCHART INPUT ERROR SUBROUTINE

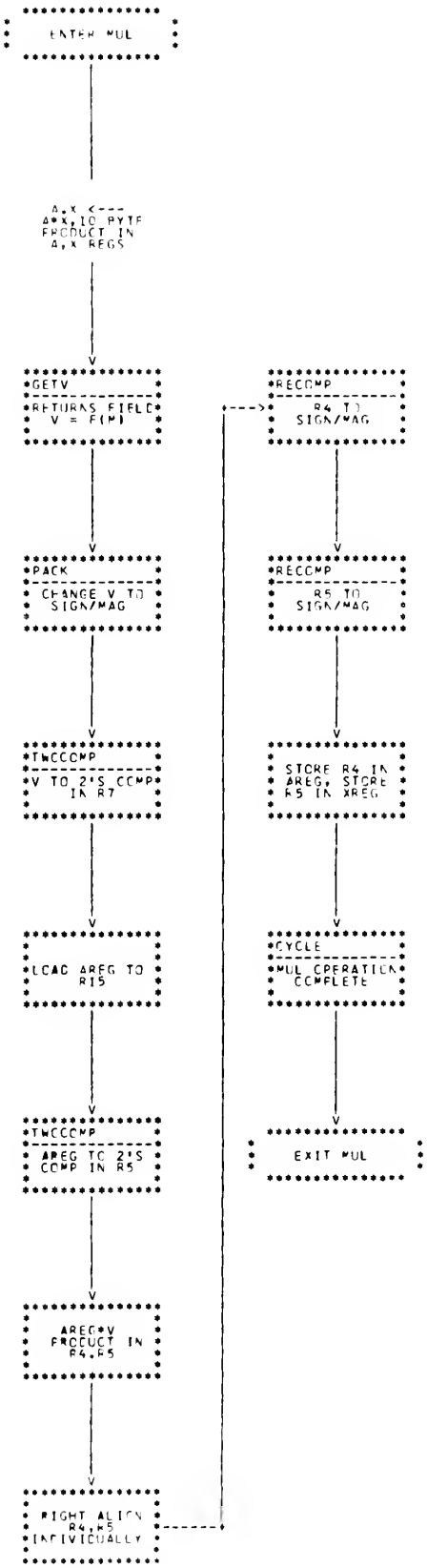



TAKES ERROR CODE IN R15 PRINTS ERROR MESSAGE
SAVE REGISTERS
CLEAR OUTPUT LINE MOVE IN HEADER
EXECUTION ERROR?
YES
NO
BUILD BRANCH TABLE INDEXING ON ERROR CODE
BRANCH AND MOVE MESSAGE TO OUTPUT LINE
GETOUT
ISSUE PUT TO PRINT MESSAGE
SET INTERPRET SWITCH OFF
RESTORE REGISTERS
RETURN
SUBTRACT 1 FROM ERROR CODE FOR INDEXING
CONVERT BIN ADDRESS TO MIXED FOR OUTPUT
CONVERT MIXED TO PRINTABLE CHARACTERS
BUILD BRANCH TABLE INDEXING ON ERROR CODE
BRANCH AND MOVE MESSAGE TO OUTPUT LINE
ENDIT
ISSUE PUT TO PRINT ERROR MESSAGE
TERMINATE

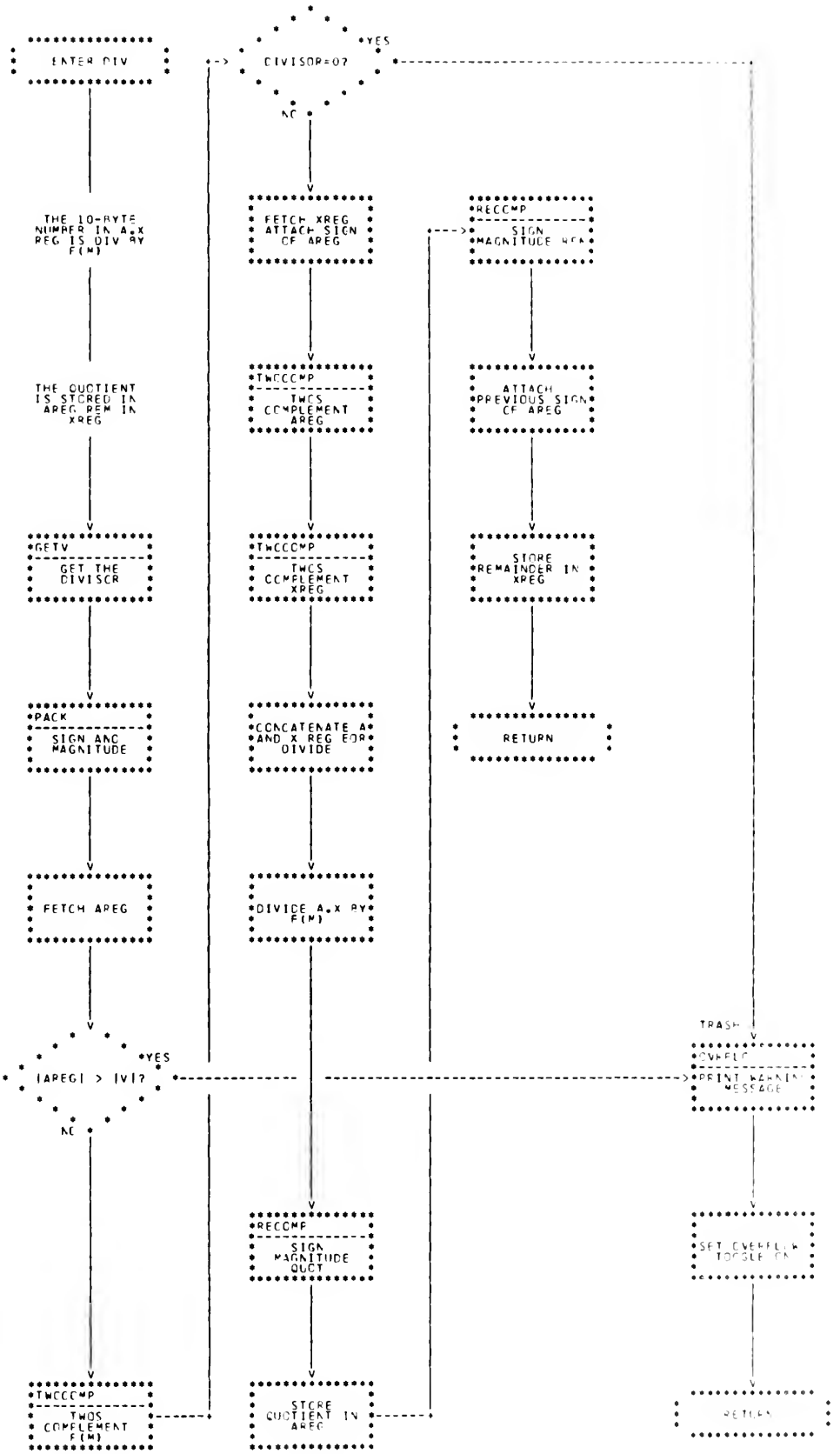
FLOWCHART ADD/SUBTRACT OPERATOR SUBROUTINE



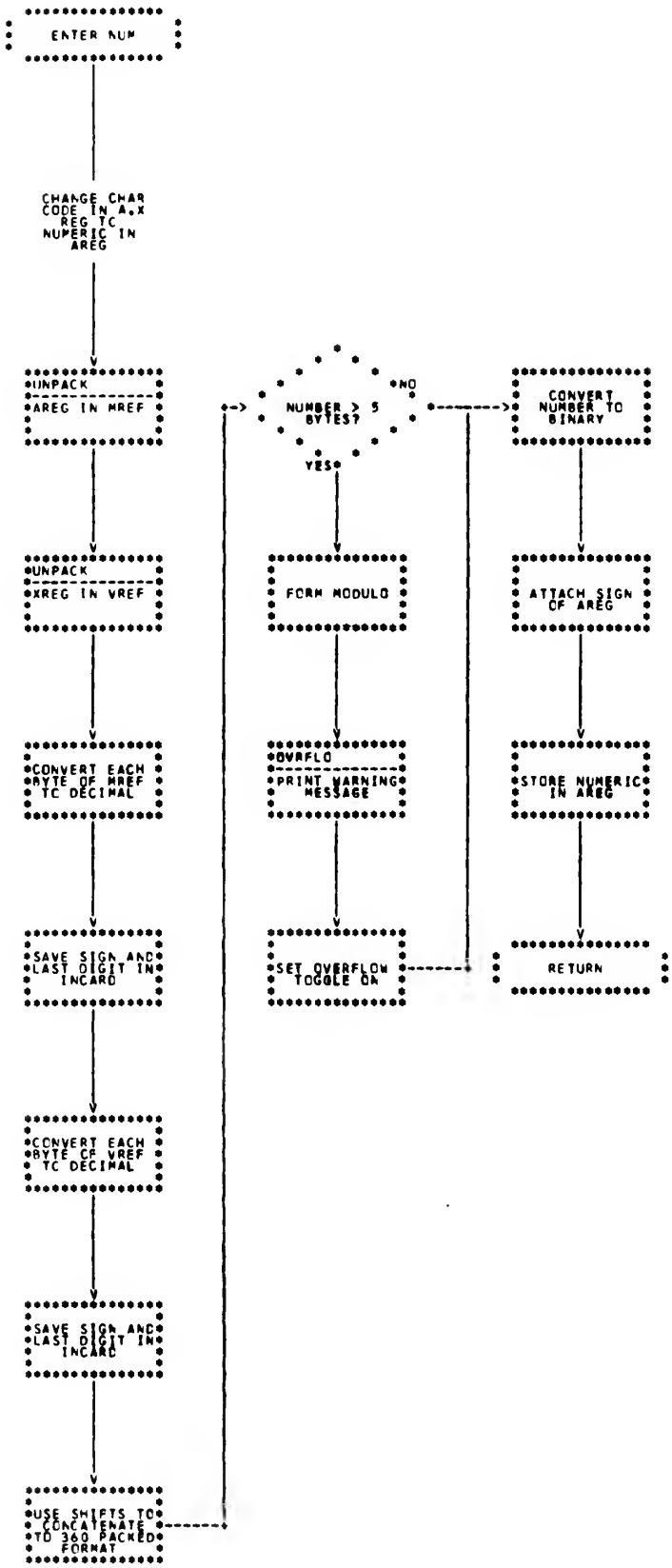
MULTIPLY OPERATOR SUBROUTINE



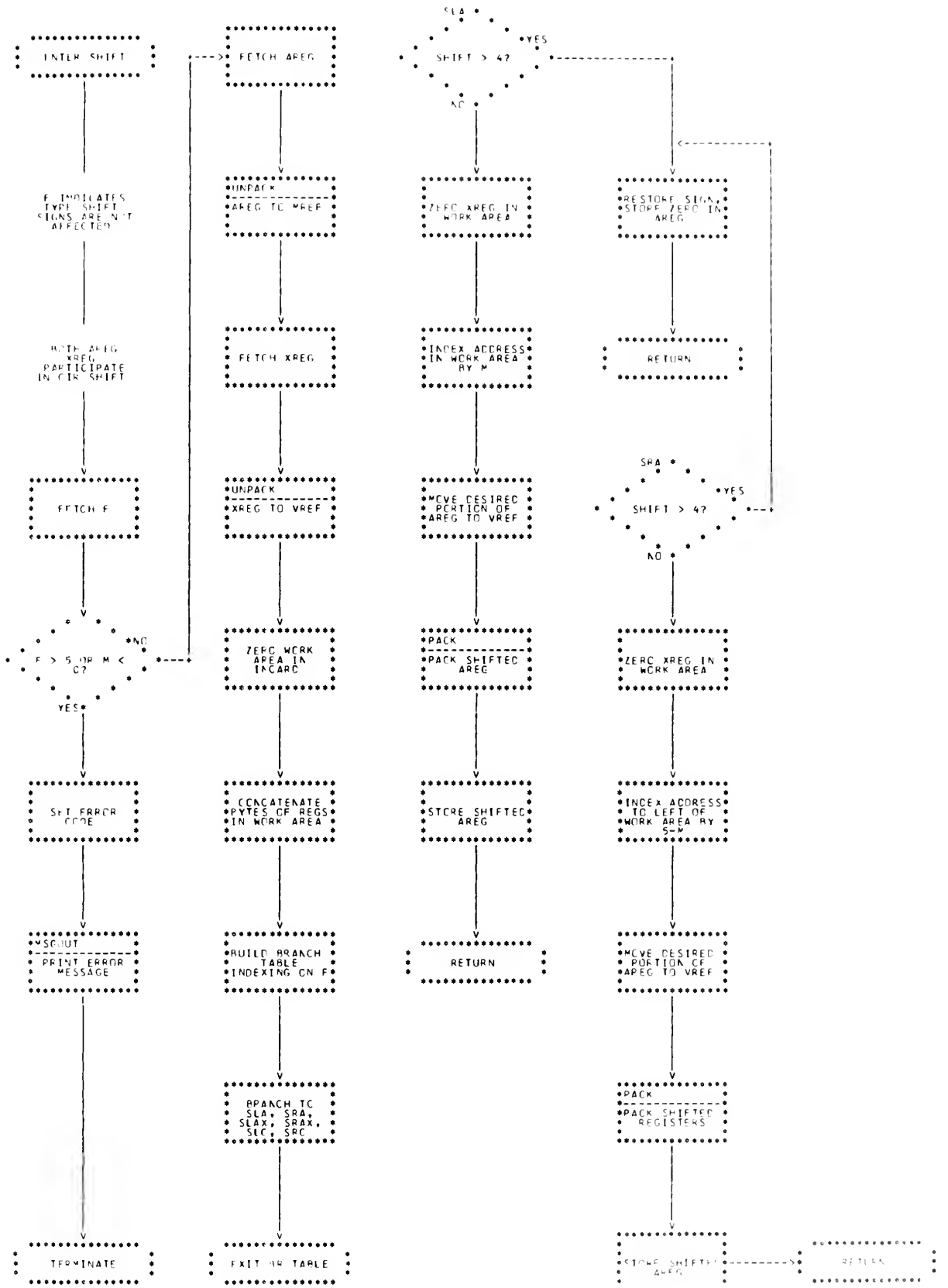
FLOWCHART SUBROUTINE DIVIDE OPERATOR



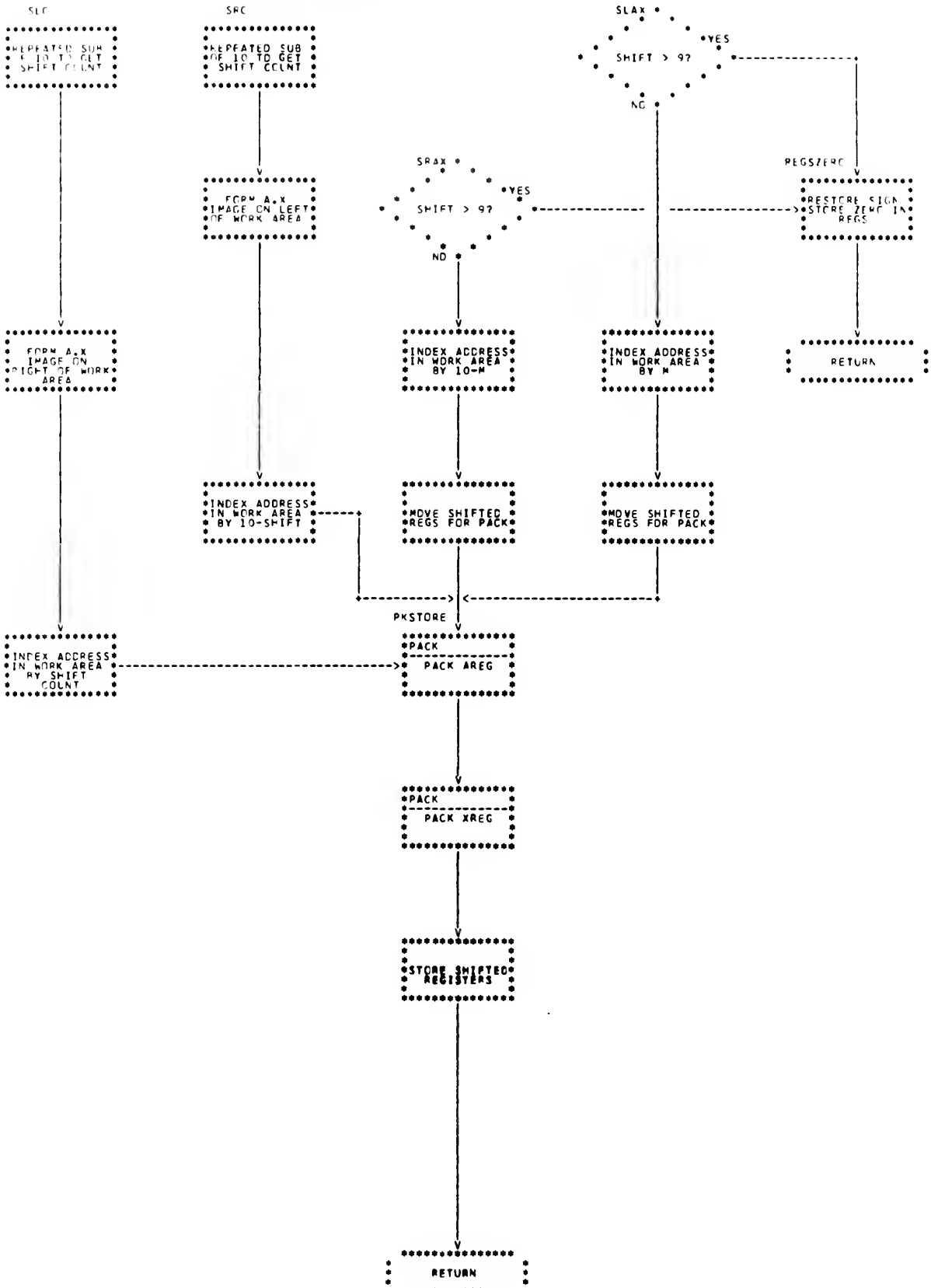
FLOWCHART SUBROUTINE NUMERIC



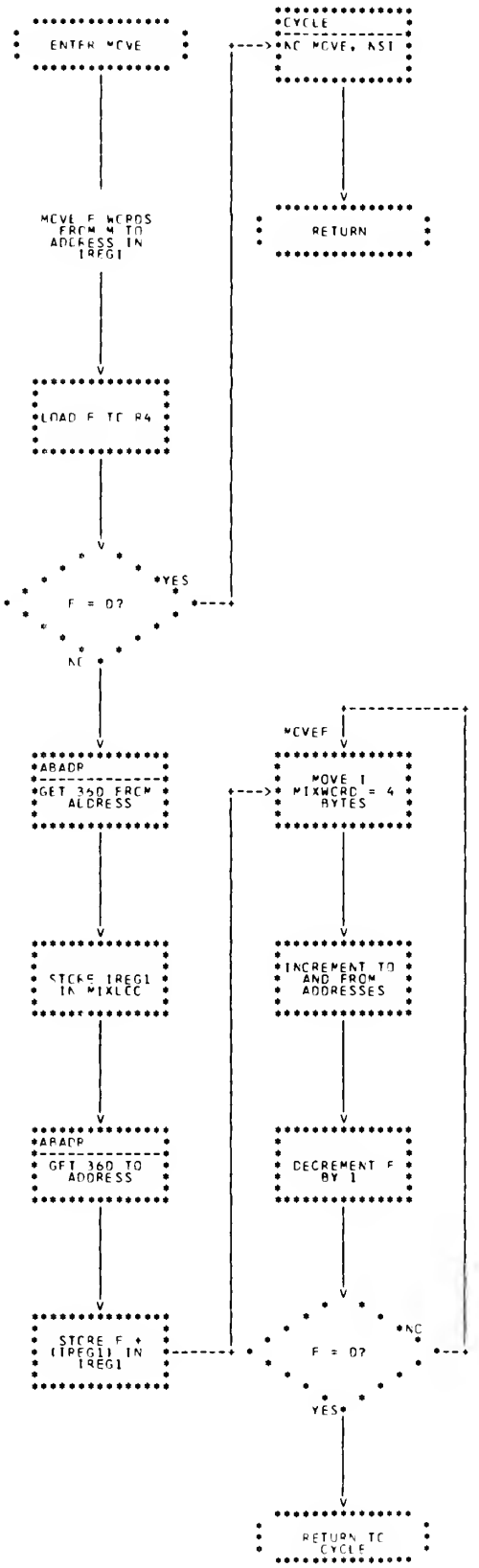
FLOWCHART SUBROUTINE SHIFT(SLA,SRA) OPERATORS

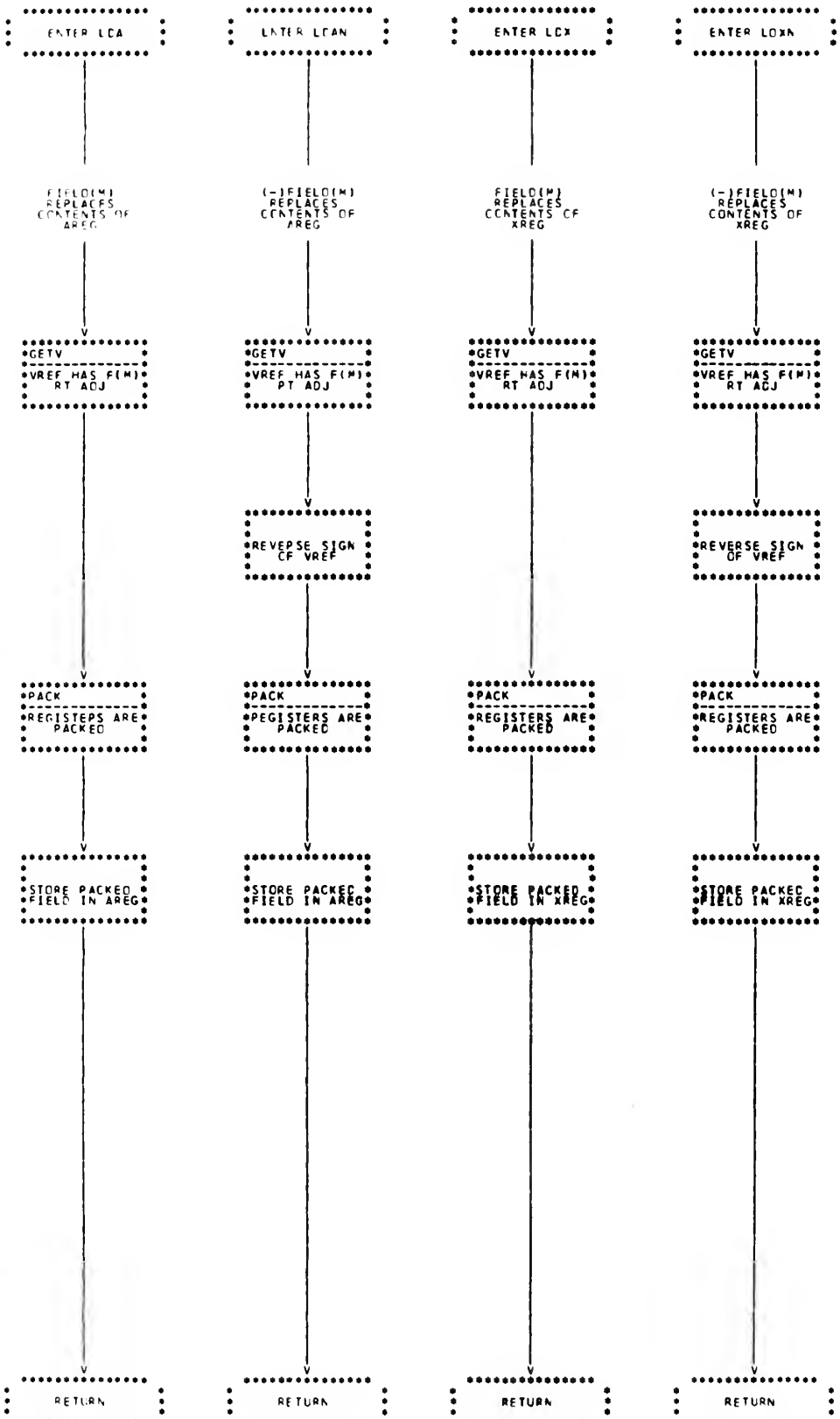


FLOWCHART SUBROUTINE SHIFT(SLC,SRC,SRAX,SLAX)

