NBSIR 77-1230

Crossi - A Univac Processor Which Assembles Code for Interdata Minicomputers

Carol V. Young

Computer Services Division
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

April 1977

U. S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

NBSIR 77-1230

CROSSI - A UNIVAC PROCESSOR WHICH ASSEMBLES CODE FOR INTERDATA MINICOMPUTERS

Carol V. Young

Computer Services Division
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

April 1977

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, *Secretary*
Dr. Betsy Ancker-Johnson, *Assistant Secretary for Science and Technology*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Acting Director*

Contents

ACKNOWLEDGEMENT

The author wishes to acknowledge the generous assistance and day to day guidance of Mr. Richard Freemire of the IBS Laboratory Automation group and Mr. Edward Barkmeyer of the NBS Computer Services Division. The author also wishes to thank Ms. Joan Dailey for her help in the preparation of this document.

CROSSI - A UNIVAC PROCESSOR WHICH ASSEMBLES CODE FOR INTERDATA MINICOMPUTERS

by Carol V. Young

Instructions are given for the use of CROSSI, a computer program which assembles Interdata assembly code on the UNIVAC 1108 producing an assembly listing and relocatable code for the minicomputer. The assembly statements, relocatable output formats, and the structure of the processor are described. A pre-processor is given for converting CROSSI's zoned relocatable format to Interdata's zoned relocatable format. A method of reconfiguring the CROSSI Processor to assemble code for newer, upward compatible minicomputer models is explained.

Key Words: Assembler; cross-assembler; machine code; relocatable code.

DISCLAIMER: Mention of commercially produced equipment or programs by brand name may not be construed as an endorsement or approval of such products by the National Bureau of Standards. No tests or evaluations of these products have been conducted.

1. INTRODUCTION

CROSSI is an UNIVAC 1108 processor which assembles Interdata assembly source statements producing source listings and relocatable machine code. It is a two-pass assembler and accepts as input all source statements acceptable to the Interdata OS Assembler (03-025).

It is presumed that the reader is familiar with UNIVAC 1100 OS control statements (see UNIVAC Manual UP-4144 or UP-7824) and Interdata assembly programming (see Interdata Model 70 User's Manual).

Some characteristics and advantages of CROSSI are:

1. Permits creating and updating of Interdata assembly source files on UNIVAC mass storage from either batch or interactive mode.

2. Permits use of other UNIVAC processors, including the Editor, to manipulate Interdata files.

3. Allows programmer to specify instruction sets for Interdata Models 3, 4, 74, 70, and three configurations of Model 7/16.

4. Allows additional operations for newer models which are compatible with the Model 70.

5. Generates relocatable machine code in two formats: zoned paper tape format using ASCII printing characters or packed format on magnetic tape.

6. Produces program listing in a variety of formats.

7. Accepts all Interdata assembler directives.

8. Accepts extended branch mnemonic instructions.

9. Flags statements containing errors with the same flags as the Interdata OS assembler.

CROSSI has successfully assembled Interdata programs including the RTOS operating system and the Dataplot 70 routines.

2. ASSEMBLER SOURCE STATEMENTS

2.1 Comment Statements

Comments are descriptive text which can occupy an entire source statement. These statements are distinguished from an instruction by placing an asterisk (*) in column one. Text can occupy columns 2 through 72.

2.2 Instruction Statements

An Interdata instruction consists of four fields: the label field, the operation field, the operand field, and the comment field. If an instruction has a label, the label must start in column one. If the instruction statement does not have a label, column one must be blank. At least one blank must separate fields. No blanks can be embedded within the label or operation fields. A blank can appear in the operand field only if it is within a character string (DC C'.....'). Within the operand field the optional index register is enclosed in parentheses and a comma separates the other subfields. The comment field may contain any character and this field is always optional.

2.2.1 Statement Labels

Statement labels are symbols used to name a location or a value. Symbols consist of one to six characters. The first character must be alphabetic or one of three special characters ($, ., @). The remaining characters can be alphabetic or numeric.