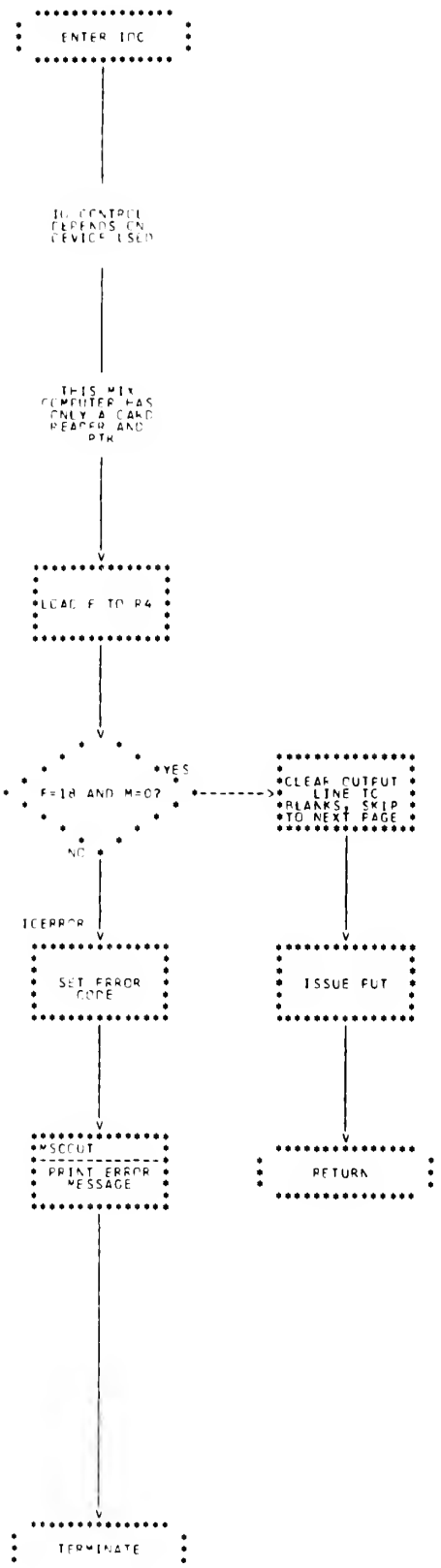


FLOWCHART OPERATOR SUBROUTINE MOVE

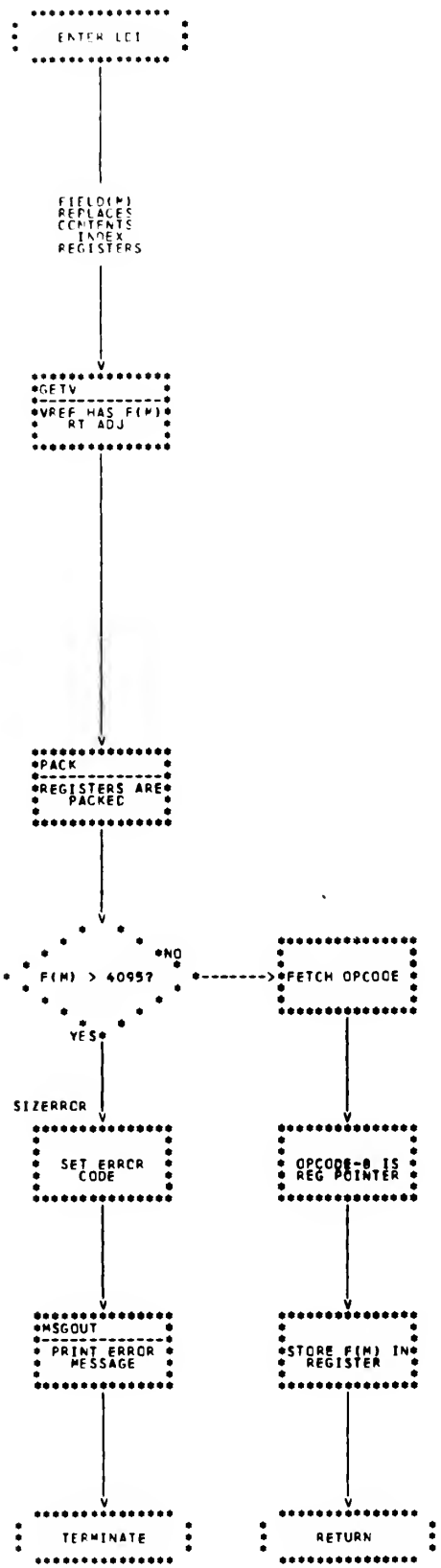



ENTER LCA
ENTER LCAN
ENTER LCX
ENTER LCXN
FIELD(M) REPLACES CONTENTS OF AREG
(-)FIELD(M) REPLACES CONTENTS OF AREG
FIELD(M) REPLACES CONTENTS OF XREG
(-)FIELD(M) REPLACES CONTENTS OF XREG
GETV
VREF HAS F(M) RT ADJ
REVERSE SIGN OF VREF
PACK
REGISTERS ARE PACKED
STORE PACKED FIELD IN AREG
STORE PACKED FIELD IN XREG
RETURN