When using CROSSI, it is advisable to avoid using the special character '@' as the first character of a statement label because insertion of such a statement into a runstream causes the UNIVAC

Operating System to process the image as a control statement. Such statements can be inserted into a mass storage file with the UNIVAC ELT processor using the D option (see Section 3.2.1 and 3.2.2). CROSSI will then have no problem reading these files.

2.2.2 Machine Instruction Sets

The following instruction sets, listed in the Model 70 User's Guide, are included in CROSSI:

53	Standard Instructions
4	High Speed Arithematic Instructions
4	High Speed I/O Instructions
4	Basic Model 4 Extensions
13	Floating Point Instructions
12	Basic Model 5 Extensions
27	Extended Model 5 Instructions

CROSSI permits the user to choose one of six combinations of these instruction sets by naming the target Interdata model. CROSSI supports the following models and assumes each to be configured as follows:

1.	Model 3	53	Standard Instructions
		8	High Speed Arithmetic and I/O Instructions
		--	
		61	Total
2.	Model 4	53	Standard Instructions
		8	High Speed Arithmetic and I/O Instructions
		13	Floating Point Instructions
		4	Basic Model 4 Extensions
		--	
		78	Total
3.	Model 74	53	Standard Instructions
	and	4	High Speed I/O Instructions
	Model 7/16A	11	Basic Model 5 Extensions (excludes SINT)
		23	Extended Model 5 Instructions (excludes ATL, ABL, RBL, RTL)
		--	
		91	Total
4.	Model 7/16B	53	Standard Instructions
		8	High Speed Arithmetic and I/O Instructions
		12	Basic Model 5 Extensions
		27	Extended Model 5 Instructions
		--	
		100	Total

5. Model 70	53	Standard Instructions
and	8	High Speed Arithmetic and I/O Instructions
Model 7/16C	13	Floating Point Instructions
	12	Basic Model 5 Extensions
	27	Extended Model 5 Instructions
	--	
	113	Total

6. Model 70/xxx Same Instruction Sets as Model 70 plus any operations added for newer Interdata models.

2.2.3 Extended Branch Mnemonic Instructions

CROSSI operation codes include:

1. 15 long branch extended mnemonics available on all Interdata models.

2. 15 register to register extended branch mnemonics available on all Interdata models.

3. 14 short extended branch mnemonics available on Model 74, Model 70 and the three configurations of Model 7/16.

2.2.4 Assembler Instruction Set

The following assembler directives (pseudo-ops) are included in CROSSI and operate in the same way as in the Interdata OS assembler except as noted below.

1. EQU - Equate Symbol

2. ENTRY - Identify Entry-Point Symbol

3. EXTRN - Identify External Symbol

4. DC - Define Constant
   Operands of DC instructions specify the following data type:

C'....'	Character constant
X'....'	Hexadecimal constant
A'....'	Address constant
H'....'	Halfword decimal constant
E'....'	Floating Point constant (single precision)
D'....'	Floating Point constant (double precision)

   If no data type is specified, halfword decimal constant is assumed for a numeric operand and address constant is assumed for a symbolic operand.

5. DS - Define Storage

6. DB - Define Byte
Following one or more define byte instructions, the value of the location counter for a non-define byte instruction is automatically forced to be even.

7. ORG - Sets Value and Mode of Location Counter

8. END - End Assembly and Transfer Address

9. DO - Conditional/Multiple Assembly Instruction

10. IF - Conditional Assembly

11. TITLE - Listing Title and Format Control

12. OPT - Specify Options
CROSSI will read an OPT statement and process only the label subfield. All other subfields will be ignored.

    a) LAB = nnnnnn The object program label nnnnnn will be written on the relocatable file in symbolic form (R or M option only).

    b) Number of pass option has no meaning since CROSSI is a two-pass assembler.

    c) PUNCH, NOPNCH, PRINT, NOPRINT, and SQCHK options are available on the processor call statement (see Sections 3.2.3, 3.2.4, 3.2.5).

    d) FLOAT, SCRI and GO options are automatic.

13. PAUSE - Assembly Pause
CROSSI will ignore this statement. The statement will not appear in the listing.

3. PROCESSOR CALL STATEMENT

CROSSI is a standard UNIVAC two-specification processor with two optional fields for designating the target Interdata model and a magnetic tape file for relocatable output. The processor call statement is as follows:

```
@CROSSI[,options] SI-ELTNAME[,SO-ELTNAME][,INTERDATA-MODEL][,MAG-FILE]
```

3.1 Specification Fields

The first and second specification fields are standard UNIVAC source input and source output fields (see I and U option below).

The third specification field codes the model number of the target Interdata computer as follows:

3	for Model 3
4	for Model 4
74	for Model 74
70	for Model 70
7/16A 7/16B 7/16C	for Model 7/16
70/xxx	for newer Model xxx

If this field is not present, Model 70 is assumed. CROSSI will permit, as a legal operation, only those operations available for the designated model (see Section 2.2.2).

The fourth specification field is named only with the M option and is ignored in all other cases. With the M option this field must specify an assigned, write-enabled, nine-track magnetic tape file onto which the relocatable file is to be written (see Section 3.2.4).

3.2 Options

3.2.1 I Option

This option inserts a new source language element into a mass storage file from the runstream. With this option the source language input parameter specifies the file and element into which the source language output is to be written. The source language output (SO) parameter is never used and should not be specified.

Example 1: Source is a card deck

```
@CROSSI,IL   FILE1.ELT1
source deck
@EOF
```

Example 2: Source is a SDF DATA file

```
@CROSSI,IL   FILE1.ELT1
@ADD,E SOURCEFILE.
```

In both examples above, CROSSI will insert the source statements into FILE1.ELT1, assemble the source statements and produce a standard listing (L option).

If the input source file contains one or more statement labels which begin with an '@', producing a statement with an '@' in column one, CROSSI can not be used for initial insertion of this source file. The user can insert the source file using the UNIVAC ELT processor with the I and D options (see 2.2.1).

Example 1:

```
@ELT,ID      FILE1.ELT2
source deck
@END
```

CROSSI can then read and assemble this file.

```
@CROSSI,L    FILE1.ELT2
@EOF
```

3.2.2 U Option

This option updates an existing source language input (SI) element to the next higher element cycle, thus saving any source language corrections that are currently being applied to the source language input element. If the source output field (second specification field) is specified, the U option is ignored and no cycling is produced. The updated source is written into the element specified in the source output field.

Example 1:

```
@CROSSI,UL   FILE1.ELT1
-5
  DC  2
-7,9
  DC  3
@EOF
```

Example 2:

```
@CROSSI,L    FILE1.ELT1,.ELT2
-5
  DC  2
-7,9
  DC  3
@EOF
```

In both examples the statement "DC 2" will be inserted following statement 5 of ELT1 in FILE1. Statement 7 thru 9 will be replaced by the statement "DC 3". The source statements will be assembled and a standard listing produced. In Example 1 the new cycle will be in ELT1 of FILE1. In Example 2 the updated element will be placed in ELT2 of FILE1 and ELT1 will be unchanged.

As in the case of the I option, CROSSI cannot be used to insert a correction line which has an '@' in column one. This insertion can be accomplished using the UNIVAC ELT processor with the U and D options.

Example 1:

```
@ELT,UD    FILE1.ELT1
-5
@LABEL  DC  2
@END
```

Followed by

```
@CROSSI,L  FILE1.ELT1
@EOF
```

3.2.3 List Options

These options specify the type of listing to be produced. CROSSI's listing attempts to reproduce the standard OS assembly listing. Statements that do not appear in the OS assembly listing, for example OPT and TITLE statements, do not appear in CROSSI's listing. CROSSI will number each statement according to its position in the file rather than its position in the listing. This will permit the user to employ the line numbers from the listing when editing or updating his files.

In almost all cases, CROSSI will flag statements containing errors with the same flags as the OS assembler (see Appendix A).

CROSSI will list the symbol table in alphabetic order. The order of this sort may not be identical to the alphabetic order produced by the OS assembler. The collating sequence used in CROSSI is not the ASCII sequence, but the UNIVAC Fieldata sequence, which results in different placement of numerics and special characters.