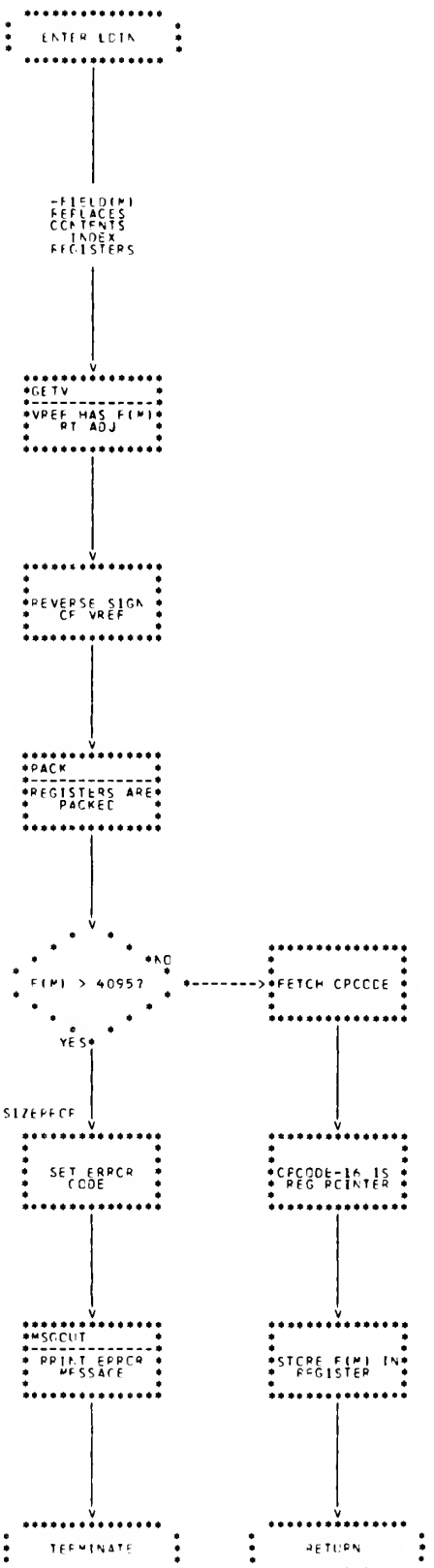
IO CONTROL OPERATOR SUBROUTINE



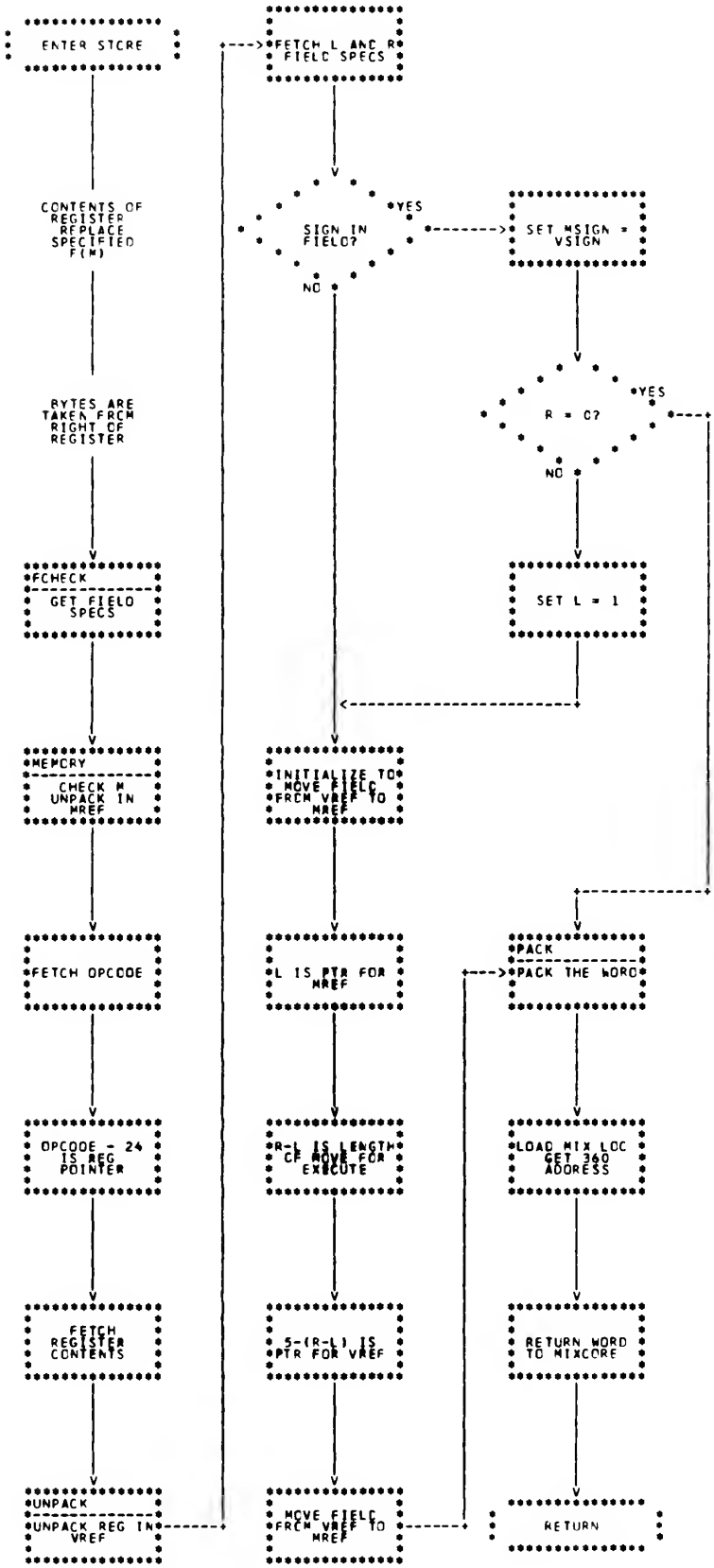
FLOWCHART SUBROUTINE LDI



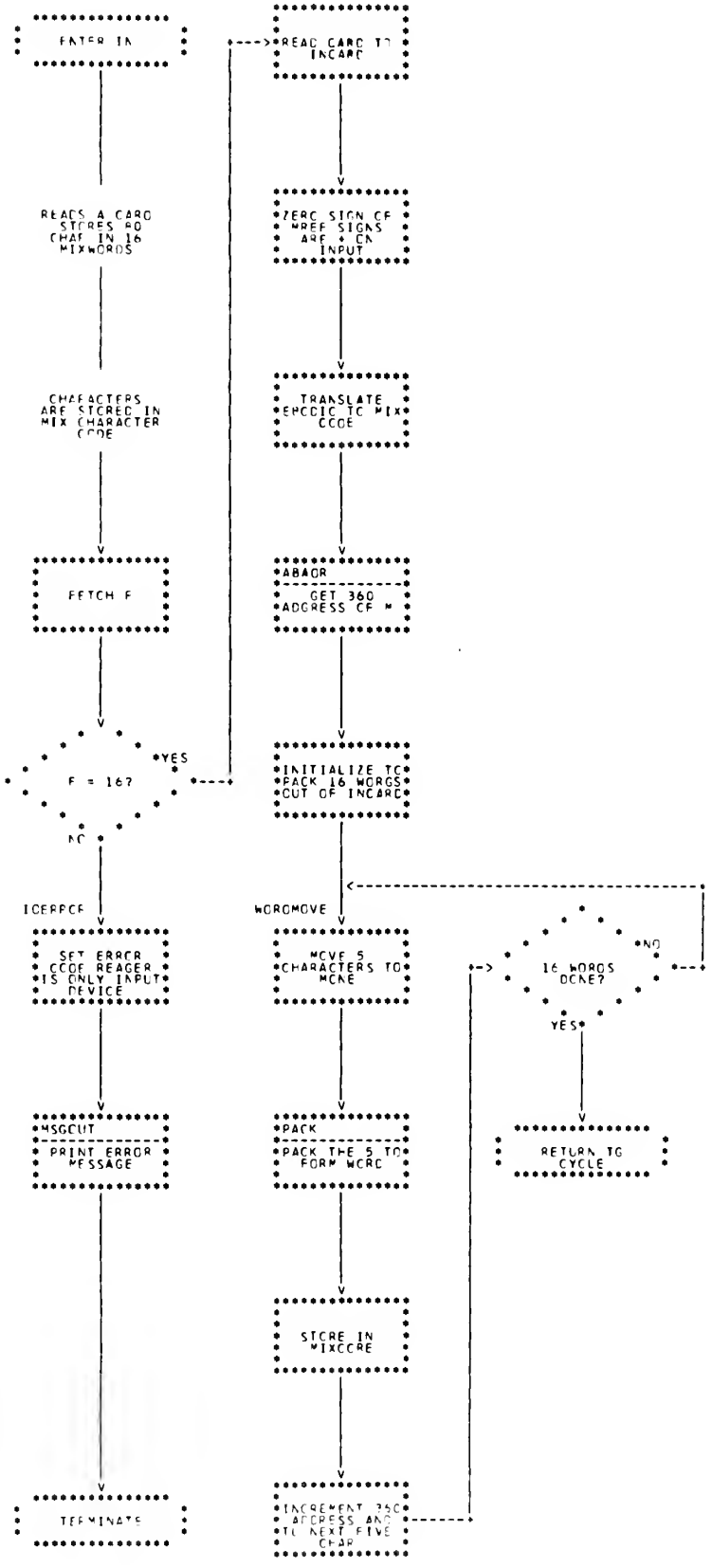
FLOWCHART SUBROUTINE LOIN



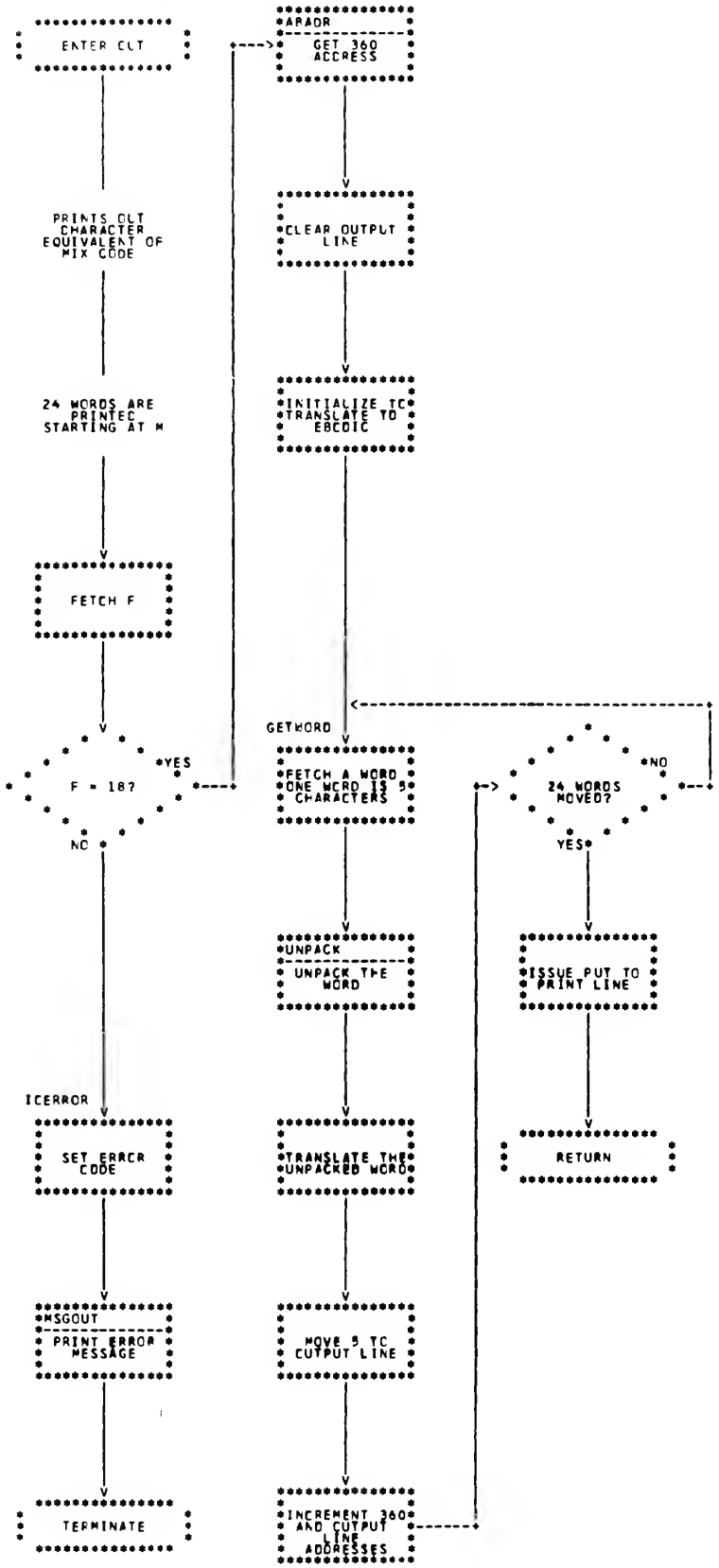
FLOWCHART SUBROUTINE STORE OPERATOR



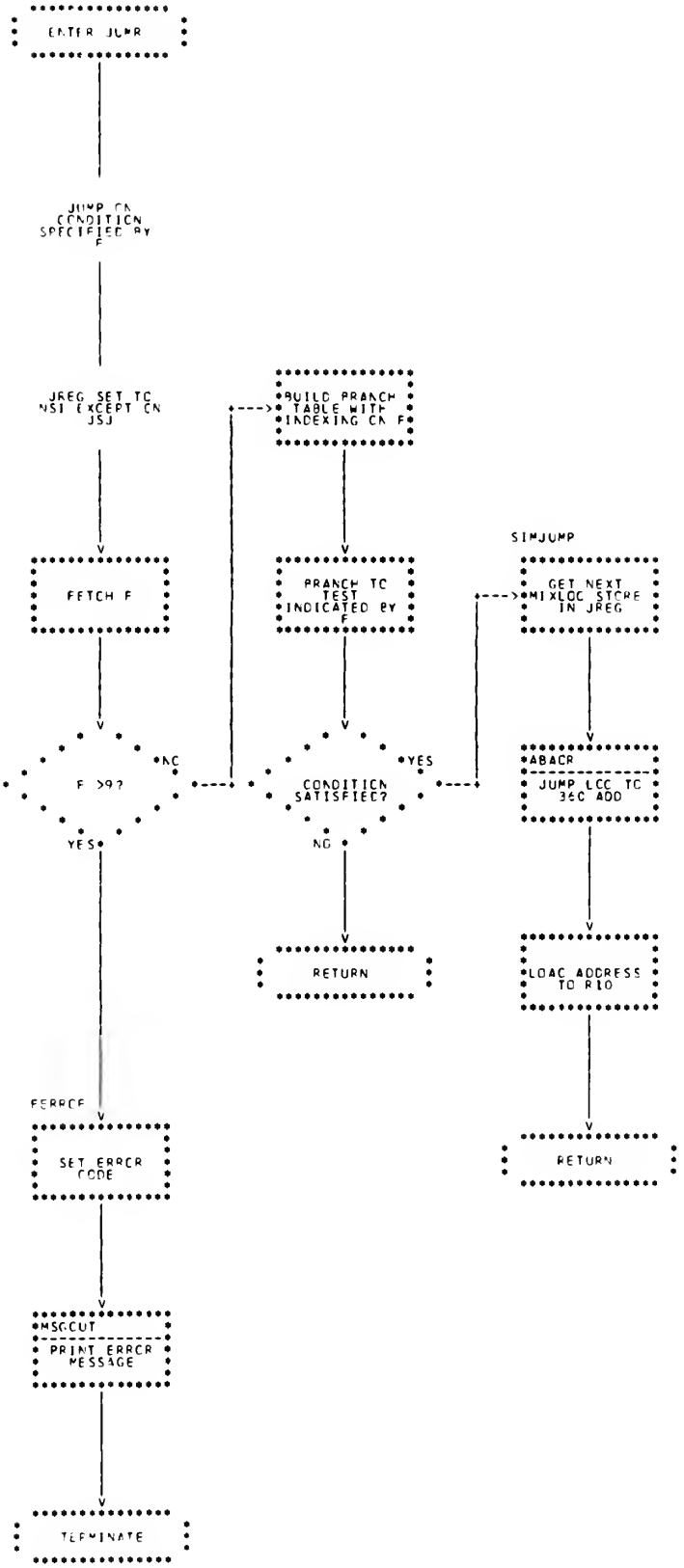
FLOWCHART SUBROUTINE IN OPERATOR



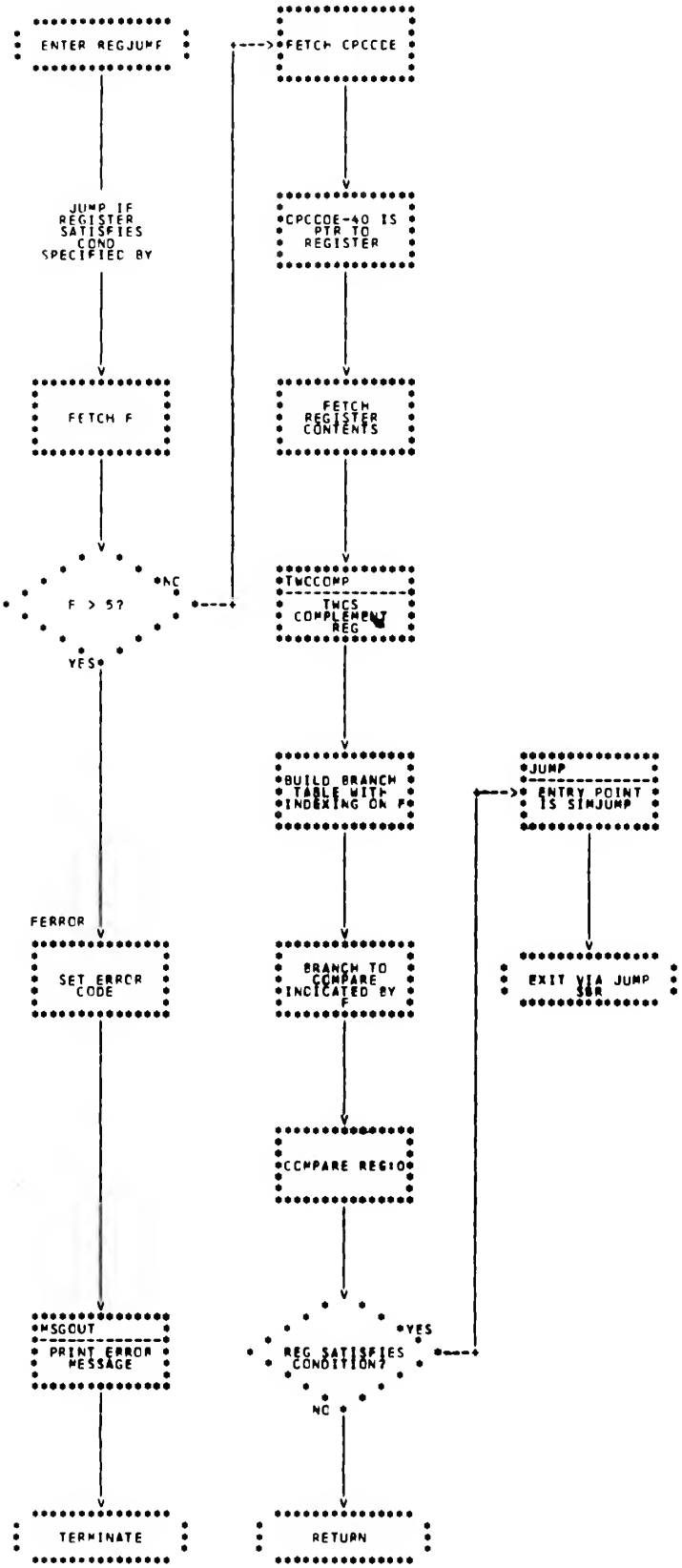
FLOWCHART SUBROUTINE CUT OPERATOR



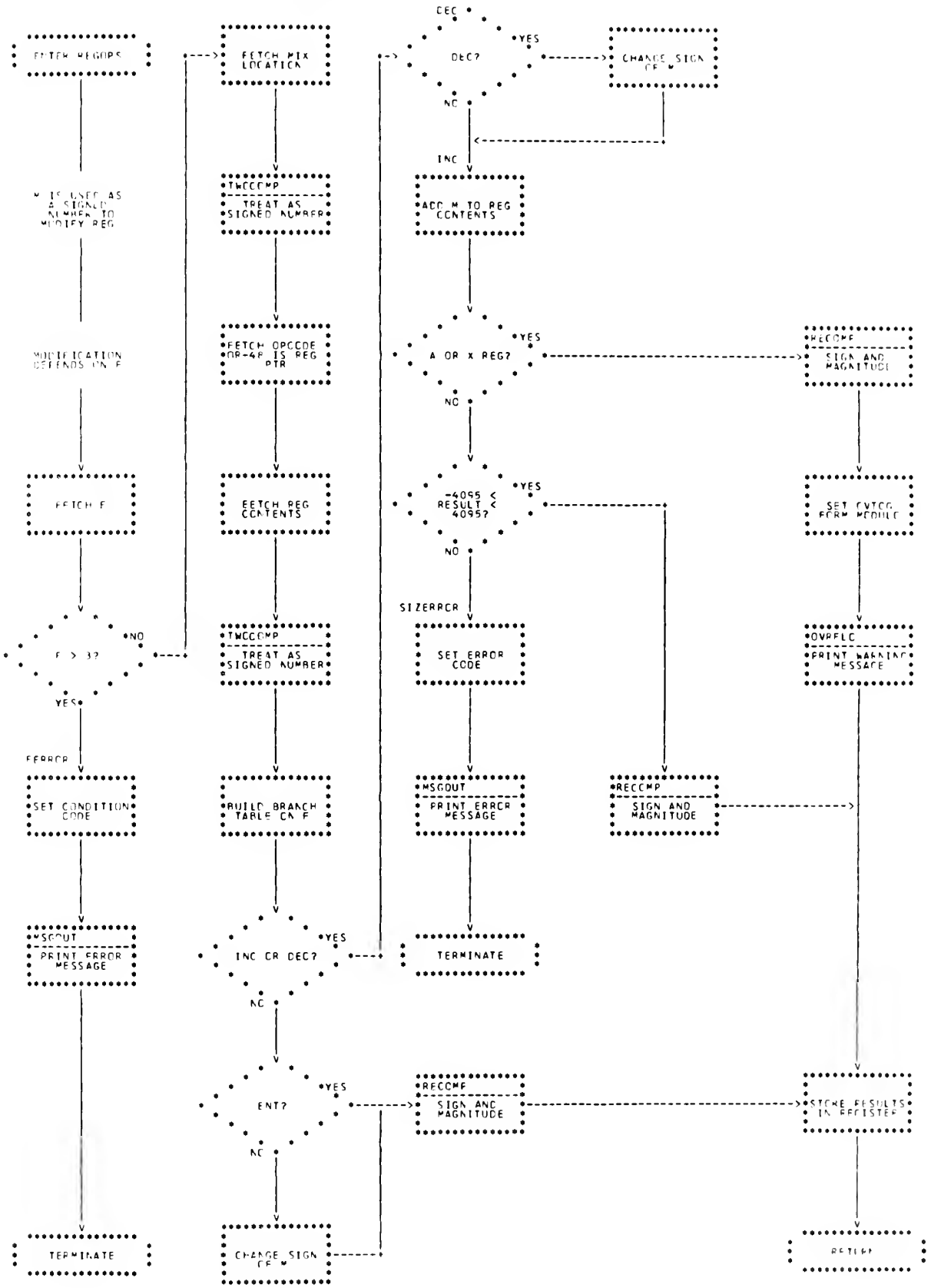
FLOWCHART SUBROUTINE JUMP OPERATOR



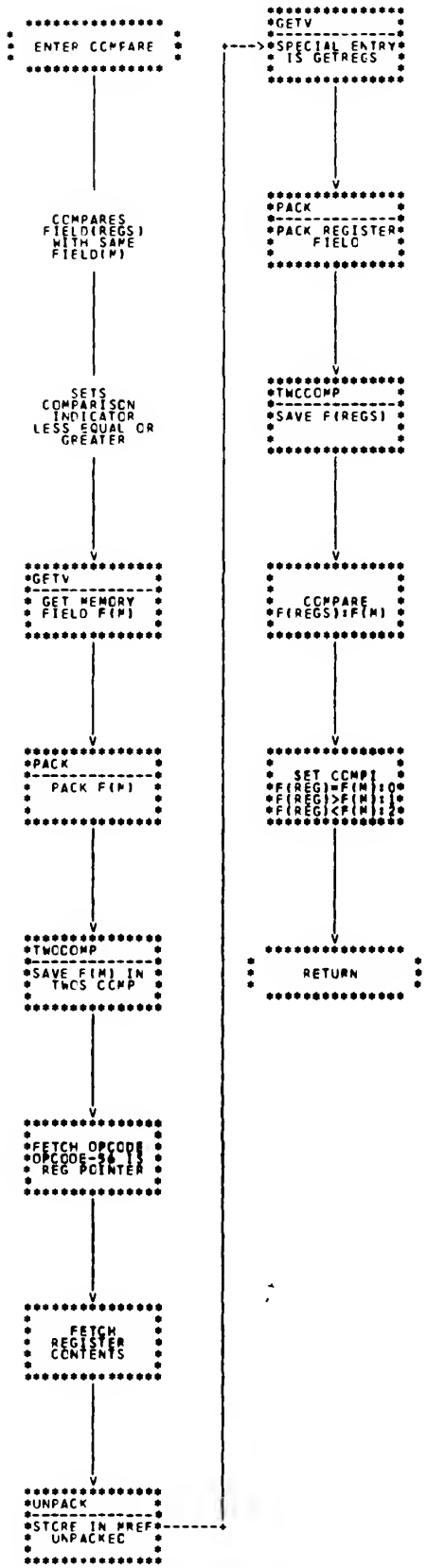
FLOWCHART SUBROUTINE REGJUMP OPERATORS



FLOWCHART SUBROUTINE REGOPS



FLOWCHART SUBROUTINE COMPARE OPERATOR



APPENDIX C

THE MIX SIMULATOR PROGRAM LISTING

The Mix Simulator program is shown in the assembled form.

**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                       4 *        ********************************************************************
                                       5 *        ***                                                              ***
                                       6 *        ***        INITIALIZE                                            ***
                                       7 *        ***                                                              ***
                                       8 *        ********************************************************************
000000                                 9 LOADER   CSECT
000000 90EC D00C              0000C   10 BEGIN    STM    R14,R12,12(R13)          SAVE REGISTERS
000004 182F                           11          LR     R2,R15
000006                                12          USING BEGIN,R2,R12
000006 41CC 2001              00001   13          LA     R12,BEGIN+1
00000A 41CC 0FFF              00FFF   14          LA     R12,4095(R12)
00000C                                15 R0       EQU    0
00000D                                16 R1       EQU    1
00000E                                17 R2       EQU    2
00000F                                18 R3       EQU    3
000010                                19 R4       EQU    4
000011                                20 R5       EQU    5
000012                                21 R6       EQU    6
000013                                22 R7       EQU    7
000014                                23 R8       EQU    8
000015                                24 R9       EQU    9
000016                                25 R10      EQU    10
000017                                26 R11      EQU    11
000018                                27 R12      EQU    12
000019                                28 R13      EQU    13
00001A                                29 R14      EQU    14
00001B                                30 R15      EQU    15
00001C 5800 C178              01178   31          L      R0,=F'20000'
                                      32          GETMAIN R,LV=(R0)
00001A 50D1 C004              00004   36          ST     R13,4(R1)          STORE OLD SA ADDRESS
00001E 5010 0008              00008   37          ST     R1,8(R13)          STORE ADDRESS OF NEW SAVE AREA
000022 18D1                           38          LR     R13,R1             NEW SA ADDRESS/BASE REG
000CCC                                39          USING STORAGE,R13
                                      40 *
                                      41 *        BUILD DCB'S
                                      42 *
000024 D25F D048 C188 0CC48 01188     43          MVC    MIXINPUT(96),DUMMY LOAD DCB IN CORE
00002A D25F D0A8 C188 00CA8 01188     44          MVC    MACHLIST(96),DUMMY LOAD THE DUMMY DCB FOR ALTERATION
00003C 4130 D0A8              000A8   45          LA     R3,MACHLIST
000000                                46          USING IHADCB,R3           BASE REG FOR DCB DSECT
000034 D2C7 3028 C0A8 00028 010A8     47          MVC    DCBDDNAM(8),=C'MACHLIST'
00003A D201 303E C17C 0003E 0117C     48          MVC    DCBBLKSI(2),=H'133'
00004C D200 3024 C186 00024 01186     49          MVC    DCBRECFM(1),=B'10000100'
000046 D201 3052 C17C 0CC52 0117C     50          MVC    DCBLRECL(2),=H'133'
00004C D201 3032 C17E 0C032 0117E     51          MVC    DCBMACR(2),=B'0000000001010000'
                                      52          DROP   R3
                                      53 *
                                      54 *        INITIALIZE INTERPRET SWITCH
                                      55 *
000052 9200 D278              00278   56          MVI    SWITCH,B'000C0000'  PROGRAM WILL INTERPRET IF SW= ZEROS
                                      57 *        BUILD OPEN PARM LIST
000056 4130 D048              00048   58          LA     R3,MIXINPUT
00005A 503C D1C8              00108   59          ST     R3,PARMLIST
00005E 4130 D0A8              000AB   60          LA     R3,MACHLIST
000062 503C D10C              0010C   61          ST     R3,PARMLIST+4
000066 928F D10C              0010C   62          MVI    PARMLIST+4,X'8F'
00006A 4110 D108              00108   63          LA     R1,PARMLIST
                                      64          OPEN   ,MF=(E,(R1))
                                      67 *
                                      68 *        SKIP TO TOP OF PAGE FOR OUTPUT LISTING
```

**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