CROSSI will permit the user to request on the call statement either a standard listing, a short listing, a listing of flagged statements, or nc listing at all. The options are as follows:

<u>L Option From Batch Or B Option From Batch Or Interactive Mode</u>: This list option will produce the most extensive listing.

1. Pages will be numbered.
2. Title cards will cause a skip to the next page.
3. The most recent title (if any) will be printed at the top of each page.
4. Comment statements will be listed.

5. All assembled statements will be listed with statement number, error flag (if any), value and mode of location, machine code and statement text.
6. Finally, error count, address of next available location and the symbol table will be listed.

L Option From Interactive Mode: In interactive mode this option will produce a listing similar to the batch mode listing with the following exceptions:

1. Pages will not be numbered.
2. Titles will appear only where they occur.

S Option: This list option will produce a short listing which is useful in interactive mode

1. Comment statements will be listed.
2. All assembled statements will be listed with statement number, error flag, value and mode of location, and statement text. Machine code is not listed.
3. Finally, error count and address of next available location are listed. The symbol table is not listed.

N Option: This list option will produce no statement listing but will print error count and address of next available location.

No List Option: If no list option is requested CROSSI will produce a listing of all flagged statements, error count and address of next available location.

3.2.4 Punch Options

These options cause the production and output of a relocatable file and specify the output format. CROSSI will write the relocatable data and the necessary control items in a temporary file. After completion of the assembly and the designated listing, CROSSI will output this relocatable file to paper tape (R option) or to magnetic tape (M option).

R Option: This option will cause the output of relocatable data in zoned paper tape format (see Section 4.1) using ASCII printing characters. In order to punch this relocatable file, the user must be in interactive mode with a teletypewriter terminal equipped with a paper tape punch. After printing the designated listing, CROSSI will print the message:

"TURN ON PUNCH"

Within 20 seconds CROSSI will read and output the relocatable file.

Since the relocatable file is output as printing characters, the file will be both listed and punched. The R option in batch or from interactive terminals not equipped with a paper tape punch will list the relocatable file.

M Option: This option will output the relocatable file in packed format (see Section 4.3) to the magnetic tape file specified in the fourth specification field on the processor call statement. If no magnetic tape file is specified, no relocatable output is produced and CROSSI will print the message:

"M OPTION IGNORED
NO MAGNETIC TAPE FILE"

3.2.5 Other Standard UNIVAC Options (see UNIVAC UP-7824 or UP-4144)

G Option: compressed symbolic input cards.

H Option: input cards contain sequence numbers in column 73-80.

J Option: card input with compressed symbolic in columns 1-72 and sequence numbers in columns 73-80.

K Option: check sequence number in column 73-80. The H option must be used with the K option.

P Option: specifies that source language output should be in Fieldata code. Identifies card image input, if any, as being Fieldata.

Q Option: specifies that source language output should be ASCII. Identifies card image input, if any, as being ASCII.

4. RELOCATABLE OUTPUT

CROSSI records relocatable information in a temporary file with the FORTRAN unformatted WRITE statement. This file cannot be read by other UNIVAC processors. Upon completion of the assembly process, CROSSI reads and outputs this file one record at a time.

Each record contains 216 hex digits. The first item in each record is a four digit sequence number. Sequence numbers are negative integers (-1, -2, -3, etc.) represented in two's complement form. The next item is a four digit checksum which is the EXCLUSIVE OR sum of all hex digits in the record except itself plus a word of all ones.

4.1 Zoned Paper Tape Format

With the R option, CROSSI will punch paper tape in M08 zoned format which is the format used for punching paper tapes on a teletypewriter punch. CROSSI combines each hex digit of information with a zone of hex 3 or hex B depending on the parity bit, forming the following ASCII printing characters:

Hex Digit	ASCII Printing Character
0	@
1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	H
9	I
A	J
B	K
C	L
D	M
E	N
F	O

One record, including zones, is punched in 216 frames on paper tape. The last printing character in the last record of CROSSI's relocatable file is an ASCII period (X'2E'), which is not included in the checksum.

4.2 Pre-processor For Zoned Paper Tapes

Standard Interdata M08 formatted paper tapes use zones 0 or 1 to produce ASCII non-printing characters. It is not possible for CROSSI to transmit some of these non-printing ASCII characters. For example, an ASCII X'04' is an end-of-transmission signal (EOT) and the UNIVAC operating system will interpret it as a message stop code.

Interdata loaders insist that the standard zones be on M08 formatted paper tapes. CROSSI's paper tapes must be processed by an Interdata computer to produce a relocatable file that the Interdata loaders can read.

A pre-processor has been written which runs on the Model 70, Model 74 or Model 7/16. The listing of this program is given in Appendix B.

The pre-processor will:

1. Read each record of the CROSSI's relocatable paper tape file from logical unit one.

2. Ignore the ASCII characters: SP, CR, LF, NUL, DEL.

3. Strip parity bit and zone off each hex digit.

4. Pack each hex digit, four digits per word.

5. Output the packed hex digits with a binary write to logical unit two.

6. Continue reading, processing, and outputting records until an ASCII period (X'2E') is read.

7. Output last record and return control to the operating system.

The format of the output file will depend on the output device:

1. M08 zoned format for teletypewriter punch.

2. M16 packed format for high speed paper tape punch or mass storage devices.

4.3 Packed Format

CROSSI with the M option will output relocatable data to magnetic tape in packed M16 format. This format is used to punch paper tape on a high speed punch. The first character of each record is a X'F0'. Following this character each record contains 216 hex digits of information packed 2 hex digits per byte. The complete record is 109 frames.

A paper tape of this file can be punched offline. An Interdata equipped with a magnetic tape drive or a high speed paper tape reader can load these relocatable files directly. No pre-processing is required.

5. STRUCTURE OF CROSSI PROCESSOR

CROSSI's main program and eight of its fifteen subroutines are written in UNIVAC FORTRAN V. Seven subroutines are written in UNIVAC assembly language. Three of the FORTRAN subroutines are adapted from subroutines from MINIMAX by H. H. Grote ¼3½.

For file manipulation CROSSI uses a FORTRAN interface to the standard UNIVAC Processor Support Library. This interface was written at NBS by Edward Barkmeyer and allows reading and writing on mass storage files in 80R1 format.

UNIVAC internal code, Fieldata, is a six bit character code. One computer word of 36 bits holds six Fieldata characters. Each character position from left to right in a computer word is referred to as word position S1, S2, S3, S4, S5, and S6. UNIVAC numbers its bit positions from low order bit 0 to high order bit 35 (see Figure 5.1).

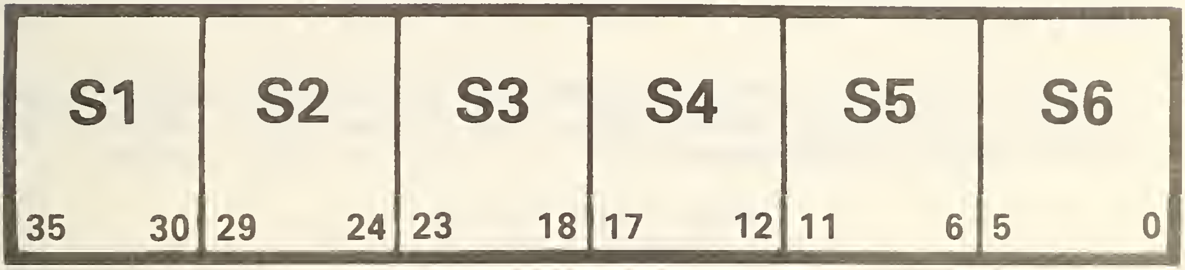

Figure 5.1

5.1 Temporary Files

CROSSI will assign a temporary source scratch file and a temporary relocatable file (R or M option only). These files will be FORTRAN logical units which are not already being used in the run. CROSSI will not interfere with any user assigned logical unit.

5.2 Symbol Tables

The symbol tables consist of two arrays: a symbolic table array (STBL) and a corresponding symbol definition table array (DTBL).

The symbol table (STBL) contains the Fieldata character representation of each symbol, left-justified and space-filled.