```
  LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                      69 *
000072 C283 D161 C21E 0C161 0121E     70          MVC   MIXOUT+1(132),MBLANK     CLEAR OUTPUT LINE
000078 D223 D17B C2A2 0017B 012A2     71          MVC   MIXOUT+27(36),HEADER     MOVE IN HEADER
00007E 92F1 D160      0C160           72          MVI   MIXOUT,C'1'             SKIP TO TOP OF PAGE FOR LISTING
                                      73          PUT   MACHLIST,MIXOUT
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                      79 *         ********************************************************************
                                      80 *         ***                                                              ***
                                      81 *         ***        READ INPUT CARD                                       ***
                                      82 *         ***                                                              ***
                                      83 *         ********************************************************************
000090 9240 D160       0C160          84 READCARD MVI   MIXOUT,C' '
000094 D24F D161 D16C 00161 00160     85          MVC   MIXOUT+1(80),MIXOUT
00009A 92F0 D160       0C160          86          MVI   MIXOUT,C'0'
                                      87          GET   MIXINPUT,INCARD
                                      92 *         CONVERT WORD COUNT AND LOCATION TO BINARY
0000AC F270 D4B8 D115 004B8 00115     93          PACK  PCOUNT,WCOUNT
0000B2 4F30 D4B8       004B8          94          CVB   R3,PCOUNT
0000B6 F273 D4C0 D116 004C0 00116     95          PACK  PLOC,LOC            LOCATION TO BINARY IN R5
0000BC 4F50 D4C0       004C0          96          CVB   R5,PLOC
0000C0 D24F D161 D110 0C161 00110     97          MVC   MIXOUT+1(80),INCARD
0000C6 95F0 D115       0C115          98          CLI   WCOUNT,C'0'
0000CA 4780 2298       00298          99          BC    8,TRANCARD
                                     100 *
                                     101 *         PRINT CARD IMAGE WITHOUT SIGN OVERPUNCH IN COL 20,30,...,80
                                     102 *
0000CE 414C D16A        0016A        103          LA    R4,MIXOUT+10        INITIALIZE FOR WORD LOOP
0000D2 4160 000A        0000A        104 NEXTWORD LA    R6,10               INITIALIZE FOR DIGIT LOOP
0000D6 414C 4001        00001        105 NOTBLK   LA    R4,1(0,R4)          INCREMENT ADDRESS TO NEXT COL
0000DA 9540 4000       00000         106          CLI   0(R4),C' '          IF COL IS BLANK,NOT A WORD
0000DE 4780 20EE        000EE        107          BC    8,BLANK
0000E2 466C 20D6        000D6        108          BCT   R6,NOTBLK           IO COL TO CHECK
0000E6 96F0 4000       C0000         109          CI    0(R4),X'F0'         MOVE IN PLUS ZONE
0000EA 47F0 20D2        000D2        110          BC    15,NEXTWORD
                                     111 BLANK    PUT   MACHLIST,MIXOUT
                                     116 *
                                     117 *         NOTE: OTHER COLUMNS ARE PRINTED ACCORDING TO EBCDIC CODE
                                     118 *
                                     119 *
                                     120 *         UNDERSCORE COL 20,30,....,80 IF OVERPUNCHED
                                     121 *         R4 HAS THE ADDRESS OF FIRST BLANK ON CARD REL TO OUTPUT LINE
                                     122 *
0000FC 9240 D16C       0C160         123          MVI   MIXOUT,C' '
000100 D24F D161 D160 00161 00160    124          MVC   MIXOUT+1(80),MIXOUT
000106 924E D160       0C160         125          MVI   MIXOUT,C'+'
00010A 4160 D16A        0016A        126          LA    R6,MIXOUT+10
00010E 4180 D119        00119        127          LA    R8,INCARD+9
000112 4180 8C0A        0000A        128 NEXTCOLM LA    R8,10(0,R8)         ADDRESS COL 20, 30,....,80 CARD IMAGE
000116 4160 600A        0000A        129          LA    R6,10(0,R6)         ADDRESS COL 20, 30,...,80 OUTPUT LINE
00011A 1964                          130          CR    R6,R4
00011C 47A0 2138        00138        131          BC    10,PRUNSC           COMMENTS BEYOND HERE
000120 9120 8000       00000         132          TM    0(R8),X'20'
000124 4710 2112        00112        133          BC    1,NEXTCOLM          NOT MINUS
000128 91D0 8000       00C00         134          TM    0(R8),X'D0'
00012C 47C0 2112        00112        135          BC    12,NEXTCOLM         NOT MINUS
000130 926D 6000       00C00         136          MVI   0(R6),C' '
000134 47F0 2112        00112        137          BC    15,NEXTCOLM
                                     138 PRUNSC   PUT   MACHLIST,MIXOUT
```

**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     144 *        ********************************************************************
                                     145 *        ***                                                              ***
                                     146 *        ***       EDIT CHECKS                                            ***
                                     147 *        ***                                                              ***
                                     148 *        ********************************************************************
000146 416C 0F9F             00F9F   149          LA    R6,3999
00014A 1956                          150          CR    R5,R6
00014C 4720 C042             01042   151          BH    EMSGA              LOCATION IS TOO LARGE
000150 416C 0000             00000   152          LA    R6,0
000154 1956                          153          CR    R5,R6
000156 4740 C042             01042   154          BL    EMSGA              LOCATION IS TOO SMALL
                                     155 *
                                     156 *        WORD COUNT CHECK
                                     157 *
00015A 416C 00C7             00007   158 WORDCHEK LA    R6,7
00015E 1936                          159          CR    R3,R6
000160 4720 CC4E             0104E   160          BH    EMSGB              WORD COUNT IS TOO LARGE
000164 4160 0000             00000   161          LA    R6,0
000168 1936                          162          CR    R3,R6
00016A 4740 C04E             0104E   163          BL    EMSGB              WORD COUNT IS TOO SMALL
                                     164 *
                                     165 *        WORD COUNT/WORDS ON CARD CONSISTENCY CHECK
                                     166 *        WORD COUNT MUST BE VALID FOR THIS CHECK TO BE CONDUCTED
                                     167 *        R4  STILL HAS ADDRESS OF FIRST BLANK
                                     168 *
00016E 1873                          169          LR    R7,R3              KEEP WORD COUNT IN R3
000170 4160 D16B             0016B   170          LA    R6,MIXOUT+11       INITIALIZE ADDRESS
000174 4160 600A             0000A   171 COUNTCN  LA    R6,10(0,R6)        COL BEYOND LAST WORD POINTER
000178 467C 2174             00174   172          BCT   R7,COUNTCN
00017C 1946                          173          CR    R4,R6
00017E 4770 C05A             0105A   174          BC    7,EMSGC            WORD/WORD COUNT NOT CONSISTENT
                                     175 *
                                     176 *        CHECK COLUMNS FOR VALID ZONES
                                     177 *        F IS VALID ZONE ALL PUNCHED COLUMNS
                                     178 *        D IS VALID ZONE COL 20,30,...80 IF PUNCHED
                                     179 *
000182 4140 D114             00114   180          LA    R4,INCARD+4        R4 WILL ADDRESS EACH COLUMN
000186 4180 C005             00005   181          LA    R8,5
00018A 417C 0C14             00014   182          LA    R7,20              COL 20, 30,...,80 INDICATOR
00018E 41B0 0001             00001   183          LA    R11,1
000192 419C 000A             0000A   184          LA    R9,10              R9 HAS CONSTANT TEN
000196 4140 4001             00001   185 NEWCOLM  LA    R4,1(0,R4)         ADDRESS THE COLUMN
00019A 1A8B                          186          AR    R8,R11
00019C 9540 40C0      00000          187          CLI   0(R4),C' '         COMMENTS ONLY AFTER FIRST BLANK
0001A0 4780 21CC             001CC   188          BC    8,LOADUM
0001A4 1987                          189          CR    R8,R7              COL 20, 30,...,80?
0001A6 4780 21B6             00186   190          BC    8,CKMINUS
0001AA 91F0 4000      C0000          191 CKPLUS   TM    0(R4),X'F0'
0001AE 47C0 C066             01066   192          BC    12,EMSGC           ITS NOT A VALID ZONE
0001B2 47F0 2196             00196   193          BC    15,NEWCOLM
0001B6 1A79                          194 CKMINUS  AR    R7,R9
00C1B8 912C 40C0      C0000          195          TM    0(R4),X'20'
00C1BC 471C 21AA             001AA   196          BC    1,CKPLUS
00C1C0 91DC 4000      C0000          197          TM    0(R4),X'D0'
0001C4 47C0 21AA             001AA   198          BC    12,CKPLUS
0001C8 47FC 2196             00196   199          BC    15,NEWCOLM
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     201 *        ****************************************************************
                                     202 *        ***                                                          ***
                                     203 *        ***       LOAD MIX CORE                                      ***
                                     204 *        ***                                                          ***
                                     205 *        ****************************************************************
                                     206 *        INITIALIZE TO PRINT WORD AND MOVE TO CORE
                                     207 *        R3 CONTAINS WORD COUNT,R5 CONTAINS LOCATION
                                     208 *
0001CC 45BC C034             01034   209 LOADUM   BAL   R11,MIXADR
                                     210 *
                                     211 *        FORM OUTPUT LINE: LOC...XXXX  AAAA  II  FF  CC
                                     212 *
0C01CC 4140 D110             00110   213          LA    R4,INCARD
0001D4 4140 400A             0000A   214 LOCCUT   LA    R4,10(0,R4)          ADDRESS THE WORD
0001D8 9240 D160       CC160         215          MVI   MIXOUT,C' '          CLEAR OUTPUT LINE
0001DC D24F D161 D160  00161 00160   216          MVC   MIXOUT+1(80),MIXOUT  CLEAR OUTPUT LINE
0001E2 D205 D164 C180  0C164 01180   217          MVC   MIXOUT+4(6),=C'LOC...'
0001E8 D203 D16A D116  0C16A 00116   218          MVC   MIXOUT+10(4),LOC     LOC VALUE TO OUTPUT LINE
0001EE D203 D171 4000  00171 00000   219          MVC   MIXOUT+17(4),0(R4)   ADDRESS TO OUTPUT LINE
0001F4 D201 D177 4CC4  00177 00004   220          MVC   MIXOUT+23(2),4(R4)   INDEX TO OUTPUT LINE
0001FA D201 D17B 4006  0017B 00006   221          MVC   MIXOUT+27(2),6(R4)   FIELD TO OUTPUT LINE
0C0200 D201 D17F 40C8  0C17F 00008   222          MVC   MIXOUT+31(2),8(R4)   OP TO OUTPUT LINE
                                     223 *
                                     224 *        PLACE SIGN IN FRONT OF ADDRESS
                                     225 *
0C0206 91F0 4009       0CC09         226          TM    9(R4),X'F0'
00C2CA 4710 2216             00216   227          BC    1,ZONEOK             ZONE IS PLUS
00020E 96F0 D180       C0180         228          OI    MIXOUT+32,X'F0'      PLUS IN LAST DIGIT ZONE
000212 9660 D170       0C170         229          OI    MIXOUT+16,X'60'      MINUS IN FRONT OF ADDRESS
                                     230 ZONEOK   PUT   MACHLIST,MIXOUT
000224 45E0 2404             00404   235          BAL   R14,MAGCHECK
                                     236 *
                                     237 *        CONVERT THE FIELDS TO BINARY AND LOAD MIX CORE
                                     238 *
000228 95FF D278       00278         239          CLI   SWITCH,B'11111111'   CEASE LOAD IF ERRORS EXISTS
00022C 478C 2280             00280   240          BC    8,NXLOC
000230 F271 D4C8 D17F  004C8 0017F   241          PACK  PACKAREA,MIXOUT+31(2)   PACK THE OP CODE
00023E 4F60 D4C8             004C8   242          CVB   R6,PACKAREA          OP CODE TO BINARY
00023A 8E60 0006             00006   243          SRDA  R6,6                 SHIFT OP COCE INTO R7
00023E F271 D4C8 D17B  004C8 0017B   244          PACK  PACKAREA,MIXOUT+27(2)   PACK THE FIELD
000244 4F60 D4C8             004C8   245          CVB   R6,PACKAREA          FIELD TO BINARY
00C248 8E60 C006             00006   246          SRDA  R6,6                 SHIFT F INTO R7
00024C F271 D4C8 D177  004C8 00177   247          PACK  PACKAREA,MIXOUT+23(2) PACK THE INDEX
000252 4F60 D4C8             004C8   248          CVB   R6,PACKAREA
000256 8E60 0006             00006   249          SRDA  R6,6                 SHIFT I INTO R7
00025A F273 D4C8 D171  004C8 00171   250          PACK  PACKAREA,MIXOUT+17(4)   PACK THE ADDRESS
000260 4F60 D4C8             004C8   251          CVB   R6,PACKAREA
000264 8E6C 000D             0000D   252          SRDA  R6,13                ALL FIELDS TO R7
000268 9560 D170       00170         253          CLI   MIXOUT+16,X'60'
00026C 4770 2274             00274   254          BC    7,ADPLUS             ADDRESS NOT MINUS
000270 4160 0001             00001   255          LA    R6,1                 MINUS IN R6
000274 8E60 0001             00001   256 ADPLUS   SRDA  R6,1                 R7 HAS THE WORD
000278 507C A000             00000   257          ST    R7,0(0,R10)          STORE WORD IN MIXCORE
00027C 41A0 A004             00004   258          LA    R10,4(0,R10)         INCREMENT MIXCORE ADDRESS
000280 FA70 C4C0 C187  004C0 01187   259 NXLOC    AP    PLOC,=P'1'
000286 F337 D116 D4CC  00116 004C0   260          UNPK  LOC(4),PLOC
00028C 96F0 D119       CC119         261          OI    LOC+3,X'F0'
00C290 4630 21D4             001D4   262          BCT   R3,LOCOUT
000294 47F0 2090             00090   263          B     READCARD
                                     264 TRANCARD PUT   MACHLIST,MIXOUT
```

```
  LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     270 *        ****************************************************************
                                     271 *        ***                                                          ***
                                     272 *        ***       TEST INTERPRET SWITCH,END OF FILE                  ***
                                     273 *        ***                                                          ***
                                     274 *        ****************************************************************
                                     275 *        PASS CONTROL TO INTERPRET ROUTINE WITH ADDRESS OF STARTING
                                     276 *        LOCATION IN R10
                                     277 *
000C2A6 9500 D278        000278      278          CLI   SWITCH,B'00000000'
000C2AA 4780 2452              00452 279          BE    STEPINT
                                     280 *
                                     281 *        END OF FILE
                                     282 *
000C2AE D284 D161 C21E 00161 0121E   283          MVC   MIXOUT+1,MBLANK
000C2B4 D22B D160 C3D5 000160 013D5  284          MVC   MIXOUT(44),MSGF
                                     285          PUT   MACHLIST,MIXOUT
000C2C8 47F0 2310              00310 290          B     ABECF
000C2CC D284 D161 C21E 00161 0121E   291 MYECF    MVC   MIXOUT+1,MBLANK
000C2D2 D238 D160 C401 000160 01401  292          MVC   MIXOUT(57),MSGG
                                     293          PUT   MACHLIST,MIXOUT
000C2E6 47F0 2310              00310 298          B     ABECF
000C2EA 5840 02B0              002B0 299 ENDJOB   L     R4,CLOCK              CLOCK TIME
000C2EE 4E40 04C8              004C8 300          CVD   R4,PACKAREA          CONVERT FOR OUTPUT
000C2F2 D277 D160 C0BC 000160 010B0  301          MVC   MIXOUT(120),=CL120' EXECUTION TIME UNITS:'
000C2F8 F3F7 D176 04C8 000176 004C8  302          UNPK  MIXOUT+22(16),PACKAREA
000C2FE 96F0 D185              00185 303          OI    MIXOUT+37,X'F0'
                                     304          PUT   MACHLIST,MIXOUT
000C31C 928C D1CC              000CC 309 ABECF    MVI   PARMLIST+4,X'80'
000C314 411C D1C8              00108 310          LA    R1,PARMLIST
                                     311          CLOSE ,MF=(E,(R1))
000C31C 58DD 00C4              00004 314          L     R13,4(R13)            RESTORE SAVEAREA ADDRESS
                                     315          RETURN (14,12)
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     319 *        ********************************************************************
                                     320 *        ***                                                              ***
                                     321 *        ***       MSGOUT-ERROR MESSAGE WRITER                            ***
                                     322 *        ***                                                              ***
                                     323 *        ********************************************************************
000326 900F D3F8              003F8  324 MSGOUT   STM   R0,R15,TWOSAVE
00032A D284 D1E6 C21E 001E6 0121E  325           MVC   MSGLINE+1,MBLANK
000330 D208 D1E5 C212 001E5 01212  326           MVC   MSGLINE(12),ERRHDR      CARRIAGE CO TROL AND HEADER
000336 414C C080              0080 327           LA    R4,128                  MAXIMUM LOADER ERROR CODE
00033A 19F4                        328           CR    R15,R4
00033C 4720 238E              038E 329           BH    EXERR
000340 4130 2376              0376 330           LA    R3,GETOUT
000344 89F0 0001              0001 331           SLL   R15,1
000348 47FF 2348              0348 332           B     *(R15)
00034C 470C 2376              0376 333           NOP   GETOUT
000350 D233 D1F2 C2C6 0C1F2 012C6  334           MVC   MSGLINE+13(52),MSGA
000356 07F3                        335           BR    R3
000358 D22E D1F2 C2FA 0C1F2 012FA  336           MVC   MSGLINE+13(47),MSGB
00035E 07F3                        337           BR    R3
000360 D235 D1F2 C329 0C1F2 01329  338           MVC   MSGLINE+13(54),MSGC
000366 07F3                        339           BR    R3
000368 D228 D1F2 C35F 0C1F2 0135F  340           MVC   MSGLINE+13(41),MSGD
00036E 07F3                        341           BR    R3
000370 D24C D1F2 C388 0C1F2 01388  342           MVC   MSGLINE+13(77),MSGE
                                   343 GETOUT    PUT   MACHLIST,MSGLINE
000384 96FF D278              0278 348           OI    SWITCH,B'11111111'
000388 980F D3F8              03F8 349           LM    R0,R15,TWOSAVE
00038C 07FE                        350           BR    R14
00038E 18F4                        351 EXERR     SR    R15,R4                  MIN ERR CODE IS NOW 4
000390 89F0 0001              0001 352           SLL   R15,1
000394 4130 23F2              03F2 353           LA    R3,ENDIT
000398 4140 D4F0              04F0 354           LA    R4,MIXCORE
00039C 18A4                        355           SR    R10,R4                  MIXLOC TIMES 4
00039E 88A0 0002              0002 356           SRL   R10,2
0003A2 414C 0C01              0001 357           LA    R4,1
0003A6 18A4                        358           SR    R10,R4
0003A8 4EA0 D4C8              04C8 359           CVD   R10,PACKAREA
0003AC D21D D1F2 C1F4 0C1F2 011F4  360           MVC   MSGLINE+13(30),MLOC
0003B2 F332 D1F7 D4CD 001F7 004CD  361           UNPK  MSGLINE+18(4),PACKAREA+5(3)
0003B8 96F0 D1FA              01FA 362           OI    MSGLINE+21,X'F0'
0003BC 47FF 23BC              03BC 363           B     *(R15)
0003C0 4700 23F2              03F2 364           NOP   ENDIT
0003C4 D216 D212 C43A 00212 0143A  365           MVC   MSGLINE+45(23),MSGAX
0003CA 07F3                        366           BR    R3
0003CC D21A D212 C451 0C212 01451  367           MVC   MSGLINE+45(27),MSGBX
0003D2 07F3                        368           BR    R3
0003D4 D228 D212 C46C 0C212 0146C  369           MVC   MSGLINE+45(41),MSGCX
0003DA 07F3                        370           BR    R3
0003DC C212 D212 C495 0C212 01495  371           MVC   MSGLINE+45(19),MSGDX
0003E2 07F3                        372           BR    R3
0003E4 C242 D212 C4A8 0C212 014A8  373           MVC   MSGLINE+45(67),MSGEX
0003EA 07F3                        374           BR    R3
0003EC D215 D212 C4E8 0C212 014E8  375           MVC   MSGLINE+45(22),MSGFX
                                   376 ENDIT     PUT MACHLIST,MSGLINE
CC04CC 47F0 22EA              02EA 381           B     ENDJOB
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     383 *        ****************************************************************
                                     384 *        ***                                                          ***
                                     385 *        ***       MAGCHECK-CHECKS BYTESIZE ON INPUT                  ***
                                     386 *        ***                                                          ***
                                     387 *        ****************************************************************
00C4C4 50E0 D4D0                     388 MAGCHECK ST    R14,AUXSAVE
00C4C8 D5C1 D17F C1E8 00C7F 011E8    389           CLC   MIXOUT+31(2),BYTESIZE
00C40E 47D0 241A                     390           BC    13,CPCK
00C412 41F0 CC14                     391           LA    R15,20                 ERROR CODE
000416 45E0 2326                     392           BAL   R14,MSGOUT
00C41A D501 D178 C1E8 00C7B 011E8    393 CPCK      CLC   MIXOUT+27(2),BYTESIZE
00C420 47D0 242C                     394           BC    13,FCK
00C424 41F0 0014                     395           LA    R15,20                 ERROR CODE
000428 45E0 2326                     396           BAL   R14,MSGOUT
00042C D501 D177 C1E8 00C77 011E8    397 FCK       CLC   MIXOUT+23(2),BYTESIZE
00C432 47DC 243E                     398           BC    13,ICK
00C436 41F0 C014                     399           LA    R15,20                 ERROR CODE
00043A 45E0 2326                     400           BAL   R14,MSGOUT
00043E D503 D171 C1EA 00C71 011EA    401 ICK       CLC   MIXOUT+17(4),ADMAX
000444 47D0 244C                     402           BC    13,ADCK
00C448 41F0 0014                     403           LA    R15,20                 ERROR CODE
00044C 58E0 D4D0                     404 ADCK      L     R14,AUXSAVE
00045C 07FE                          405           BR    R14
                                     406 *
000452 4580 CC34                     407 STEPINT   BAL   R11,MIXADR
                                     408 *         INITIALIZE CLOCK,TOGGLE,COMPI,ETC
000456 9200 D27A       0027A         409           MVI   COMPI,X'00'
00045A 9200 D279       00279         410           MVI   CVTOG,X'00'
00045E 1B44                          411           SR    R4,R4
000460 5040 D2AC                     412           ST    R4,ZERO                ZERO FOR STZ
000464 5040 D2B0                     413           ST    R4,CLOCK
                                     414 *
                                     415 *         FETCH AND UNPACK THE INSTRUCTION
```

**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                   417 *        ********************************************************************
                                   418 *        ***                                                              ***
                                   419 *        ***       CYCLE-CENTRAL SCAN ROUTINE                             ***
                                   420 *        ***                                                              ***
                                   421 *        ********************************************************************
000468 584C A000           00000   422 CYCLE    L      R4,0(0,R10)          FETCH INST
00046C 41AC A004           00004   423          LA     R10,4(0,R10)         NSI 360 ADDRESS
000470 8E40 0012           00012   424          SRDA   R4,18
000474 1894                        425          LR     R9,R4                SAVE MIX LOC
000476 1B44                        426          SR     R4,R4
000478 8D4C 0006           00006   427          SLDL   R4,6                 INDEX TO R4
00047C 4240 D4DD           004DD   428          STC    R4,BYTETHRE          STORE IT
000480 1884                        429          LR     R8,R4                SAVE INDEX
000482 1B44                        430          SR     R4,R4                ZERO
000484 8D4C 0006           00006   431          SLDL   R4,6                 F TO R4
000488 4240 D4DE           004DE   432          STC    R4,BYTEFOUR          STORE IT
00048C 1B44                        433          SR     R4,R4                ZERO
00048E 8D4C 0006           00006   434          SLDL   R4,6                 OP TO R4
000492 424C D4DF           004DF   435          STC    R4,BYTEFIVE          STORE IT
000496 417C 0006           00006   436          LA     R7,6
00049A 1987                        437          CR     R8,R7                ILLEGAL INDEX?
00049C 472C C072           01072   438          BH     IXERROR
0004A0 1B77                        439          SR     R7,R7                A ZERO
0004A2 1987                        440          CR     R8,R7                INDEX ZERO?
0004A4 4780 24DC           004DC   441          BC     8,INDXZERO
0004A8 18F9                        442          LR     R15,R9
0004AA 45E0 25C6           005C6   443          BAL    R14,TWOCOMP
0004AE 189F                        444          LR     R9,R15               TWO'S COMPLEMENT
0004B0 8980 0002           00002   445          SLL    R8,2                 INDEX TIMES 4
0004B4 5888 D288           00288   446          L      R8,AREG(R8)          R8 HAS CONTENTS IREG
0004B8 18F8                        447          LR     R15,R8
0004BA 45E0 25C6           005C6   448          BAL    R14,TWOCOMP
0004BE 188F                        449          LR     R8,R15               TWO'S COMPLEMENT
0004CC 1A98                        450          AR     R9,R8                EFFECTIVE ADDRESS
0004C2 4130 0FFF           00FFF   451          LA     R3,4095
0004C6 1993                        452          CR     R9,R3
0004C8 4720 C07A           0107A   453          BH     ADRERROR             +AA > 2 BYTES
0004CC 1173                        454          LNR    R7,R3
0004CE 1997                        455          CR     R9,R7
0004D0 4740 C07A           0107A   456          BL     ADRERROR             -AA > 2 BYTES
0004C4 18F9                        457          LR     R15,R9
000406 45E0 25DE           005DE   458          BAL    R14,RECOMP
0004DA 189F                        459          LR     R9,R15               SIGN AND MAGNITUDE
00040C 5090 D4D8           004D8   460 INDXZERO ST     R9,MIXLOC            AA SIGN AND MAG
0004E0 1874                        461          LR     R7,R4
00C4E2 8970 0002           00002   462          SLL    R7,2                 TIMES 4
0004E6 5877 C604           01604   463          L      R7,OPTABLE(R7)       OPERAND ADDRESS AND SIGN
0004EA 1866                        464          SR     R6,R6
0004EC 8D6C 0008           00008   465          SLDL   R6,8                 XTIME IN R6
0004F0 5A60 D2B0           002B0   466          A      R6,CLOCK             INC CLOCK TIME
0004F4 5060 D2B0           002B0   467          ST     R6,CLOCK
00C4F8 8870 0008           00008   468          SRL    R7,8                 REXTORE ADDRESS
0004FC 47FC 7000           00000   469          BC     15,0(0,R7)           BRANCH TO OPERATOR
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    471 *        ********************************************************************
                                    472 *        ***                                                              ***
                                    473 *        ***        GETV                                                  ***
                                    474 *        ***                                                              ***
                                    475 *        ********************************************************************
                                    476 *        GETV  V=F(M)  THIS SR FINDS THE QUANTITY V USED IN M
                                    477 *        MIX OPERATORS AND STORES IT RIGHT ADJUSTED IN VREF.
                                    478 *        M IS IN UNPACKED FORMAT.
 CC05CC 900F D338            00338  479 GETREGV  STM   R0,R15,GETSAVE
 000504 47F0 2514            00514  480          BC    15,GETR            SPECIAL ENTRY F(REG)
 C0C508 900F D338            00338  481 GETV     STM   R0,R15,GETSAVE
 C0050C 45E0 2566            00566  482          BAL   R14,FCHECK         COMPUTE L AND R
 000510 45E0 2598            00598  483          BAL   R14,MEMORY         UNPACK TO MREF
 C0C514 1844                        484 GETR     SR    R4,R4
 00C516 504C D4E8            004E8  485          ST    R4,VREF
 00C51A 404C D4EC            004EC  486          STH   R4,VREF+4          ZERO
 00C51E 1855                        487          SR    R5,R5
 00C520 4340 D27B            0027B  488          IC    R4,LEFT
 000524 4350 D27C            0027C  489          IC    R5,RIGHT
 000528 1833                        490          SR    R3,R3              ZERO
 00C52A 1943                        491          CR    R4,R3              SIGN IN FIELD?
 00052C 4770 2544            00544  492          BC    7,NOSIGN
 0CC530 4330 D4E0            004E0  493          IC    R3,MSIGN
 000534 423C D4E8            004E8  494          STC   R3,VSIGN
 00C538 1833                        495          SR    R3,R3
 00C53A 1953                        496          CR    R5,R3
 00053C 4780 255A            0055A  497          BE    SIGNONLY
 00C540 4140 4001            00001  498          LA    R4,1(0,R4)         L=1
 000544 4184 D4E0            004E0  499 NOSIGN    LA    R8,MREF(R4)        FIRST BYTE OF MREF MOVE
 000548 1854                        500          SR    R5,R4              R-L
 00054A 1875                        501          LR    R7,R5              LENGTH -1
 00054C 4130 0004            00004  502          LA    R3,4
 00C55C 1835                        503          SR    R3,R5              5-LENGTH
 000552 4163 D4E9            004E9  504          LA    R6,VREF+1(R3)      RECEIVE FIELD START
 000556 4470 2560            00560  505          EX    R7,MOVMREF
 00C55A 980F D338            00338  506 SIGNONLY LM    R0,R15,GETSAVE
 00C55E 07FE                        507          BR    R14                RETURN
 00C560 D200 6000 8000 00000 00000  508 MOVMREF  MVC   0(0,R6),0(R8)
```