The symbol definition table (DTBL) contains:

1. Word position S3, S4, S5, S6 (bits 23 to 0) - the value or location of the symbol in hexadecimal Fieldata character representation.

2. Word position S2 (bits 29 to 24) - the mode of the symbol definition, either a Fieldata R for relative or a Fieldata space for absolute.

3. Word position S1 (bits 35 to 30) - any flag associated with that symbol in Fieldata character representation.

Example: X for external symbol
* for entry symbol
U for undefined symbol
M for multiplied defined symbol

If no flag is associated with the symbol, the S1 position contains a Fieldata space.

The size of the symbol table is set at compilation time by setting parameter LSCT.

It is not possible to continue assembly if an attempt is made to insert a label into a full symbol table. In this case CROSSI will halt the assembly and print the message:

"SYMBOL TABLE OVERFLOW
ASSEMBLY HALTED"

5.3 Operation Code Tables

There are two operation code tables: mnemonic operation code table (OPTBL) and a corresponding hex machine operation code table (HTBL).

5.3.1 Mnemonic Operation Code Table (OPTBL)

This table (OPTBL) contains mnemonic operation codes consisting of one to five characters, left-justified, blank-filled, starting at bit position 29 (word positions: S2, S3, S4, S5, S6). All operation codes including instruction set, extended branch mnemonic instructions and assembler directives are included in this table.

MODEL 3	MODEL 4	MODEL 74 MODEL 7-16A	MODEL 7-16B	MODEL 70 MODEL 7-16C	MODEL 70/xxx
BIT 35	BIT 34	BIT 33	BIT 32	BIT 31	BIT 30

Figure 5.2

The six bits of the S1 word position of the OPTBL table codes the legal computer model or models for the operation. Figure 5.2 shows which bits code the legal operation state for the various models. In the case of the standard set of instructions which are legal on all models, these six bits are all zeros. Otherwise, a one in a bit position indicates the legal state for the operation on that model.

Example:

6H@BAL	legal on all models
6GHSVC	legal on all models except 3 or 4
6H7DHR	legal on all models except 74 and 7/16A
6HBABL	legal on 7/16B, 70, 7/16C and the extended models

5.3.2 Hexadecimal Operation Code Table (HTBL)

This table (HTBL) contains four items of information per entry for the corresponding operation listed in table OPTBL.

1. Word position S5 and S6 (bit 11-0) contains the hexadecimal machine code for the operation as two Fieldata characters.

2. Word position S4 contains the value of the mask for the R1 field of extended branch mnemonics instruction as a Fieldata digit. For all other operations this position contains a Fieldata space.

3. Word position S2 contains a Fieldata digit from 1 to 3 or space which indicates hardware restrictions on the register fields as shown below.

    a. Fieldata 1 for short (16 bit) instructions: both register fields must have even operands.
    Examples: floating point register to register instructions.
    b. Fieldata 1 for long (32 bit) instructions: the R1 field must be even.
    Example: floating point double shift, multiply and divide long instructions.
    c. Fieldata 2 for short (16 bit) instructions: the R1 field must be even.
    Example: Register to register multiply and divide instructions.
    d. Fieldata 3 for instructions whose R1 or R2 fields are limited to R14 or lower, or R13 or lower, because the R+1 register or both R+1 and R+2 registers are used during the execution of the instruction.
    Examples: BXLE, BXH, WBR, RBR.

e. Fieldata space for all instructions with no hardware limitations on the register fields.

4. Word position S1 contains a binary digit which is used by the program to direct the analysis of the instruction through the assembly process (see Section 6.1).

5.4 Flow Through Processor

The following is an example of the order of events during CROSSI's assembly of an Interdata file assuming both the list option (L) and the punch option (R) are requested by the processor call statement.

5.4.1 Initiatization Phase

1. Assigns a temporary source file for source output from pass 1.

2. Reads processor call statement.

3. Processes options and determines list and punch mode.

4. Determines source input file, source output file (if any), and the target model.

5. Obtains date, time and cycle information and prints sign-on message.

6. Sets location counter value to X'0000' and mode to relative.

7. Assigns a temporary relocatable file.

8. Turns on automatic assembly indicator.

9. Branches to pass 1.

5.4.2 Pass 1 Phase

1. Reads each data image in 80 R1 format.

2. Outputs comment statements to source scratch file.

3. Parses instruction statements into 3 fields: label field, operation field, and operand and comment field.

4. Checks automatic assembly indicator and if on, continues processing. If off and the statement operation is not an IF or END, the statement is not evaluated, is not passed

on to pass 2 and will not appear in the listing or the relocatable file.

5. Adds label, label location and mode of location to the symbol tables.

6. Checks for and flags duplicate symbols.

7. Determines index of location of the operation in the operation code table.

8. Evaluates operands of DO statements and processes the statement following the DO statement the number of times indicated by the value of the DO operand.

9. Evaluates operand of IF statements and turns the automatic assembler indicator off if the operand is zero, and on if the operand is nonzero.

10. Evaluates operand of ORG statements, sets the location counter to the value of the operand and sets the location counter mode to mode of the operand.

11. Evaluates operand of EQU statements and adds the value and mode of operand to the symbol definition table corresponding to the statement label entry in the symbol table.

12. Evaluates operand of DS statement and increments the location counter by the value of the operand.

13. Evaluates length of DC C&...& statements.

14. Adds to the symbol tables: the operand symbols of an EXTRN statement, the value of absolute zero and an appropriate flag.

15. Increments location counter by 0, 1, 2, 4 or more bytes as indicated by the instruction.

16. Searches operand of an OPT statement for a program label and if found adds the appropriate loader control item and symbolic label to the relocatable buffer.

17. Forces the location counter to be even if
    a. operand of a DS instruction is odd (LC = LC+1),
    b. operand of an ORG instruction is odd (LC = LC-1), or
    c. location counter is odd after one or more DB instructions (LC = LC+1).

18. Outputs to the source scratch file, the following fields for each instruction statement:

    statement numbers
    error flag or space
    value of location counter
    mode of location counter
    label
    operation
    operand and comments
    sequence number
    index to position of operation in tables (OPTBL, HTBL)

19. Writes an EOF and rewinds the scratch file after encountering an END statement or an end of file.

5.4.3 Pass 2 Phase

1. Reads each record from the source scratch file.

2. Prints comment statements.

3. Parses other records into its various fields.

4. Searches symbol table for label duplication (labeled statements only) and flags the record if duplication is found.

5. Determines the hex machine operation code by a table look-up.

6. Evaluates operands of machine instructions and Define Byte and Define Constant assembly instructions. If one extra byte location was generated during Pass 1 to realign the location counter after Define Byte instructions, one zero byte is generated for that location.

7. Determines the appropriate loader control item for each machine instruction and for each Define Byte, Define Constant, Define Storage, and ORG assembly instruction.

8. Adds each loader control item and each data or address item to the relocatable buffer. That is, adds each item to checksum, converts each item to ASCII printing characters and adds these to the relocatable buffer.

9. Monitors the relocatable buffer and when buffer is full, adds record sequence number and final checksum to the buffer and outputs the buffer to the relocatable file.

10. Processes operands of ENTRY statements and flags each entry label in the symbol table.

11. Checks operands of EXTRN statements and flags any external symbols that become defined.

12. Processes title from TITLE statement and prints title at top of next page.

13. Prints page number and most recent title at top of each page.

14. Prints each record as follows:

    statement number
    error flag or space
    value of location counter
    mode of location counter
    machine code
    label
    operation
    operand
    comments

15. Evaluates operand (if any), remembers value of operand, and branches to the final phase when an END statement is encountered.

5.4.4 Final Phase

1. Prints symbol table in alphabetic order.

2. Adds symbolic external and entry symbols and their definitions along with the appropriate control item to the relocatable buffer.