```
 LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                     510 *
                                     511 *        ***                                                   ***
                                     512 *        ***        FCHECK,MEMORY                              ***
                                     513 *        ***                                                   ***
                                     514 *        *******************************************************
                                     515 *        FCHECK     PROCESSES A PARTIAL FIELD SPECIFICATION,MAKING
                                     516 *                   SURE IT HAS THE FORM BL * R WITH L LE R LE 5
000C5E6  90CF D388            000388 517 FCHECK  STM    R0,R15,FSAVE
000C5EA  1B44                        518        SR     R4,R4
000C5EC  1B55                        519        SR     R5,R5                  ZERO
000C5EE  434C 04DE            0004DE 520        IC     R4,BYTEFCUR
000C5F2  8C4C 0003            000003 521        SRDL   R4,3                   R4 HAS LEFT
000C5F6  8B5C 001D            00001D 522        SRL    R5,29                  R5 HAS RIGHT
000C5FA  413C 0005            000005 523        LA     R3,5
000C5FE  1953                        524        CR     R5,R3
000C600  472C CC8A            000108A 525        BC     2,FERROR               R > 5?
000C604  1945                        526        CR     R4,R5
000C606  472C CC8A            000108A 527        BC     2,FERROR               L>R?
000C60A  425C 027C            00027C 528        STC    R5,RIGHT
000C60E  424C 027B            00027B 529        STC    R4,LEFT
000C612  98CF D3B8            0003B8 530        LM     R0,R15,FSAVE
000C616  07FE                        531        BR     R14                    RETURN
                                     532 *
                                     533 *        MEMORY     THIS SR WILL FETCH A MIXWORD WITH A VALID MEMORY
                                     534 *                   ADDRESS, UNPACK THE WORD TO MREF. THE MIX LOCATION
                                     535 *                   IS SAVED IN MEMREF
000C61A  90CF D378            000378 536 MEMORY  STM    R0,R15,PSAVE
000C61E  5859 D408            00D408 537        L      R9,PIXLOC
000C622  1299                        538        LTR    R9,R9
000C624  4740 C682            01C682 539        BP     MEMERR
000C628  413C CF9F            00F9F 540        LA     R3,3999
000C62C  1993                        541        CR     R9,R3
000C62E  472C CC82            01C682 542        BH     MEMERR
000C632  8959 C002            0C0002 543        SLL    R9,2
000C636  58F9 D4F0            00D4F0 544        L      R15,PIXCORE(R9)        THE WORD FOR UNPACK
000C63A  4110 D4E0            00D4E0 545        LA     R1,MREF
000C63E  45EC 25FC            005FC 546        BAL    R14,UNPACK
000C642  98CF D378            000378 547        LM     R0,R15,PSAVE
000C646  07FE                        548        BR     R14                    RETURN
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     550 *        ********************************************************************
                                     551 *        ***                                                              ***
                                     552 *        ***        TWCCOMP,RECOMP                                        ***
                                     553 *        ***                                                              ***
                                     554 *        ********************************************************************
                                     555 *        TWOCOMP   CONVERT SIGN AND MAGNITUDE TO TWO'S COMPLEMENT
00005C6 12FF                         556 TWCCOMP  LTR   R15,R15
00005C8 C7AE                         557          BCR   10,R14              RETURN IF 0 OR +
00005CA 90CE D3F8           003F8    558          STM   R0,R14,TWCSAVE
00005CE 88F0 0001           00001    559          SLL   R15,1               DUMP SIGN BIT
00005D2 88F0 C001           00001    560          SRL   R15,1               RESTORE MAGNITUDE
00005D6 13FF                         561          LCR   R15,R15             COMPLEMENT MAGNITUDE
00005D8 980E D3F8           003F8    562          LM    R0,R14,TWCSAVE
00005DC C7FE                         563          BR    R14
                                     564 *
                                     565 *        RECOMP    CONVERT TWO'S COMPLEMENT TO SIGN AND MAGNITUDE.
00005DE 12FF                         566 RECOMP   LTR   R15,R15
00005EC C7AE                         567          BCR   10,R14              RETURN IF 0 OR +
00005E2 900E D3F8           003F8    568          STM   R0,R14,TWCSAVE
00005E6 135F                         569          LCR   R5,R15              RECOMPLEMENT
00005E8 8950 0001           00001    570          SLL   R5,1                ROOM FOR SIGN
00005EC 4140 0001           00001    571          LA    R4,1                SIGN
00005F0 8C40 0C01           00001    572          SRDL  R4,1                ATTACH SIGN
00005F4 18F5                         573          LR    R15,R5
00005F6 980E D3F8           003F8    574          LM    R0,R14,TWCSAVE
00005FA C7FE                         575          BR    R14
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                     577 *        ********************************************************************
                                     578 *        ***                                                              ***
                                     579 *        ***       UNPACK,PACK                                            ***
                                     580 *        ***                                                              ***
                                     581 *        ********************************************************************
                                     582 *        UNPACK    THIS SBR TAKES THE MIXWORD IN 32 BIT FORMAT
                                     583 *                  AND CONVERTS IT TO A 6 BYTE FORMAT OF SIGN AND
                                     584 *                  5 BYTES.
                                     585 *             ENTRY: R15 HAS THE 32 BIT WORD TO BE CONVERTED.  R1 HAS
                                     586 *             THE ADDRESS AT WHICH THE UNPACKED VERSION IS TO BE
                                     587 *             STORED.
000C5FC 900F D2F8           002F8    588 UNPACK    STM   R0,R15,UNPSAVE
000C6CC 1B44                         589           SR    R4,R4              ZERO
000C6C2 185F                         590           LR    R5,R15
000604 8D40 0001            00001    591           SLDL  R4,1               R4 HAS SIGN
000C608 424C 1000           00000    592           STC   R4,0(0,R1)         STORE SIGN
000C6CC 8950 0001           00001    593           SLL   R5,1
000610 1B44                          594           SR    R4,R4
000612 8D40 0006            00006    595           SLDL  R4,6               R4 HAS BYTE ONE
000616 424C 1001            00001    596           STC   R4,1(0,R1)
00061A 1B44                          597           SR    R4,R4
00061C 8D40 0006            00006    598           SLDL  R4,6               R4 HAS BYTE TWO
000620 4240 1002            00002    599           STC   R4,2(0,R1)
000624 1B44                          600           SR    R4,R4
000626 8D40 0006            00006    601           SLDL  R4,6               R4 HAS BYTE THREE
00062A 424C 1003            00003    602           STC   R4,3(0,R1)
00062E 1B44                          603           SR    R4,R4
000630 8D40 0006            00006    604           SLDL  R4,6               R4 HAS BYTE FOUR
000634 4240 1004            00004    605           STC   R4,4(0,R1)
000638 1B44                          606           SR    R4,R4
00063A 8D40 0006            00006    607           SLDL  R4,6               R4 HAS BYTE FIVE
00063E 4240 1005            00005    608           STC   R4,5(0,R1)
000642 980F D2F8            002F8    609           LM    R0,R15,UNPSAVE
000646 07FE                          610           BR    R14                RETURN
                                     611 *
                                     612 *        PACK  THIS SR TAKES THE MIXWORD IN 6 BYTE FORMAT AND CONVERTS
                                     613 *        IT TO A 32 BIT MIXWORD FORMAT.
                                     614 *        ENTRY:  R1 HAS THE ADDRESS AT WHICH THE UNPACKED VERSION
                                     615 *                IS STORED.
                                     616 *        EXIT: R15 HAS THE PACKED VERSION
000648 900F D2B8            002B8    617 PACK      STM   R0,R15,PSAVE
00064C 1B44                          618           SR    R4,R4              ZERO
00064E 5850 1000            00000    619           L     R5,0(0,R1)         SIGN AND THREE BYTES
000652 8D40 0008            00008    620           SLDL  R4,8               SIGN IN R4
000656 895C 0001            00001    621           SLL   R5,1
00065A 8D40 0007            00007    622           SLDL  R4,7               BYTE ONE IN R4
00065E 895C 0002            00002    623           SLL   R5,2
000662 8D4C 0006            00006    624           SLDL  R4,6               BYTE TWO IN R4
000666 8950 0002            00002    625           SLL   R5,2
00066A 8D40 0006            00006    626           SLDL  R4,6               BYTE THRE
00066E 5850 1004            00004    627           L     R5,4(0,R1)
000672 8950 0002            00002    628           SLL   R5,2
000676 8D40 0006            00006    629           SLDL  R4,6               BYTE FOUR IN R4
00067A 895C 0002            00002    630           SLL   R5,2
00067E 8D40 C006            00006    631           SLDL  R4,6               R4 HAS THE WORD
000682 18F4                          632           LR    R15,R4
000684 980E D2B8            002B8    633           LM    R0,R14,PSAVE
000688 07FE                          634           BR    R14                RETURN
```

**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    636 *        ********************************************************************
                                    637 *        ***                                                              ***
                                    638 *        ***        ABADR,CVRFLC,ADDSUB                                   ***
                                    639 *        ***                                                              ***
                                    640 *        ********************************************************************
                                    641 *        ABADR TAKES ADDRESS IN UNPACKED FORMAT OF INST AND
                                    642 *              RETURNS IBM 360 ADDRESS IN IBMCORE
00068A 5040 D4D0             004D0  643 ABADR    ST     R4,AUXSAVE
00068E 5050 D4D4             004D4  644          ST     R5,AUXSAVE+4          SAVE REGISTERS
000692 5840 D4D8             004D8  645          L      R4,MIXLOC
000696 1244                         646          LTR    R4,R4
000698 4740 C082             01082  647          BM     MEMERR
00069C 4150 0F9F             00F9F  648          LA     R5,3999
0006A0 1945                         649          CR     R4,R5
0006A2 4720 C082             01082  650          BH     MEMERR
0006A6 894C 0002             00002  651          SLL    R4,2
0006AA 4144 D4F0             004F0  652          LA     R4,MIXCORE(R4)
0006AE 5040 D280             00280  653          ST     R4,IBMCORE
0006B2 584C D4D0             004D0  654          L      R4,AUXSAVE            RESTORE REGS
0006B6 5850 D4D4             004D4  655          L      R5,AUXSAVE+4
0006BA 07FE                         656          BR     R14                   RETURN
                                    657 *
0006BC 900F D478             00478  658 CVRFLC   STM    R0,R15,SAVEC
0006C0 4180 D4F0             004F0  659          LA     R11,MIXCORE
0006C4 184A                         660          LR     R4,R10
0006C6 184B                         661          SR     R4,R11
0006C8 8A40 0002             00002  662          SRA    R4,2                  GET LOC IN MIXCORE
0006CC 4180 0001             00001  663          LA     R11,1
0006D0 184B                         664          SR     R4,R11                INST AT LOC R4
0006D2 4E40 D4C8             004C8  665          CVD    R4,PACKAREA
                                    666          MVC    MSGLINE(80),=CL80' LOC      : ARITHMETIC OVERFLOW,EXECUTX
0006D6 D24F D1E5 C128 001E5 01128                       ION CONTINUES MODULO WORD SIZE'
0006DC F342 D1EA D4CD 001EA 004CD  667          UNPK   MSGLINE+5(5),PACKAREA+5(3)
0006E2 96F0 D1EE      001EE        668          OI     MSGLINE+9,X'F0'
                                    669          PUT    MACHLIST,MSGLINE
0006F4 980F D478             00478  674          LM     R0,R15,SAVEC
0006F8 07FE                         675          BR     R14
                                    676 *
                                    677 *        ADDSUB     AREG <---AREG+-V
0006FA 45E0 2508             00508  678 ADDSUB   BAL    R14,GETV              VREF HAS FIELD
0006FE 4110 D4E8             004E8  679          LA     R1,VREF
000702 45E0 2648             00648  680          BAL    R14,PACK
000706 45E0 25C6             005C6  681          BAL    R14,TWOCOMP
00070A 187F                         682          LR     R7,R15                TWO'S COMPLEMENT
00070C 58F0 D288             00288  683          L      R15,AREG
000710 45EC 25C6             005C6  684          BAL    R14,TWOCOMP
000714 186F                         685          LR     R6,R15                TWO'S COMPLEMENT
000716 1888                         686          SR     R8,R8
000718 4380 D4DF             004DF  687          IC     R8,BYTEFIVE           GET OP CODE
00071C 413C 0001             00001  688          LA     R3,1
000720 1983                         689          CR     R8,R3                 IS IT ADD?
000722 47E0 2F74             00F74  690          BE     INC
000726 47F0 2F70             00F70  691          B      DEC
```

```
 LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     693 *        ********************************************************************
                                     694 *        ***                                                              ***
                                     695 *        ***        MULTIPLY                                              ***
                                     696 *        ***                                                              ***
                                     697 *        ********************************************************************
                                     698 *        MUL    A,X <--- A*V; 10 BYTE PRODUCT SET TO SAME SIGN
00072A 45E0 2508              00508  699 MUL      BAL    R14,GETV              VREF HAS F(M)
00072E 4110 D4E8              004E8  700          LA     R1,VREF
000732 45E0 2648              00648  701          BAL    R14,PACK
000736 45E0 25C6              005C6  702          BAL    R14,TWOCOMP
00073A 187F                          703          LR     R7,R15                TWO'S COMPLEMENT
00073C 58F0 D288              00288  704          L      R15,AREG
000740 45E0 25C6              005C6  705          BAL    R14,TWOCOMP
000744 185F                          706          LR     R5,R15                TWO'S COMPLEMENT
000746 1C47                          707          MR     R4,R7                 PRODUCT IN R4-R5
000748 8F4C 0002              00002  708          SLDA   R4,2                  ALIGN X X REG
00074C 885C 0002              00002  709          SRL    R5,2
000750 8A4C 0002              00002  710          SRA    R4,2
000754 18F4                          711          LR     R15,R4
000756 45E0 25DE              005DE  712          BAL    R14,RECOMP
00075A 184F                          713          LR     R4,R15                SIGN AND MAGNITUDE
00075C 18F5                          714          LR     R15,R5
00075E 45E0 25DE              005DE  715          BAL    R14,RECOMP
000762 185F                          716          LR     R5,R15                SIGN AND MAGNITUDE
000764 5040 D288              00288  717          ST     R4,AREG
000768 5050 D2A4              002A4  718          ST     R5,XREG
00076C 47F0 2468              00468  719          B      CYCLE
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                   721 *        ********************************************************************
                                   722 *        ***                                                              ***
                                   723 *        ***        DIVIDE                                                ***
                                   724 *        ***                                                              ***
                                   725 *        ********************************************************************
                                   726 *        DIV    A,X/V--->QUCT:AREG,REM:XREG
000770 45EC 2508                   727 DIV      BAL    R14,GETV
000774 4110 D4E8                   728          LA     R1,VREF
000778 45E0 2648                   729          BAL    R14,PACK
00077C 189F                        730          LR     R9,R15              R9 HAS DIVISOR
00077E 5850 D288                   731          L      R5,AREG
000782 8D8C 0001                   732          SLDL   R8,1
000786 8D4C 0001                   733          SLDL   R4,1                SAVE A SIGN
00078A 1959                        734          CR     R5,R9
00078C 472C 27F8                   735          BH     TRASH               OVERFLO
000790 1864                        736          LR     R6,R4
000792 8C4C 0001                   737          SRDL   R4,1                RESTORE SIGNS
000796 8C8C 0001                   738          SRDL   R8,1
00079A 18F9                        739          LR     R15,R9
00079C 45E0 25C6                   740          BAL    R14,TWOCOMP         TWOS COMP V
0007A0 189F                        741          LR     R9,R15
0007A2 1833                        742          SR     R3,R3               ZERO
0007A4 1993                        743          CR     R9,R3               DIVISOR = 0?
0007A6 47E0 27F8                   744          BE     TRASH
0007AA 5870 D2A4                   745          L      R7,XREG
0007AE 8970 0001                   746          SLL    R7,1                CUMP XREG SIGN
0007B2 8C6C 0001                   747          SRDL   R6,1                ATTACH A SIGN
0007B6 18F5                        748          LR     R15,R5              TWOS COMP AREG
0007B8 45E0 25C6                   749          BAL    R14,TWOCOMP
0007BC 184F                        750          LR     R4,R15              R4 HAS AREG
0007BE 18F7                        751          LR     R15,R7              TWOS COMP XREG
0007C0 45EC 25C6                   752          BAL    R14,TWOCOMP
0007C4 185F                        753          LR     R5,R15              R5 HAS XREG
0007C6 895C 0002                   754          SLL    R5,2
0007CA 8E40 0002                   755          SRDA   R4,2
0007CE 1D49                        756          DR     R4,R9               R4: REM; R5,QUOT
0007D0 18F5                        757          LR     R15,R5
0007D2 45E0 25DE                   758          BAL    R14,RECCMP
0007D6 185F                        759          LR     R5,R15              SIGN AND MAGNITUDE
0007D8 5050 D288                   760          ST     R5,AREG
0007DC 8D60 0001                   761          SLDL   R6,1                PREVIOUS A SIGN
0007E0 18F4                        762          LR     R15,R4
0007E2 45EC 25DE                   763          BAL    R14,RECCMP
0007E6 187F                        764          LR     R7,R15              ATTACH TO XREG
0007E8 897C 0001                   765          SLL    R7,1
0007EC 8C6C 0001                   766          SRDL   R6,1
0007F0 5070 D2A4                   767          ST     R7,XREG
0007F4 47F0 2468                   768          B      CYCLE
0007F8 5C5C D288                   769 TRASH    ST     R5,AREG             AREG UNDEFINED
0007FC 5C90 D2A4                   770          ST     R9,XREG
000800 45EC 26BC                   771          BAL    R14,OVRFLC
000804 9201 D279          000279   772          MVI    OVTOG,X'01'         TOGGLE ON
000808 47F0 2468                   773          B      CYCLE
```

```
  LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                      775 *        ********************************************************************
                                      776 *        ***                                                              ***
                                      777 *        ***       HALT,CHAR                                              ***
                                      778 *        ***                                                              ***
                                      779 *        ********************************************************************
                                      780 *        SPEC  NUM: F=0
                                      781 *              CHAR:F=1
                                      782 *              HALT: F=2
000080C 1844                          783 SPEC     SR    R4,R4
000080E 434C D4DE           004DE     784          IC    R4,BYTEFOUR          GET F
000812 415C 0002            00002     785          LA    R5,2
000816 1945                           786          CR    R4,R5
000818 4720 CC8A            0108A     787          BC    2,ERROR
00C81C 894C 0002            00002     788          SLL   R4,2
000820 47F4 2824            00824     789          B     *+4(R4)              ALIGN BRANCH ROUTINES
000824 47FC 28A8            008A8     790          B     NUM
000828 47F0 2830            00830     791          B     CHAR
00082C 47F0 22EA            002EA     792          B     ENDJOB
                                      793 *
                                      794 *        CHAR  CONVERTS NUMERIC VALUE IN AREG TO 10 BYTE CHARACTER
                                      795 *              CODE IN A.XREG SUITABLE FOR OUTPUT.
00C830 5850 D288            00288     796 CHAR     L     R5,AREG
000834 8D40 0001            00001     797          SLDL  R4,1                 SAVESIGN
0C0838 8850 C001            00001     798          SRL   R5,1                 RESTORE MAGNITUDE
00083C 4E50 D4C8            004C8     799          CVD   R5,PACKAREA
0C0840 F397 D111 D4C8 00111 004C8     800          UNPK  INCARD+1(10),PACKAREA
000846 4130 001E            0001E     801          LA    R3,30
00084A 4150 D111            00111     802          LA    R5,INCARD+1
00084E 418C 000A            0000A     803          LA    R8,10
000852 1866                           804 DIGIT    SR    R6,R6                ZERO
000854 940F 5000      C0000           805          NI    0(R5),X'0F'          ZERO ZONE
000858 436C 5000            00000     806          IC    R6,0(0,R5)           GET DIGIT
00085C 1A63                           807          AR    R6,R3                ADD 30
00085E 426C 5000            00000     808          STC   R6,0(0,R5)           RETURN IT
000862 4150 5001            00001     809          LA    R5,1(0,R5)           INC
000866 4680 2852            00852     810          BCT   R8,DIGIT
00086A 9200 D110      00110           811          MVI   INCARD,X'00'
00086E 4110 D110            00110     812          LA    R1,INCARD
00C872 45E0 2648            00648     813          BAL   R14,PACK
000876 185F                           814          LR    R5,R15
0CC878 895C C001            00001     815          SLL   R5,1
00087C 8C40 0001            00001     816          SRDL  R4,1                 RESTORE SIGN
000880 5050 D288            00288     817          ST    R5,AREG
000884 D204 D111 D116 00111 00116     818          MVC   INCARD+1(5),INCARD+6    POSITION FOR PACK
00088A 45E0 2648            00648     819          BAL   R14,PACK             R15 HAS PACKED XREG
00C88E 5850 D2A4            002A4     820          L     R5,XREG
000892 8D40 C001            00001     821          SLDL  R4,1                 SAVE SIGN
00C896 185F                           822          LR    R5,R15
00C898 8950 0001            00001     823          SLL   R5,1
00C89C 8C40 C001            00001     824          SRDL  R4,1                 RESTORE SIGN
00C8AC 5050 D2A4            002A4     825          ST    R5,XREG
0008A4 47F0 2468            00468     826          B     CYCLE
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     828 *        ********************************************************************
                                     829 *        ***                                                              ***
                                     830 *        ***        NUM                                                   ***
                                     831 *        ***                                                              ***
                                     832 *        ********************************************************************
                                     833 *        NUM    CHANGE 10 BYTE CHAR CODE IN A X REGS TO NUMERIC
                                     834 *               IN A REG
00C8A8 4110 D4E0                     835 NUM      LA     R1,MREF             PARMS FOR UNPACK
00C8AC 58F0 D288                     836          L      R15,AREG
00C8B0 45E0 25FC                     837          BAL    R14,UNPACK          MREF HAS UNPL AREG
00C8B4 4110 D4E8                     838          LA     R1,VREF             PARMS FOR UNPACK
00C8B8 58F0 D2A4                     839          L      R15,XREG
00C8BC 45E0 25FC                     840          BAL    R14,UNPACK          BREF HAS UNPL XREG
00C8C0 417C 0005                     841          LA     R7,5                COUNTER
00C8C4 4140 D4E1                     842          LA     R4,MONE             INITIAL FOR AREG
00C8C8 4150 D110                     843          LA     R5,INCARD
00C8CC 1B33                          844          SR     R3,R3
00C8CE 4330 4000                     845 MBYTE    IC     R3,0(0,R4)          TAKE A AREG BYTE
00C8D2 4E30 D4C8                     846          CVD    R3,PACKAREA         DECIMAL
00C8D6 D200 5000 D4CF 00C00  004CF   847          MVC    0(1,R5),PACKAREA+7      SIGN & DIGIT ONLY
00C8DC 4140 4001                     848          LA     R4,1(0,R4)          INC AREG BYTE POINTER
00C8E0 4150 5001                     849          LA     R5,1(0,R5)          INC TEMP STOR OINTER
00C8E4 4670 28CE                     850          BCT    R7,MBYTE            5 BYTES DONE?
00C8E8 4140 D4E9                     851          LA     R4,VONE             INITIAL FOR XREG
00C8EC 4170 0005                     852          LA     R7,5
00C8F0 4330 4C00                     853 VBYTE    IC     R3,0(0,R4)          TAKE A XREG BYTE
00C8F4 4E30 D4C8                     854          CVD    R3,PACKAREA         DECIMAL
00C8F8 D2C0 5000 D4CF 00C00  004CF   855          MVC    0(1,R5),PACKAREA+7    SIGN & DIGIT ONLY
00C8FE 4140 4001                     856          LA     R4,1(0,R4)          INC XREG BYTE POINTER
00C902 4150 5001                     857          LA     R5,1(0,R5)          INC REMP STOR POINTER
00C906 4670 28F0                     858          BCT    R7,VBYTE            5 BYTES DONE
00C9CA 1B44                          859          SR     R4,R4               ZERO
00C90C 4840 D118                     860          LH     R4,INCARD+8 LAST 2  DIGITS
00C910 8C4C CCC8                     861          SRDL   R4,8                SAVE SIGN & DIGIT
00C914 884C CC04                     862          SRL    R4,4                DUMP SIGN ZONE
00C918 8C4C 0004                     863          SRDL   R4,4                SAVE D2
00C91C 584C D114                     864          L      R4,INCARD+4
00C920 884C CC04                     865          SRL    R4,4                DUMP SING
00C924 8C40 CCC4                     866          SRDL   R4,4                SAVE D3
00C928 884C CC04                     867          SRL    R4,4                DUMP SIGN
00C92C 8C40 C004                     868          SRDL   R4,4                SAVE D4
00C930 884C C004                     869          SRL    R4,4                DUMP SIGN
00C934 8C4C CC04                     870          SRDL   R4,4                SAVE D5
00C938 884C C004                     871          SRL    R4,4                SUMP SIGN
00C93C 8C4C C004                     872          SRDL   R4,4                SAVE D6
00C940 5840 D110                     873          L      R4,INCARD
00C944 884C 0004                     874          SRL    R4,4                DUMP SIGN
00C948 8C40 0004                     875          SRDL   R4,4                SAVE D7
00C94C 5050 D4CC                     876          ST     R5,PACKAREA+4       7 DIGITS & SIGN
00C950 884C 0004                     877          SRL    R4,4                DUMP SIGN
00C954 8C40 C004                     878          SRDL   R4,4                SAVE D8
00C958 884C C004                     879          SRL    R4,4                DUMP SIGN
00C95C 8C40 C004                     880          SRDL   R4,4                SAVE D9
00C960 884C C004                     881          SRL    R4,4                DUMP SIGN
00C964 8C4C C004                     882          SRDL   R4,4                SAVE D10
00C968 8850 0014                     883          SRL    R5,20               ALIGN
00C96C 5050 D4C8                     884          ST     R5,PACKAREA
00C970 F975 D4C8 C1EE C04C8  011EE   885          CP     PACKAREA(8),MAXNC
00C976 4720 2994                     886          8H     MODULO
00C97A 4F50 D4C8                     887 TOBIN    CVB    R5,PACKAREA
```

```
  LOC   OBJECT CODE     ADDR1 ADDR2   STMT   SOURCE STATEMENT

00097E 1844                          888           SR    R4,R4
000980 4340 D4E0                     889           IC    R4,PSIGN            GET SIGN
000984 8950 0001                     890           SLL   R5,1
000988 8C40 C001                     891           SRDL  R4,1                ATTACH SIGN
00098C 5050 D288                     892           ST    R5,AREG
000990 47F0 2468                     893           B     CYCLE
000994 F8D7 D26A D4C8 0026A 004C8    894 MODULO    ZAP   QUOREM(14),PACKAREA
00099A FCD5 D26A C1EE 0026A 011EE    895           DP    QUOREM(14),MAXNO
0009A0 45E0 26BC                     896           BAL   R14,OVRFLO
0009A4 F875 D4C8 D272 C04C8 00272    897           ZAP   PACKAREA(8),QUOREM+8(6)
0009AA 9201 D279      00279          898           MVI   OVTOG,X'01'
0009AE 47F0 297A                     899           B     TOBIN
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     901 *        ****************************************************************
                                     902 *        ***                                                          ***
                                     903 *        ***        SHIFTS                                            ***
                                     904 *        ***                                                          ***
                                     905 *        ****************************************************************
                                     906 *        SHIFT F INDICATES TYPE SHIFT. BOTH A,X PARTICIPATE IN
                                     907 *        CIRCULATING SHIFTS. REG SIGNS ARE NOT AFFECTED.
0009B2 1B88                          908 SHIFT    SR     R8,R8
0009B4 4380 D4DE            004DE    909          IC     R8,BYTEFCUR          GET F
0009B8 4130 0005            00005    910          LA     R3,5
0009BC 19B3                          911          CR     R8,R3                F > 5?
0009BE 4720 C0BA            0108A    912          BH     FERRCR
0009C2 5850 D4DB            004D8    913          L      R5,MIXLCC            GET AA
0009C6 1B44                          914          SR     R4,R4
0009C8 1954                          915            CR     R5,R4                0 > M?
0009CA 47D0 C0B2            010B2    916          BNH    MEMERR
0009CE 58F0 D2B8            002B8    917          L      R15,AREG             FETCH AREG
0009D2 411C D4E0            004E0    918          LA     R1,MREF
0009D6 45E0 25FC            005FC    919          BAL    R14,UNPACK           AREG IN MREF
0009DA 58F0 D2A4            002A4    920          L      R15,XREG             FETCH XREG
0009DE 411C D4E8            004E8    921          LA     R1,VREF
0009E2 45E0 25FC            005FC    922          BAL    R14,UNPACK           XREG IN VREF
0009E6 9200 D11A      0011A          923          MVI    INCARD+10,X'00'      ZERO WORK AREA
0009EA D21C D11B D11A 0011B 0011A    924          MVC    INCARD+11(29),INCARD+10
0009F0 D204 D124 D4E1 00124 004E1    925          MVC    INCARD+20(5),MONE    CONCATENATE A,X
0009F6 D204 D129 D4E9 00129 004E9    926          MVC    INCARD+25(5),VONE
0009FC 8988 C002            00002    927          SLL    R8,2
000A00 47F8 2A04            00A04    928          B      *+4(R8)
000A04 47FC 2A1C            00A1C    929          B      SLA
000A08 47F0 2A46            00A46    930          B      SRA
000A0C 47F0 2A72            00A72    931          B      SLAX
000A10 47F0 2AA8            00AA8    932          B      SRAX
000A14 47F0 2AC8            00AC8    933          B      SLC
000A18 47FC 2AEE            00AEE    934          B      SRC
                                     935 *
000A1C 4130 0005            00005    936 SLA      LA     R3,5
000A20 1953                          937          CR     R5,R3                SHIFT GT 5: AREG IS ZERO
000A22 47B0 2B28            00B2B    938          BNL    AREGZERC
000A26 D204 D129 D11A 0C129 0011A    939          MVC    INCARD+25(5),INCARD+10    ZERO XREG
000A2C 4165 D124            00124    940          LA     R6,INCARD+20(R5)
000A30 D204 D4E1 6CCC 004E1 00000    941          MVC    MONE(5),0(R6)        MOVE SHIFTED AREG FOR PACK
000A36 4110 D4E0            004E0    942          LA     R1,MREF
000A3A 45EC 2648            00648    943          BAL    R14,PACK
000A3E 50F0 D2B8            002B8    944          ST     R15,AREG
000A42 47FC 2468            00468    945          B      CYCLE
                                     946 *
000A46 413C 0005            00005    947 SRA      LA     R3,5
000A4A 1953                          948          CR     R5,R3                SHIFT GT 5: AREG IS ZERO
000A4C 47BC 2B28            00B28    949          BNL    AREGZERC
000A50 D204 D129 D11A 0C129 0011A    950          MVC    INCARD+25(5),INCARD+10
000A56 1B35                          951          SR     R3,R5                POINTER
000A58 4163 D11F            0011F    952          LA     R6,INCARD+15(R3)
000A5C D204 D4E1 6CCC 004E1 00000    953          MVC    MONE(5),0(R6)        MOVE SHIFTED REG FOR PACK
000A62 4110 D4E0            004E0    954          LA     R1,MREF
000A66 45EC 2648            00648    955          BAL    R14,PACK
000A6A 50F0 D2B8            002B8    956          ST     R15,AREG
000A6E 47F0 2468            00468    957          B      CYCLE
                                     958 *
000A72 413C 000A            0000A    959 SLAX     LA     R3,10
000A76 1953                          960          CR     R5,R3
```

```
  LCC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

000A78 4780 2B16             00B16   961            BNL    REGSZERO
000A7C 4165 D124             00124   962            LA     R6,INCARD+20(R5)
000A80 D204 D4E1 6000 004E1  00000   963            MVC    MONE(5),0(R6)       MOVE AREG FOR PACK
000A86 D204 D4E9 60C5 004E9  00005   964            MVC    VONE(5),5(R6)       MOVE XREG FOR PACK
000A8C 4110 D4E0             004E0   965 PKSTORE    LA     R1,PREF
000A90 45E0 2648             00648   966            BAL    R14,PACK
000A94 50F0 D288             00288   967            ST     R15,AREG
000A98 4110 D4E8             004E8   968            LA     R1,VREF
000A9C 45E0 2648             00648   969            BAL    R14,PACK
000AA0 50F0 D2A4             002A4   970            ST     R15,XREG
000AA4 47F0 2468             00468   971            B      CYCLE
                                     972 *
000AA8 4130 000A             0000A   973 SRAX       LA     R3,10
000AAC 1953                  00974   974            CR     R5,R3
000AAE 4780 2B16             00B16   975            BNL    REGSZERO
000AB2 1B35                  00976   976            SR     R3,R5
000AB4 4165 D11A             0011A   977            LA     R6,INCARD+10(R5)    SHIFT ADDRESS
000AB8 D204 D4E1 6000 004E1  00000   978            MVC    MONE(5),0(R6)       MOVE AREG FOR PACK
000ABE D204 D4E9 60C5 004E9  00005   979            MVC    VONE(5),5(R6)
000AC4 47F0 2A8C             00A8C   980            B      PKSTORE
                                     981 *
000AC8 4130 000A             0000A   982 SLC        LA     R3,10
000ACC 1B53                  00983   983            SR     R5,R3               REPEATED SUB TO GET MODULO REM
000ACE 47A0 2ACC             00ACC   984            BC     10,*-2
000AD2 1A53                  00985   985            AR     R5,R3               RESTORE TO PLUS
000AD4 D209 D12E D124 0012E  00124   986            MVC    INCARD+30(10),INCARD+20   EXTEND ON RIGHT
000ADA 4165 D124             00124   987            LA     R6,INCARD+20(R5)
000ADE D204 D4E1 60CC 004E1  00000   988            MVC    MONE(5),0(R6)
000AE4 D204 D4E9 60C5 004E9  00005   989            MVC    VONE(5),5(R6)
000AEA 47F0 2A8C             00A8C   990            B      PKSTORE
                                     991 *
000AEE 4130 000A             0000A   992 SRC        LA     R3,10
000AF2 1B53                  00993   993            SR     R5,R3               REPEATED SUB TO GET MODULO REM
000AF4 47A0 2AF2             00AF2   994            BC     10,*-2
000AF8 1A53                  00995   995            AR     R5,R3
000AFA D209 D11A D124 C011A  00124   996            MVC    INCARD+10(10),INCARD+20       EXTEND ON LEFT
000B00 1B35                  00997   997            SR     R3,R5
000BC2 4163 D11A             0011A   998            LA     R6,INCARD+10(R3)
000B06 D204 D4E1 6000 004E1  00000   999            MVC    MONE(5),0(R6)
000B0C D204 D4E9 60C5 004E9  00005  1000            MVC    VONE(5),5(R6)
000B12 47F0 2A8C             00A8C  1001            B      PKSTORE
                                    1002 *
000B16 5870 D2A4             002A4  1003 REGSZERO   L      R7,XREG             FETCH FOR SIGN
000B1A 8D60 0001             00001  1004            SLDL   R6,1                SAVE SIGN
000B1E 1B77                 01005   1005            SR     R7,R7               ZERO
000B20 8C60 0001             00001  1006            SRDL   R6,1
000B24 5070 D2A4             002A4  1007            ST     R7,XREG
000B28 5870 D288             00288  1008 AREGZERO  L      R7,AREG
000B2C 8D60 0001             00001  1009            SLDL   R6,1                SAVE SIGN
000B30 1B77                 01010   1010            SR     R7,R7
000B32 8C60 C001             00001  1011            SRDL   R6,1                RESTORE SIGN
000B36 5070 D288             00288  1012            ST     R7,AREG
000B3A 47F0 2468             00468  1013            B      CYCLE
```

**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                   1015 *        ********************************************************************
                                   1016 *        ***                                                              ***
                                   1017 *        ***        MOVE                                                  ***
                                   1018 *        ***                                                              ***
                                   1019 *        ********************************************************************
                                   1020 *        MOVE  MOVE F WORDS FROM M TO ADDRESS IN IREG1.
000B3E 1B44                        1021 MOVE     SR    R4,R4
000B40 4340 D4DE             004DE 1022         IC    R4,BYTEFOUR          GET F
000B44 1244                        1023         LTR   R4,R4
000B46 4780 246B             00468 1024         BZ    CYCLE                F=0--->NO MOVE
000B4A 45EC 268A             0068A 1025         BAL   R14,ABADR
000B4E 5850 D2B0             002B0 1026         L     R5,IBMCORE           360 FROM ADDRESS
000B52 5860 D2BC             0028C 1027         L     R6,IREG1             IREG1 HAS TO LOCATION
000B56 5060 D4D8             004D8 1028         ST    R6,MIXLOC
000B5A 45E0 26BA             0068A 1029         BAL   R14,ABADR            CONVERT TO 360 ADDRESS
000B5E 5870 D2B0             00280 1030         L     R7,IBMCORE           360 TO ADDRESS
000B62 1A64                        1031         AR    R6,R4                ADD F TO IREG1
000B64 5060 D28C             0028C 1032         ST    R6,IREG1
000B68 D203 7000 5CC0 00C00  00000 1033 MOVEF   MVC   0(4,R7),0(R5)        MOVE 4 BYTES = 1 MIX WORD
000B6E 4150 5004             00004 1034         LA    R5,4(0,R5)           INC FROM ADDRESS
000B72 4170 7C04             00004 1035         LA    R7,4(0,R7)           INC TO ADDRESS
000B76 4640 2B68             00B6B 1036         BCT   R4,MOVEF
000B7A 47F0 2468             0046B 1037         B     CYCLE
```

**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    1039 *        ****************************************************************
                                    1040 *        ***                                                          ***
                                    1041 *        ***       LOADS                                              ***
                                    1042 *        ***                                                          ***
                                    1043 *        ****************************************************************
                                    1044 *        LDA   AREG <--- F(M)
000B7E 45E0 250B              00508 1045 LDA      BAL   R14,GETV            PROCESS F(M)
000B82 4110 D4EB              004E8 1046          LA    R1,VREF
000B86 45E0 264B              00648 1047          BAL   R14,PACK
000B8A 50F0 D28B              0028B 1048          ST    R15,AREG
000B8E 47FC 2468              00468 1049          BC    15,CYCLE
                                    1050 *        LDI   IREG <--- F(M)
000B92 45E0 2508              00508 1051 LDI      BAL   R14,GETV            PROCESS F(M)
000B96 4110 D4E8              004E8 1052          LA    R1,VREF
000B9A 45E0 2648              00648 1053          BAL   R14,PACK
000B9E 185F                         1054          LR    R5,R15              FIELD IS PACKED
000BA0 8D4C 0C01              00001 1055          SLDL  R4,1                SAVE SIGN
000BA4 885C 0001              00001 1056          SRL   R5,1
000BA8 4130 CFFF              00FFF 1057          LA    R3,4095
000BAC 1953                         1058          CR    R5,R3
000BAE 4720 CC92              01092 1059          BC    2,SIZERROR          2 BYTES ALLOWED
000BB2 885C C001              00001 1060          SLL   R5,1
000BB6 8C4C 0001              00C01 1061          SRDL  R4,1                RESTORE SIGN
000BBA 1833                         1062          SR    R3,R3               ZERO
000BBC 4330 D4DF              004DF 1063          IC    R3,BYTEFIVE         OP=8+INDEX NO
000BC0 4160 0008              00008 1064          LA    R6,8
000BC4 1836                         1065          SR    R3,R6               INDEX NO
000BC6 8930 0002              00002 1066          SLL   R3,2                TIMES FOUR
000BCA 5053 D28B              0028B 1067          ST    R5,AREG(R3)         STORE IN IREG(I)
000BCE 47F0 2468              00468 1068          BC    15,CYCLE
                                    1069 *        LDX   XREG   <---  F(M)
000BD2 45E0 250B              00508 1070 LDX      BAL   R14,GETV            PROCESS F(M)
000BD6 4110 D4E8              004E8 1071          LA    R1,VREF
000BDA 45E0 264B              00648 1072          BAL   R14,PACK
000BDE 50F0 D2A4              002A4 1073          ST    R15,XREG
000BE2 47F0 2468              00468 1074          BC    15,CYCLE
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                   1076 *        ********************************************************************
                                   1077 *        ***                                                              ***
                                   1078 *        ***       LOAD NEGATIVES                                         ***
                                   1079 *        ***                                                              ***
                                   1080 *        ********************************************************************
                                   1081 *        LDAN  AREG <---    -F(M)
000BE6 45E0 2508           00508   1082 LDAN     BAL   R14,GETV             PROCESS F(M)
000BEA 9701 D4E8     004E8         1083          XI    VSIGN,X'01'          REVERSE SIGN
000BEE 4110 D4E8           004E8   1084          LA    R1,VREF
000BF2 45E0 2648           00648   1085          BAL   R14,PACK
000BF6 50F0 D288           00288   1086          ST    R15,AREG
000BFA 47F0 2468           00468   1087          BC    15,CYCLE
                                   1088 *        LDIN  IREG <---   -F(M)
000BFE 45E0 2508           00508   1089 LDIN     BAL   R14,GETV             PROCESS F(M)
000C02 9701 D4E8     0C4E8         1090          XI    VSIGN,X'01'          REVERSE SIGN
000C06 4110 D4E8           004E8   1091          LA    R1,VREF
000C0A 45E0 2648           00648   1092          BAL   R14,PACK
000C0E 185F                        1093          LR    R5,R15               FIELD IS PACKED
000C10 8D40 0001           00001   1094          SLDL  R4,1                 SAVE SIGN
000C14 885C 0001           00001   1095          SRL   R5,1
000C18 4130 0FFF           00FFF   1096          LA    R3,4095
000C1C 1953                        1097          CR    R5,R3
000C1E 472C C092           01092   1098          BC    2,SIZERROR
000C22 895C 0001           00001   1099          SLL   R5,1
000C26 8C40 0001           00001   1100          SRDL  R4,1                 RESTORE SIGN
000C2A 1B33                        1101          SR    R3,R3                ZERO
000C2C 433C D4DF           004DF   1102          IC    R3,BYTEFIVE          OP-16 IS INDEX
000C30 4160 C010           00010   1103          LA    R6,16
000C34 1B36                        1104          SR    R3,R6
000C36 893C 0002           00002   1105          SLL   R3,2
000C3A 5053 D288           00288   1106          ST    R5,AREG(R3)          STORE IN IREG
000C3E 47FC 2468           00468   1107          BC    15,CYCLE
                                   1108 *        LDXN      XREG   <---    -F(M)
000C42 45EC 2508           00508   1109 LDXN     BAL   R14,GETV             PROCESS F(M)
000C46 9701 D4E8     0C4E8         1110          XI    VSIGN,X'01'
000C4A 411C D4E8           004E8   1111          LA    R1,VREF
000C4E 45EC 2648           00648   1112          BAL   R14,PACK
000C52 5CF0 D2A4           002A4   1113          ST    R15,XREG
000C56 47F0 2468           00468   1114          BC    15,CYCLE
```

**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

```
   LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     1116 *        ********************************************************************
                                     1117 *        ***                                                              ***
                                     1118 *        ***        STORES                                                ***
                                     1119 *        ***                                                              ***
                                     1120 *        ********************************************************************
                                     1121 *        STORE F(M) <--- AREG,XREG,IREG,ETC
000C5A 45E0 2566                00566 1122 STORE  BAL   R14,FCHECK
000C5E 45E0 2598                00598 1123        BAL   R14,MEMORY
000C62 1B44                           1124        SR    R4,R4               ZERO
000C64 4340 D4DF                004DF 1125        IC    R4,BYTEFIVE         OPCODE
000C68 4150 0018                00018 1126        LA    R5,24
000C6C 1B45                           1127        SR    R4,R5               CP CODE-24
000C6E 8940 0002                00002 1128        SLL   R4,2                TIMES TWO
000C72 58F4 D288                00288 1129        L     R15,AREG(R4)        FETCH DATA FROM DESIRED REG
000C76 4110 D4E8                004E8 1130        LA    R1,VREF
000C7A 45E0 25FC                005FC 1131        BAL   R14,UNPACK
000C7E 1B44                           1132        SR    R4,R4
000C80 1B55                           1133        SR    R5,R5
000C82 4340 D27B                0027B 1134        IC    R4,LEFT
000C86 4350 D27C                0027C 1135        IC    R5,RIGHT
000C8A 1B33                           1136        SR    R3,R3
000C8C 1943                           1137        CR    R4,R3               SIGN IN FIELD
000C8E 4770 2CA6                00CA6 1138        BC    7,ANCSIGN
000C92 4330 D4E8                004E8 1139        IC    R3,VSIGN
000C96 4230 D4E0                004E0 1140        STC   R3,MSIGN
000C9A 1B33                           1141        SR    R3,R3
000C9C 1953                           1142        CR    R5,R3               R = 0?
000C9E 4780 2CBA                00CBA 1143        BE    SIGALONE
000CA2 4140 4001                00001 1144        LA    R4,1(0,R4)          L=1
000CA6 4164 D4E0                004E0 1145 ANCSIGN LA   R6,MREF(R4)        FIRST BYTE TO GET AREG
000CAA 4180 0005                00005 1146        LA    R8,5
000CAE 1B54                           1147        SR    R5,R4               R-L
000CB0 1B85                           1148        SR    R8,R5               FIRST BYTE OF AREG
000CB2 4188 D4E8                004E8 1149        LA    R8,VREF(R8)
000CB6 4450 2CD2                00CD2 1150        EX    R5,AMOVE
000CBA 4110 D4E0                004E0 1151 SIGALONE LA  R1,MREF
000CBE 45E0 2648                00648 1152        BAL   R14,PACK
000CC2 5830 D4D8                004D8 1153        L     R3,MIXLCC
000CC6 8930 0002                00002 1154        SLL   R3,2
000CCA 50F3 D4F0                004F0 1155        ST    R15,MIXCORE(R3)
000CCE 47F0 2468                00468 1156        BC    15,CYCLE
000CD2 D200 6000 8000 00C00 00000 1157 AMOVE    MVC   0(0,R6),0(R8)
```

```
   LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     1159 *         ********************************************************************
                                     1160 *         ***                                                              ***
                                     1161 *         ***       ICC,IN                                                 ***
                                     1162 *         ***                                                              ***
                                     1163 *         ********************************************************************
                                     1164 *         IOC   CONTROL DEPENDS ON DEVICE USED
000CC8 1844                          1165 ICC       SR    R4,R4                    ZERO
000CCA 4150 0012           00012     1166           LA    R5,18
000CCE 4340 D4DE           004DE     1167           IC    R4,BYTEFCUR              GET F
000CE2 1945                          1168           CR    R4,R5
000CE4 4770 C09A           0109A     1169           BNE   ICERROR                  PRINTER ONLY
000CE8 5850 D4D8           004D8     1170           L     R5,MIXLOC
000CEC 895C 0001           00001     1171           SLL   R5,1                     DUMP SIGN
000CF0 1255                          1172           LTR   R5,R5
000CF2 4770 C09A           0109A     1173           BC    7,ICERROR                M=0 THIS INST
000CF6 9240 D160      CC16C          1174           MVI   MIXOUT,C' '
000CFA D283 D161 D16C 00161 00160    1175           MVC   MIXOUT+1(132),MIXOUT
000D0C 92F1 D160      CC160          1176           MVI   MIXOUT,C'1'              SKIP TO TOP OF PAGE
                                     1177           PUT   MACHLIST,MIXOUT
000D12 47F0 2468           00468     1182           B     CYCLE
                                     1183 *
                                     1184 *         IN    READS A CARD AND STORES THE 80 CHARACTERS IN MIX
                                     1185 *               CHARACTER CODE IN 16 MIXWORDS
                                     1186 *               STARTING AT LOCATION M.
000D16 1844                          1187 IN        SR    R4,R4                    ZERO
000D18 4150 0010           00010     1188           LA    R5,16
000D1C 4340 D4DE           004DE     1189           IC    R4,BYTEFCUR              GET F
000D2C 1945                          1190           CR    R4,R5
000D22 4770 C09A           0109A     1191           BNE   IOERROR                  ONLY INPUT DEVICE IS READER
                                     1192           GET   MIXINPUT,INCARD          READ THE CARD
000D34 92C0 D4E0      CC4E0          1197           MVI   MSIGN,X'00'              SET SIGN TO ZERO
000D38 DC4F D110 C501 00110 01501    1198           TR    INCARD(80),INCODE        TRANSLATE TO MIX CHAR CODE
000D3E 45EC 268A           0068A     1199           BAL   R14,ABADR                GET 360 ADDRESS FOR M
000D42 584C D280           00280     1200           L     R4,IBMCORE
000D46 415C 0010           C0010     1201           LA    R5,16                    WORD COUNTER
000D4A 416C D110           00110     1202           LA    R6,INCARD
000D4E D204 D4E1 6CCC 0C4E1 00000    1203 WORDMOVE  MVC   MONE(5),0(R6)
000D54 4110 D4E0           004E0     1204           LA    R1,MREF                  PARM FOR PACK
000D58 45EC 2648           00648     1205           BAL   R14,PACK                 R15 RETURNS PACKED WORD
000D5C 5CFC 4000           00000     1206           ST    R15,0(0,R4)              STORE IN MIX CORE
000D6C 414C 40C4           00004     1207           LA    R4,4(0,R4)               INC 360 ADDRESS
000D64 416C 6CC5           00005     1208           LA    R6,5(0,R6)               INC CARD COLUMN
000D68 465C 2D4E           00D4E     1209           BCT   R5,WORDMOVE              16 WORDS DONE
000D6C 47F0 2468           00468     1210           B     CYCLE
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    1212 *        ********************************************************************
                                    1213 *        ***                                                              ***
                                    1214 *        ***        CUT                                                   ***
                                    1215 *        ***                                                              ***
                                    1216 *        ********************************************************************
                                    1217 *        CUT PRINTS OUT CHARACTER EQUIVALENT OF MIX CODE IN 24
                                    1218 *            WORDS STARTING AT M.
000D70 1B44                         1219 CUT      SR    R4,R4                 ZERO
000D72 4150 C012          00012     1220          LA    R5,18
000D76 4340 D4DE          004DE     1221          IC    R4,BYTEFOUR           GET F
000D7A 1945               0109A     1222          CR    R4,R5
000D7C 4770 C09A          00280     1223          BNE   ICERROR
000D80 45E0 268A          00000     1224          BAL   R14,ABADR             GET 360 ADDRESS
000D84 5840 D280          00160     1225          L     R4,IBMCORE
000D88 9240 D160    CC160           1226          MVI   MIXOUT,C' '           CHARRIAGE CONTROL
000D8C D283 D161 D160 00161 00161     1227          MVC   MIXOUT+1(132),MIXOUT
000D92 4160 D161          00018     1228          LA    R6,MIXOUT+1           ADDRESS OUTPUT LINE
000D96 4150 0018          00000     1229          LA    R5,24                 WORD COUNTER
000D9A 58FC 4000          004E0     1230 GETWORD  L     R15,0(0,R4)           R15 HAS THE WORD FOR UNPACK
000D9E 4110 D4E0          005FC     1231          LA    R1,MREF               PARM FOR UNPACK
000DA2 45E0 25FC          01501     1232          BAL   R14,UNPACK
000DA6 DC04 D4E1 C501 004E1 004E1     1233          TR    MONE(5),INCODE
000DAC D204 6000 D4E1 0C0C0 00004     1234          MVC   0(5,R6),MONE          MOVE FIVE CHARACTERS
000DB2 4140 4004          00005     1235          LA    R4,4(0,R4)            INC 360 ADDRESS
000DB6 4160 6005          00D9A     1236          LA    R6,5(0,R6)            INC OUTPUT LINE ADDRESS
000DBA 465C 2D9A                    1237          BCT   R5,GETWORD
                                    1238          PUT   MACHLIST,MIXOUT
000DCC 47F0 2468          00468     1243          B     CYCLE
```

```
LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                   1245 *        **********************************************************************
                                   1246 *        ***                                                                ***
                                   1247 *        ***        JUMPS                                                   ***
                                   1248 *        ***                                                                ***
                                   1249 *        **********************************************************************
                                   1250 *        JUMP  JMP: F=0
                                   1251 *              JSJ: F=1
                                   1252 *              JCV: F=2
                                   1253 *              JNOV: F=3
                                   1254 *              JL: F=4
                                   1255 *              JE: F=5
                                   1256 *              JG: F=6
                                   1257 *              JGE: F=7
                                   1258 *              JNE: F=8
                                   1259 *              JLE: F=9
00CDD0 1844                        1260 JUMP     SR    R4,R4                 ZERO
00CDD2 4340 D4DE          004DE    1261          IC    R4,BYTEFOUR           F(INST)
00CDD6 415C 0009          00009    1262          LA    R5,9
00CDDA 1945                        1263          CR    R4,R5
00CDDC 4720 CC8A          0108A    1264          BH    ERROR                 F TOO LARGE?
00CDE0 894C 0002          00002    1265          SLL   R4,2                  F TIMES 4
00CDE4 47F4 2DE8          00DE8    1266          B     **4(R4)               JUMP TO APPROPRIATE OPERATOR
00CDE8 47F0 2E78          00E78    1267          B     SIMJUMP
00CDEC 47F0 2E86          00E86    1268          B     JREGCK
00CDF0 47F0 2E10          00E10    1269          B     JCV
00CDF4 47F0 2E20          00E20    1270          B     JNOV
00CDF8 47F0 2E30          00E30    1271          B     JL
00CDFC 47F0 2E3C          00E3C    1272          B     JE
00CE00 47F0 2E48          00E48    1273          B     JG
00CE04 47F0 2E54          00E54    1274          B     JGE
00CE08 47F0 2E60          00E60    1275          B     JNE
00CE0C 47F0 2E6C          00E6C    1276          B     JLE
00CE10 9500 D279  00279            1277 JCV      CLI   OVTOG,B'00000000'   TOGGLE OFF?
00CE14 4780 2468          00468    1278          BE    CYCLE
00CE18 9200 D279  00279            1279          MVI   OVTOG,B'00000000'   TURN OFF TOGGLE
00CE1C 47F0 2E78          00E78    1280          B     SIMJUMP
00CE20 9500 D279  00279            1281 JNOV     CLI   OVTOG,B'00000000'   TOGGLE OFF?
00CE24 478C 2E78          00E78    1282          BE    SIMJUMP               TAKE JUMP
00CE28 9200 D279  00279            1283          MVI   OVTOG,B'00000000'   TURN IT OFF
00CE2C 47F0 2468          00468    1284          B     CYCLE
00CE30 9502 D27A  00C7A            1285 JL       CLI   COMPI,B'00000010'   LESS THAN?
00CE34 4780 2E78          00E78    1286          BE    SIMJUMP               TAKE JUMP
00CE38 47F0 2468          00468    1287          B     CYCLE
00CE3C 9500 D27A  00C7A            1288 JE       CLI   COMPI,B'00000000'   COMPARE EQUAL?
00CE40 4780 2E78          00E78    1289          BE    SIMJUMP               TAKE JUMP
00CE44 47F0 2468          00468    1290          B     CYCLE
00CE48 9501 D27A  00C7A            1291 JG       CLI   COMPI,B'00000001'   GREATER THAN?
00CE4C 478C 2E78          00E78    1292          BE    SIMJUMP               TAKE JUMP
00CE50 47F0 2468          00468    1293          B     CYCLE
00CE54 9501 D27A  00C7A            1294 JGE      CLI   COMPI,B'00000001'   GREATER THAN?
00CE58 478C 2E78          00E78    1295          BE    SIMJUMP
00CE5C 47F0 2E3C          00E3C    1296          B     JE                    EQUAL
00CE60 9500 D27A  00C7A            1297 JNE      CLI   COMPI,B'00000000'   NOT EQUAL?
00CE64 477C 2E78          00E78    1298          BNE   SIMJUMP
00CE68 47FC 2468          00468    1299          B     CYCLE
00CE6C 9502 D27A  00C7A            1300 JLE      CLI   COMPI,B'00000010'   LESS THAN?
00CE70 4780 2E78          00E78    1301          BE    SIMJUMP
00CE74 47F0 2E3C          00E3C    1302          B     JE                    EQUAL?
00CE78 4130 D4FC          004F0    1303 SIMJUMP  LA    R3,MIXCORE
00CE7C 18A3                        1304          SR    R10,R3
```

```
   LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
000E7E 88A0 0002              00002  1305           SRL   R10,2              MIX LOCATION
000E82 50AC D2A8              02A8   1306           ST    R10,JREG
000E86 45E0 268A              0068A  1307 JREGCK    BAL   R14,ABADR          GET 360 ADDRESS
000E8A 58A0 D280              00280  1308           L     R10,IBMCORE        R10 HAS ADDRESS
000E8E 47F0 2468              00468  1309           BC    15,CYCLE
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                  1311 *        ********************************************************************
                                  1312 *        ***                                                              ***
                                  1313 *        ***        REGJUMPS                                              ***
                                  1314 *        ***                                                              ***
                                  1315 *        ********************************************************************
                                  1316 *        REGJUMP   JUMP IF REGISTER SATISFIES CONDITION
                                  1317 *                F=0: NEGATIVE
                                  1318 *                F=1: ZERO
                                  1319 *                F=2: POSITIVE
                                  1320 *                F=3: NONNEGATIVE
                                  1321 *                F=4: NONZERO
                                  1322 *                F=5: NONPOSITIVE
                                  1323 *                OTHERWISE NOTHING HAPPENS
00CE92 1B44                       1324 REGJUMP   SR     R4,R4                 ZERO
00CE94 4340 D4DE          004DE   1325           IC     R4,BYTEFOUR
00CE98 4150 0005          00005   1326           LA     R5,5
00CE9C 1945                       1327           CR     R4,R5
00CE9E 4720 C08A          0108A   1328           BH     FERROR                F > 5?
00CEA2 8940 0002          00002   1329           SLL    R4,2                  F TIMES FOUR
00CEA6 1B55                       1330           SR     R5,R5
00CEA8 4350 D4DF          004DF   1331           IC     R5,BYTEFIVE           GET OP CODE
00CEAC 4130 C028          00028   1332           LA     R3,40
00CEB0 1853                       1333           SR     R5,R3                 OP CODE - 40
00CEB2 8950 0002          00002   1334           SLL    R5,2                  TIMES FOUR
00CEB6 5BF5 D288          00288   1335           L      R15,AREG(R5)          FETCH REG CONTENTS
00CEBA 45E0 25C6          005C6   1336           BAL    R14,TWOCOMP
00CEBE 185F                       1337           LR     R5,R15                TWO'S COMPLEMENT
00CEC0 47F4 2EC4          00EC4   1338           B      *+4(R4)               BRANCH TABLE REG COND
00CEC4 47F0 2EDC          00EDC   1339           B      REGNEG
00CEC8 47F0 2EE6          00EE6   1340           B      REGZERO
00CECC 47F0 2EF0          00EF0   1341           B      REGPLUS
00CED0 47F0 2EFA          00EFA   1342           B      REGNNEG
00CED4 47F0 2F04          00F04   1343           B      REGNZERO
00CED8 47F0 2F0E          00F0E   1344           B      REGNPLUS
00CEDC 1255                       1345 REGNEG    LTR    R5,R5
00CEDE 4740 2E78          00E78   1346           BM     SIMJUMP
00CEE2 47F0 2468          00468   1347           B      CYCLE
00CEE6 1255                       1348 REGZERO   LTR    R5,R5
00CEE8 4780 2E78          00E78   1349           BZ     SIMJUMP
00CEEC 47F0 2468          00468   1350           B      CYCLE
00CEF0 1255                       1351 REGPLUS   LTR    R5,R5
00CEF2 4720 2E78          00E78   1352           BP     SIMJUMP
00CEF6 47F0 2468          00468   1353           B      CYCLE
00CEFA 1255                       1354 REGNNEG   LTR    R5,R5
00CEFC 4740 2468          00468   1355           BM     CYCLE
00CF00 47F0 2E78          00E78   1356           B      SIMJUMP
00CF04 1255                       1357 REGNZERO  LTR    R5,R5
00CF06 4780 2468          00468   1358           BZ     CYCLE
00CF0A 47F0 2E78          00E78   1359           B      SIMJUMP
00CF0E 1255                       1360 REGNPLUS  LTR    R5,R5
00CF10 4720 2468          00468   1361           BP     CYCLE
00CF14 47F0 2E78          00E78   1362           B      SIMJUMP
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                   1364 *        ********************************************************************
                                   1365 *        ***                                                              ***
                                   1366 *        ***       REGOPS                                                 ***
                                   1367 *        ***                                                              ***
                                   1368 *        ********************************************************************
                                   1369 *        REGOPS  THE ADDRESS M IS USED AS A SIGNED NUMBER TO MODIFY
                                   1370 *                THE REGISTER CONTENTS DEPENDING ON F.
                                   1371 *                F=0: INC
                                   1372 *                F=1: DEC
                                   1373 *                F=2: ENT
                                   1374 *                F=3: ENTN
000F18 1844                        1375 REGOPS   SR     R4,R4              ZERO
000F1A 4340 D4DE          004DE    1376          IC     R4,BYTEFOUR        GET F
000F1E 415C 00C3          00003    1377          LA     R5,3
000F22 1945                        1378          CR     R4,R5
000F24 4720 CC8A          0108A    1379          BH     FERROR             F > 3?
000F28 58F0 D4D8          004D8    1380          L      R15,MIXLOC         GET LOCATION
000F2C 45E0 25C6          005C6    1381          BAL    R14,TWOCOMP        GET 2'S COMP
000F30 187F                        1382          LR     R7,R15             TREAT AA AS SIGNED NUMBER
000F32 1855                        1383          SR     R5,R5
000F34 4350 D4DF          004DF    1384          IC     R5,BYTEFIVE        GET CP CODE
000F38 4130 C030          00030    1385          LA     R3,48
000F3C 1853                        1386          SR     R5,R3              CP CODE - 48
000F3E 8950 C002          00002    1387          SLL    R5,2               R5 INDECATES WHICH REG
000F42 58F5 D288          00288    1388          L      R15,AREG(R5)       GET REG CONTENTS
000F46 45E0 25C6          005C6    1389          BAL    R14,TWOCOMP
000F4A 186F                        1390          LR     R6,R15             TWO'S COMPLEMENT
000F4C 894C 0002          00002    1391          SLL    R4,2
000F50 47F4 2F54          00F54    1392          B      *+4(R4)            BRANCH ON F
000F54 47FC 2F74          00F74    1393          B      INC
000F58 47F0 2F70          00F70    1394          B      DEC
000F5C 47F0 2F64          00F64    1395          B      ENT
000F60 1337                        1396          LCR    R3,R7              NEGATE M
000F62 1873                        1397          LR     R7,R3
000F64 18F7                        1398 ENT      LR     R15,R7
000F66 45E0 25DE          005DE    1399          BAL    R14,RECCMP
000F6A 187F                        1400          LR     R7,R15
000F6C 47FC 2FCA          00FCA    1401          B      STORIT
000F70 1337                        1402 DEC      LCR    R3,R7              NEGATE M
000F72 1873                        1403          LR     R7,R3              BECOMES INCREASE
000F74 1A76                        1404 INC      AR     R7,R6
000F76 1833                        1405          SR     R3,R3              ZERO
000F78 1953                        1406          CR     R5,R3              AREG
000F7A 47F0 2FA6          00FA6    1407          B      CHECKA
000F7E 4130 0022          00022    1408          LA     R3,34
000F82 1953                        1409          CR     R5,R3              XREG
000F84 47F0 2FA6          00FA6    1410          B      CHECKA
000F88 4130 0FFF          00FFF    1411          LA     R3,4095            IT IS INDEX REG
000F8C 1973                        1412          CR     R7,R3              2 BYTES PER IREG
000FEE 4720 C092          01092    1413          BH     SIZERROR
000F92 1183                        1414          LNR    R8,R3              -4095
000F94 1978                        1415          CR     R7,R8
000F96 4740 C092          01092    1416          BL     SIZERROR
000F9A 18F7                        1417          LR     R15,R7
000F9C 45EC 25DE          005DE    1418          BAL    R14,RECCMP
000FA0 187F                        1419          LR     R7,R15             SIGN AND MAGNITUDE
000FA2 47F0 2FCA          00FCA    1420          B      STORIT
000FA6 18F7                        1421 CHECKA   LR     R15,R7
000FA8 45E0 25DE          005DE    1422          BAL    R14,RECCMP
000FAC 187F                        1423          LR     R7,R15             SIGN AND MAGNITUDE
```

```
  LOC   OBJECT CODE     ADDR1  ADDR2   STMT   SOURCE STATEMENT

000FAE 5070 D4C8             004C8   1424           ST     R7,PACKAREA
000FB2 9140 D4C8       CC4CE         1425           TM     PACKAREA,B'01OC0000'
000FB6 4780 2FCA             00FCA   1426           BC     8,STORIT
000FBA 9201 D279       0C279         1427           MVI    CVTCG,X'01'
000FBE 8970 0002             00002   1428           SLL    R7,2                  MODULO 30 BITS
000FC2 8870 0002             00002   1429           SRL    R7,2
000FC6 45E0 26BC             006BC   1430           BAL    R14,CVRFLC
000FCA 5075 D288             00288   1431 STORIT    ST     R7,AREG(R5)
000FCE 47F0 2468             00468   1432           B      CYCLE
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                  1434 *        ********************************************************************
                                  1435 *        ***                                                              ***
                                  1436 *        ***       COMPARE                                                ***
                                  1437 *        ***                                                              ***
                                  1438 *        ********************************************************************
                                  1439 *        COMPARE   F(REG)::F(M)
                                  1440 *                  F(REG) > F(M): COMPI=BL1'00000001'
                                  1441 *                  F(REG) < F(M): COMPI = BL1'00000011'
                                  1442 *                  F(REG) = F(M): COMPI = BL1'00000000'
                                  1443 *                  NOTE: +0 = -0
000FD2 45E0 2508            00508 1444 COMPARE   BAL     R14,GETV            F(M) IN VREF
000FD6 4110 D4E8            004E8 1445           LA      R1,VREF
000FDA 45E0 2648            00648 1446           BAL     R14,PACK
000FDE 45E0 25C6            005C6 1447           BAL     R14,TWOCOMP
000FE2 184F                       1448           LR      R4,R15              R4 HAS PACKED F(M)
000FE4 1B55                       1449           SR      R5,R5               ZERO
000FE6 4350 D4DF            004DF 1450           IC      R5,BYTEFIVE         OP CODE
000FEA 4130 0038            00038 1451           LA      R3,56
000FEE 1B53                       1452           SR      R5,R3               OP CODE - 56 = 0,1,2,...
000FF0 8950 0002            00002 1453           SLL     R5,2                TIMES FOUR
000FF4 5855 D288            00288 1454           L       R5,AREG(R5)         CONTENTS OF DESIRED REG
000FF8 18F5                       1455           LR      R15,R5
000FFA 4110 D4E0            004E0 1456           LA      R1,MREF             PARMS FOR UNPACK
000FFE 45E0 25FC            005FC 1457           BAL     R14,UNPACK          MREF HAS UNPACKED REG
001002 45E0 2500            00500 1458           BAL     R14,GETREGV         VREF GETS F(REG)
001006 4110 D4E8            004E8 1459           LA      R1,VREF
00100A 45E0 2648            00648 1460           BAL     R14,PACK            PACK THE FIELD
00100E 45E0 25C6            005C6 1461           BAL     R14,TWOCOMP
001012 1954                       1462           CR      R5,R4
001014 4780 C024            01024 1463           BE      COMPEQU             F(REG) = F(M)
001018 4720 C02C            0102C 1464           BH      REGHI               F(REG) = F(M)
00101C 9202 D27A      0027A       1465           MVI     COMPI,X'02'
001020 47F0 C030            01030 1466           B       COMPSET
001024 920C D27A      0027A       1467 COMPEQU   MVI     COMPI,X'00'
001028 47F0 C030            01030 1468           B       COMPSET
00102C 9201 D27A      0027A       1469 REGHI     MVI     COMPI,X'01'
001030 47F0 2468            00468 1470 COMPSET   BC      15,CYCLE
001034 1845                       1471 MIXADR    LR      R4,R5               R5 HAS LOCATION IN BINARY
001036 8840 0002            00002 1472           SLA     R4,2                R4 NOW HAS LOCATION TIMES FOUR
00103A 41A0 D4F0            004F0 1473           LA      R10,MIXCORE         START OF MIXCORE
00103E 1AA4                       1474           AR      R10,R4              R10 HAS ADDRESS IN MIXCORE
001040 07FB                       1475           BCR     15,R11              RETURN
```

```
  LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                     1477 *        ********************************************************************
                                     1478 *        ***                                                              ***
                                     1479 *        ***        SET ERROR CODES                                       ***
                                     1480 *        ***                                                              ***
                                     1481 *        ********************************************************************
001042 41F0 0004              00004  1482 EMSGA    LA     R15,4                  ERROR CODE TO R15
001046 41EC 2C9C              00090  1483          LA     R14,READCARD
00104A 47F0 2326              00326  1484          B      MSGOUT
001C4E 41FC C008              00008  1485 EMSGB    LA     R15,8                  ERROR CODE
001052 41EC 2C90              00090  1486          LA     R14,READCARD
001056 47F0 2326              00326  1487          B      MSGOUT
00105A 41F0 00CC              0000C  1488 EMSGC    LA     R15,12                 ERROR CODE
00105E 41EC 2C90              00090  1489          LA     R14,READCARD
001C62 47F0 2326              00326  1490          B      MSGOUT
001066 41F0 0010              00010  1491 EMSGD    LA     R15,16                 ERROR CODE
00106A 41EC 2090              00090  1492          LA     R14,READCARD
00106E 47F0 2326              00326  1493          B      MSGOUT
001C72 41F0 0084              00084  1494 IXERROR  LA     R15,132                ERROR CODE
001C76 47FC 2326              00326  1495          B      MSGOUT
C0107A 41F0 0C88              00088  1496 ADRERRCR LA     R15,136                ERROR CODE
001C7E 47F0 2326              00326  1497          B      MSGOUT
001C82 41F0 0C8C              C008C  1498 MEMERR   LA     R15,140                ERROR CODE
001086 47F0 2326              00326  1499          B      MSGOUT
00108A 41FC CC90              00090  1500 FERRCR   LA     R15,144                ERROR CODE
00108E 47F0 2326              00326  1501          B      MSGOUT
C01092 41FC CC94              00094  1502 SIZERRCR LA     R15,148                ERROR CODE
001096 47F0 2326              00326  1503          B      MSGOUT
001C9A 41FC C098              0C098  1504 ICERRCR  LA     R15,152                ERROR CODE
00109E 47F0 2326              00326  1505          B      MSGOUT
                                     1506 *
0010A8                               1507          LTORG
0010A8 D4C1C3C8D3C5E2E3              1508                 =C'MACHLIST'
0C10BC 40C5E7C5C3E4E3C5              1509                 =CL120' EXECUTION TIME UNITS:'
                                     1510                 =CL80' LOC      : ARITHMETIC OVERFLOW,EXECUTION CONTINUEX
0C1128 40D3D6C34C4C4C4C                                   S  MODULE WORD SIZE'
001178 0CC04E20                      1511                 =F'20000'
00117C 0085                          1512                 =H'133'
00117E 0050                          1513                 =B'0000000001010000'
0C118C C3D6C34B4B4B                  1514                 =C'LOC...'
001186 84                            1515                 =B'10000100'
001187 1C                            1516                 =P'1'
                                     1517 DUMMY    DCB    DDNAME=MIXINPUT,BLKSIZE=80,DSORG=PS,MACRF=(GM),RECFM=F,L*
                                                          RECL=80,EODAD=MYEOF
```

```
**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    1574 *        ************************************************************
                                    1575 *        ***                                                      ***
                                    1576 *        ***        MESSAGES AND CONSTANTS                        ***
                                    1577 *        ***                                                      ***
                                    1578 *        ************************************************************
0011E8 F6F3                         1579 BYTESIZE DC    C'63'
0011EA F4F0F9F5                     1580 ADMAX    DC    C'4095'
0011EE 01073741823C                 1581 MAXNO    DC    PL6'1073741823'
0011F4 D3C6C3404C4C4040             1582 MLCC     DC    C'LOC   : EXECUTION TERMINATED'
001212 F05C5C5CC5D5D5D6             1583 ERRHDR   DC    C'0***ERROR***'
00121E 4040404040404040             1584 MBLANK   DC    CL132' '
0012A2 D4C9E740D4C1C3C8             1585 HEADER   DC    C'MIX MACHINE LANGUAGE PROGRAM LISTING'
0012C6 C9D5E5C1D3C9C440             1586 MSGA     DC    C'INVALID LOCATION IN COLUMN 7-10; NOT IN RANGE 0-3999'
0012FA C9D5E5C1D3C9C440             1587 MSGB     DC    C'INVALID WORD COUNT IN COLUMN 6,NOT IN RANGE 0-7'
                                    1588 MSGC     DC    C'WORD COUNT NOT CONSISTENT WITH NUMBER OF WORDS ON CARDX
001325 E6D6D5C440C3D6E4                           '
00135F C9D5E5C1D3C9C440             1589 MSGD     DC    C'INVALID CHARACTER OR SIGN PUNCHED ON CARD'
                                    1590 MSGE     DC    C'OP,F,OR I FIELD EXCEEDS BYTESIZE OF 63 OR AA FIELD EXCX
001388 D6D76BC66BD6D940                           EEDS 2 BYTESIZE OF 4095'
0013D5 F0C5E7C5C3E4E3C5             1591 MSGF     DC    C'0EXECUTION NOT ATTEMPTED DUE TO INPUT ERRORS'
                                    1592 MSGG     DC    C'0CARD READER EMPTY OR VALID TRANSFER CARD NOT ENCOUNTEX
0014C1 F0C3C1D9C44CD5C5                           RED'
00143A C9D3D3C5C7C1D340             1593 MSGAX    DC    C'ILLEGAL INDEX SPECIFIED'
001451 C5C6C6C5C3E3C9E5             1594 MSGBX    DC    C'EFFECTIVE ADDRESS TOO LARGE'
00146C D4C5D4D6D5E840D9             1595 MSGCX    DC    C'MEMORY REFERENCE BEYOND LIMITS OF MIXCORE'
001495 C9D3D3C5C7C1D340             1596 MSGDX    DC    C'ILLEGAL F SPECIFIED'
                                    1597 MSGEX    DC    C'ILLEGAL ATTEMPT TO LOAD INDEX REGISTER WITH VALUE GREAX
0014A8 C9D3D3C5C7C1D340                           TER THAN 4095'
0014EB E4D5C4C5C6C9D5C5             1598 MSGFX    DC    C'UNDEFINED IO OPERATION'
                                    1599 *        ************************************************************
                                    1600 *        ***                                                      ***
                                    1601 *        ***        TRANSLATE TABLE                               ***
                                    1602 *        ***                                                      ***
                                    1603 *        ************************************************************
001501 40C1C2C3C4C5C6C7             1604 INCODE   DC    X'40C1C2C3C4C5C6C7C8C900D1D2D30405'
001511 C6D7D8D50000E2E3             1605          DC    X'C6D7D8D50000E2E3E4E5E6E7E8E9F0F1'
001521 F2F3F4F5F6F7F8F9             1606          DC    X'F2F3F4F5F6F7F8F9848684C5D4E605C6'
001531 7E5B4C6E7C5E7A7D             1607          DC    X'7E5B4C6E7C5E7A7D00000000000000000'
001541 000C000000000000             1608          DC    X'000000000000000000000028322A2C00'
001551 000C0C000C0C0000             1609          DC    X'00000000000000000000003122283500'
001561 2D2F0000C0CC0000             1610          DC    X'2D2F0000000000000000002900001100'
001571 0000000000C0C000             1611          DC    X'00000000000000000000360034373000'
001581 000C00C0000C0000             1612          DC    X'00000000000000000000000000000000'
001591 0000000000C00000             1613          DC    X'00000000000000000000000000000000'
0015A1 0000000000000000             1614          DC    X'00000000000000000000000000000000'
0015B1 000000000C0CCC00             1615          DC    X'00000000000000000000000000000000'
0015C1 0001020304050607             1616          DC    X'0001020304050607080900000000000'
0015D1 0008C0D0E0F10111             1617          DC    X'0008C0D0E0F1011121300000000000'
0015E1 000016171819 1A1B            1618          DC    X'000016171819 1A1B1C1D00000000000'
0015F1 1E1F202122232425             1619          DC    X'1E1F20212223242526270000000000'
```

**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    1621 *        ********************************************************************
                                    1622 *        ***                                                              ***
                                    1623 *        ***       OPERATOR ADDRESS TABLE                                 ***
                                    1624 *        ***                                                              ***
                                    1625 *        ********************************************************************
001604                              1626 OPTABLE   DS    0F                  FORCE ALIGNMENT
001604 01                           1627           DC    BL1'00000001'
001605 000468                       1628           DC    AL3(CYCLE)          NOP
001608 02                           1629           DC    BL1'00000010'
001609 0006FA                       1630           DC    AL3(ADDSUB)
00160C 02                           1631           DC    BL1'00000010'
00160D 000FFA                       1632           DC    AL3(ADDSUB)
001610 02                           1633           DC    BL1'00000010'
001611 00072A                       1634           DC    AL3(MUL)
001614 02                           1635           DC    BL1'00000010'
001615 000770                       1636           DC    AL3(DIV)
001618 01                           1637           DC    BL1'00000001'
001619 00080C                       1638           DC    AL3(SPEC)           FLT,CHAR,NUM
00161C 02                           1639           DC    BL1'00000010'
00161D 0009B2                       1640           DC    AL3(SHIFT)          SHIFT OPERATORS
001620 01                           1641           DC    BL1'00000001'
001621 000B3E                       1642           DC    AL3(MOVE)
001624 02                           1643           DC    BL1'00000010'
001625 000B7E                       1644           DC    AL3(LDA)            LOAD A
001628 02                           1645           DC    BL1'00000010'
001629 000B92                       1646           DC    AL3(LDI)            LOAD INDEX REGS
00162C 02                           1647           DC    BL1'00000010'
00162D 000B92                       1648           DC    AL3(LDI)            LOAD INDEX REGS
001630 02                           1649           DC    BL1'00000010'
001631 000B92                       1650           DC    AL3(LDI)            LOAD INDEX REGS
001634 02                           1651           DC    BL1'00000010'
001635 000B92                       1652           DC    AL3(LDI)            LOAD INDEX REGS
001638 02                           1653           DC    BL1'00000010'
001639 000B92                       1654           DC    AL3(LDI)            LOAD INDEX REGS
00163C 02                           1655           DC    BL1'00000010'
00163D 000B92                       1656           DC    AL3(LDI)            LOAD INDEX REGS
001640 02                           1657           DC    BL1'00000010'
001641 000BD2                       1658           DC    AL3(LDX)            LOAD X
001644 02                           1659           DC    BL1'00000010'
001645 000BE6                       1660           DC    AL3(LDAN)           LOAD A NEG
001648 02                           1661           DC    BL1'00000010'
001649 000BFE                       1662           DC    AL3(LDIN)           LOAD INDEX REGS NEGATIVE
00164C 02                           1663           DC    BL1'00000010'
00164D 000BFE                       1664           DC    AL3(LDIN)           LOAD INDEX REGS NEGATIVE
001650 02                           1665           DC    BL1'00000010'
001651 000BFE                       1666           DC    AL3(LDIN)           LOAD INDEX REGS NEGATIVE
001654 02                           1667           DC    BL1'00000010'
001655 000BFE                       1668           DC    AL3(LDIN)           LOAD INDEX REGS NEGATIVE
001658 02                           1669           DC    BL1'00000010'
001659 000BFE                       1670           DC    AL3(LDIN)           LOAD INDEX REGS NEGATIVE
00165C 02                           1671           DC    BL1'00000010'
00165D 000BFE                       1672           DC    AL3(LDIN)           LOAD INDEX REGS NEGATIVE
001660 02                           1673           DC    BL1'00000010'
001661 000C42                       1674           DC    AL3(LDXN)           LDXN
001664 02                           1675           DC    BL1'00000010'
001665 000C5A                       1676           DC    AL3(STORE)          STORE AREG,ETC
001668 02                           1677           DC    BL1'00000010'
001669 000C5A                       1678           DC    AL3(STORE)          STORE AREG,ETC
00166C 02                           1679           DC    BL1'00000010'
00166D 000C5A                       1680           DC    AL3(STORE)          STORE AREG,ETC
```

```
  LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
001670 02                          1681          DC    BL1'00000010'
001671 000005A                     1682          DC    AL3(STORE)          STORE AREG,ETC
001674 02                          1683          DC    BL1'00000010'
001675 000005A                     1684          DC    AL3(STORE)          STORE AREG,ETC
001678 02                          1685          DC    BL1'00000010'
001679 000005A                     1686          DC    AL3(STORE)          STORE AREG,ETC
00167C 02                          1687          DC    BL1'00000010'
00167D 000005A                     1688          DC    AL3(STORE)          STORE AREG,ETC
001680 02                          1689          DC    BL1'00000010'
001681 000005A                     1690          DC    AL3(STORE)          STORE AREG,ETC
001684 02                          1691          DC    BL1'00000010'
001685 000005A                     1692          DC    AL3(STORE)          STORE AREG,ETC
001688 01                          1693          DC    BL1'00000001'
001689 000005A                     1694          DC    AL3(STORE)          STORE AREG,ETC
00168C 01                          1695          DC    BL1'00000001'
00168D 0000468                     1696          DC    AL3(CYCLE)          JBUS
001690 FA                          1697          DC    BL1'11111010'
001691 000CDB                      1698          DC    AL3(IOC)
001694 FA                          1699          DC    BL1'11111010'
001695 000D16                      1700          DC    AL3(IN)
001698 FA                          1701          DC    BL1'11111010'
001699 000D70                      1702          DC    AL3(CUT)
00169C 01                          1703          DC    BL1'00000001'
00169D 0000468                     1704          DC    AL3(CYCLE)          JRED
0016A0 01                          1705          DC    BL1'00000001'
0016A1 000DDC                      1706          DC    AL3(JUMP)
0016A4 01                          1707          DC    BL1'00000001'
0016A5 000E92                      1708          DC    AL3(REGJUMP)
0016A8 01                          1709          DC    BL1'00000001'
0016A9 000E92                      1710          DC    AL3(REGJUMP)
0016AC 01                          1711          DC    BL1'00000001'
0016AD 000E92                      1712          DC    AL3(REGJUMP)
0016B0 01                          1713          DC    BL1'00000001'
0016B1 000E92                      1714          DC    AL3(REGJUMP)
0016B4 01                          1715          DC    BL1'00000001'
0016B5 000E92                      1716          DC    AL3(REGJUMP)
0016B8 01                          1717          DC    BL1'00000001'
0016B9 000E92                      1718          DC    AL3(REGJUMP)
0016BC 01                          1719          DC    BL1'00000001'
0016BD 000E92                      1720          DC    AL3(REGJUMP)
0016C0 01                          1721          DC    BL1'00000001'
0016C1 000E92                      1722          DC    AL3(REGJUMP)
0016C4 01                          1723          DC    BL1'00000001'
0016C5 000F18                      1724          DC    AL3(REGCPS)
0016C8 01                          1725          DC    BL1'00000001'
0016C9 000F18                      1726          DC    AL3(REGCPS)
0016CC 01                          1727          DC    BL1'00000001'
0016CD 000F18                      1728          DC    AL3(REGCPS)
0016D0 01                          1729          DC    BL1'00000001'
0016D1 000F18                      1730          DC    AL3(REGCPS)
0016D4 01                          1731          DC    BL1'00000001'
0016D5 000F18                      1732          DC    AL3(REGCPS)
0016D8 01                          1733          DC    BL1'00000001'
0016D9 000F18                      1734          DC    AL3(REGCPS)
0016DC 01                          1735          DC    BL1'00000001'
0016DD 000F18                      1736          DC    AL3(REGCPS)
0016E0 02                          1737          DC    BL1'00000010'
0016E1 000F18                      1738          DC    AL3(REGCPS)
0016E4 02                          1739          DC    BL1'00000010'
0016E5 000FD2                      1740          DC    AL3(COMPARE)
```

```
   LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT
0016E8 02                          1741            DC     BL1'00000010'
0016E9 000FD2                      1742            DC     AL3(COMPARE)
0016EC 02                          1743            DC     BL1'00000010'
0016ED 000FD2                      1744            DC     AL3(COMPARE)
0016F0 02                          1745            DC     BL1'00000010'
0016F1 000FD2                      1746            DC     AL3(COMPARE)
0016F4 02                          1747            DC     BL1'00000010'
0016F5 000FD2                      1748            DC     AL3(COMPARE)
0016F8 02                          1749            DC     BL1'00000010'
0016F9 000FD2                      1750            DC     AL3(COMPARE)
0016FC 02                          1751            DC     BL1'00000010'
0016FD 000FD2                      1752            DC     AL3(COMPARE)
001700 01                          1753            DC     BL1'00000001'
0017C1 000FD2                      1754            DC     AL3(COMPARE)
                                   1755 *
                                   1756 *
                                   1757            DCBD   DSORG=PS           FORM THE DCB DSECT
                                   1758                   *,***   IHB069  DEVD NOT SPECIFIED-ALL ASSUMED
```

**********APPENDIX C: ASSEMBLED PROGRAM LISTING**********

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                   1893 *        ********************************************************************
                                   1894 *        ***                                                              ***
                                   1895 *        ***        DSECT                                                 ***
                                   1896 *        ***                                                              ***
                                   1897 *        ********************************************************************
000000                             1898 STORAGE  DSECT
000000                             1899 SAVEAREA DS    18F
000048                             1900 MIXINPUT DS    24F
0000A8                             1901 MACHLIST DS    24F
000108                             1902 PARMLIST DS    2F
000110                             1903 INCARD   DS    0CL80
000110                             1904          DS    CL5
000115                             1905 WCOUNT   DS    CL1
000116                             1906 LOC      DS    CL4
00011A                             1907 WCRD1    DS    CL10
000124                             1908 WCRD2    DS    CL10
00012E                             1909 WCRD3    DS    CL10
000138                             1910 WCRD4    DS    CL10
000142                             1911 WCRD5    DS    CL10
00014C                             1912 WCRD6    DS    CL10
000156                             1913 WCRD7    DS    CL10
000160                             1914 MIXOUT   DS    CL133
0001E5                             1915 MSGLINE  DS    CL133
00026A                             1916 COUREM   DS    PL14
000278                             1917 SWITCH   DS    BL1
000279                             1918 CVTCG    DS    BL1
00027A                             1919 COMPI    DS    BL1
000278                             1920 LEFT     DS    BL1
00027C                             1921 RIGHT    DS    BL1
000280                             1922 IBMCORE  DS    F
000284                             1923 MEMREF   DS    F
000288                             1924 AREG     DS    F
00028C                             1925 IREG1    DS    F
000290                             1926 IREG2    DS    F
000294                             1927 IREG3    DS    F
000298                             1928 IREG4    DS    F
00029C                             1929 IREG5    DS    F
0002A0                             1930 IREG6    DS    F
0002A4                             1931 XREG     DS    F
0002A8                             1932 JREG     DS    F
0002AC                             1933 ZERO     DS    F
0002B0                             1934 CLOCK    DS    F
0002B4                             1935 CONVERT  DS    F
0002B8                             1936 PSAVE    DS    16F
0002F8                             1937 UNPSAVE  DS    16F
000338                             1938 GETSAVE  DS    16F
000378                             1939 MSAVE    DS    16F
0003B8                             1940 FSAVE    DS    16F
0003F8                             1941 TWOSAVE  DS    16F
000438                             1942 RECSAVE  DS    16F
000478                             1943 SAVEO    DS    16F
0004B8                             1944 FCOUNT   DS    D
0004C0                             1945 PLOC     DS    D
0004C8                             1946 PACKAREA DS    D
0004D0                             1947 AUXSAVE  DS    D
0004D8                             1948 INST     DS    0D
0004E8                             1949 MIXLOC   DS    F
0004EC                             1950          DS    BL1
0004ED                             1951 BYTETHRE DS    BL1
0004EE                             1952 BYTEFOUR DS    BL1
```

```
   LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

000C4DF                            1953 BYTEFIVE CS     BL1
000C4EC                            1954 MREF     DS     OD
000C4EC                            1955 MSIGN    DS     BL1
000C4E1                            1956 MONE     DS     BL1
000C4E2                            1957 MTWO     DS     BL1
000C4E3                            1958 MTHREE   DS     BL1
000C4E4                            1959 MFOUR    DS     BL1
000C4E5                            1960 MFIVE    DS     BL1
000C4E8                            1961 VREF     DS     OD
000C4E8                            1962 VSIGN    DS     BL1
000C4E9                            1963 VONE     DS     BL1
000C4EA                            1964 VTWO     DS     BL1
000C4EB                            1965 VTHREE   DS     BL1
000C4EC                            1966 VFOUR    DS     BL1
000C4ED                            1967 VFIVE    DS     BL1
000C4F0                            1968 MIXCORE  DS     F
                                   1969          END
```

BIBLIOGRAPHY

1. Hamming, R.W., "One Man's View of Computer Science," Journal of the Association for Computing Machinery, vol. 16, No. 1, p. 3-12, January 1969.

2. Knuth, D.E., "Fundamental Algorithms," The Art of Computer Programming, vol. 1, p. 120-227, Addison-Wesley, 1968.

3. Rosin, R.F., "Supervisory and Monitor Systems," Computing Surveys, vol. 1, No. 1, p. 37-54, March 1969.

4. Maurer, W.D., "An Improved Hash Code for Scatter Storage," Communications of the ACM, vol. 11, No. 1, p. 35-38, January 1968.

5. Morris, R., "Scatter Storage Techniques," Communications of the ACM, vol. 11, No. 1, p. 38-44, January 1968.

6. Hext, J.B. and Winings, J.W., "An Automatic Grading Scheme for Simple Programming Exercises," Communications of the ACM, vol. 12, No. 5, p. 272-275, May 1969.

7. Bacon, R.F., An Investigation of an Automatic Machine Grading System for Assembly Language Instruction, M.S. Thesis, U.S. Naval Postgraduate School, Monterey, California, 1969.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Chief of Naval Operations (OP-91) Department of the Navy Washington, D. C. 20350	1
4.	Prof. G. L. Barksdale, Jr. (thesis advisor) Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
5.	Professor W. S. Brainerd Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
6.	LCDR Max N. Akers (student) USS LAFAYETTE SSBN-616 (Gold) FPO New York, N.Y. 09501	1

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

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

1. ORIGINATING ACTIVITY *(Corporate author)*

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

A Proposed Programming System for Knuth's MIX Computer.

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

Master's Thesis; (June 1969)

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

Max Neil Akers

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1969	105	7

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A programming system using a hypothetical computer is proposed for use in teaching machine and assembly language programming courses. Major components such as monitor, assembler, interpreter, grader and diagnostics are described. The Interpreter is programmed and documented for use on an IBM 360/67. The interpreter can be used for teaching machine language programming and can be incorporated into the proposed programming system.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified

Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Student Programming System						
Mix Computer						
Mix Interpreter						
Tutorial Assembly Language						
Mix Programming System Design						