3. Adds transfer address (if any) to the relocatable buffer.

4. Adds end of program control item to relocatable buffer.

5. Outputs last relocatable buffer to the relocatable file.

6. Writes an end of file and rewinds the relocatable file.

7. Prints the following message:

   "TURN ON PUNCH"

8. Pauses for 20 seconds.

9. Reads and outputs the entire relocatable file as 108 ASCII character records.

10. Frees temporary files.

11. Stops.

6. RECONFIGURATION OF CROSSI ASSEMBLER

CROSSI can be re-configured to change the length of the symbol tables or to assemble new operations for newer computer models which have upward compatability with the Model 70.

6.1 Addition Of New Operation Codes

CROSSI will accept new mnemonic operation code of up to five characters long whose operands conform to one of the following standard formats.

1. Short instructions with two operands.
   Example:
   ```
   SIS R1,2
   AHR R1,R2
   SLLS R1,4
   ```

2. Short instructions with one operand.
   Example:
   ```
   BR  R1
   BZR R2
   ```

3. Long instructions with two operands.
   Example:
   ```
   LH  R1,0(R2)
   SHI R1,X'1234'
   SLHL R1,LABEL
   ```

4. Long instructions with one operand.
   Example:
   ```
   BZ    LABEL1
   LPSW  LABEL2
   SINT  0(R1)
   AL    6(R2)
   ```

To insert a new operation code:

1. Increment the parameter LOCT by one.

2. Insert into table OPTBL at its correct alphabetic position, a word containing the following information:

   Position S2,S3,S4,S5,S6 - the Fieldata representation of the operation code, left-justified and blank-filled.

   Position S1 - Fieldata [(binary 1).

```
Example:     6H[RSTUV
             4H[XYZ
```

3. Insert into table HTBL, at the location corresponding to the new entry in OPTBL, a word containing the following information:

   Position S1 - A binary number, the value of which depends on the type of operand.

14	(Fieldata I) for short instructions with one operand.
12	(Fieldata G) for short instructions with two operands.
11	(Fieldata F) for long instructions with one operand.
13	(Fieldata H) for long instructions with two operands.

   Position S2 - A Fieldata space or appropriate register restriction (see Section 5.3.2).

   Postion S3 - A Fieldata space.

   Position S4 - A Fieldata space or the appropriate mask value (see Section 5.3.2).

   Position S5 & S6 - Hex machine code as two Fieldata characters.

```
Examples:   6HI   12
            6HF  042
            6HI1  56
```

In order to assemble the new operation, the third specification field on the call statement is coded as 70/xxx where xxx can be letters or digits which represent the new model. CROSSI will then assemble as legal operations, the model 70 instruction set plus all new instructions.

A single compilation of CROSSI can not distinguish between more than one additional model. No attempt is made to analyze the "xxx" subfield.

6.2 Length Of Symbol Table

The standard edition of CROSSI is configured with a symbol table for up to 1000 entries. This value can be changed by changing parameter LSCT from 1000 to the number of symbol entries desired.

REFERENCES

1. Interdata Model 70 User's Manual, Interdata Publication 20-261

2. UNIVAC 1100 Series Operating System, UNIVAC Publication UP-4144 Rev.4

3. MINIMAX - A FORTRAN Computer Program to Assemble the Machine Code for the Interdata Model 70 Minicomputer on the Control - Data 3800, Grote, Heinz H., NOAA Technical Report ERL 269 - APCL 28.

APPENDIX A

1. STATEMENT FLAGS

Flag	Meaning
Blank	Correct assembly
E	Illegal register assignment
F	Format error, assembly attempted
G	Severe format error, assembly not attempted, statement listed as received
M	Multiple defined symbol
O	Operation mnemonic invalid
T	Truncation error: a constant or expression has overflowed the specified limits
R	Relocation error, a meaningless combination of relocatable symbols as in an expression
U	Undefined symbol

2. SYMBOL TABLE FLAGS

Flag	Meaning
U Space	Undefined local symbol
* Space	Properly used EXTRN or ENTRY symbol
**	Symbol used in the operand of EXTRN or ENTRY not used properly or at all within the program
*,	An ENTRY symbol which was never defined was referenced
*.	An EXTRN symbol became defined
D*	Symbol used in the operand of both an EXTRN and an ENTRY line (written on object program as an ENTRY, if defined; or as an EXTRN, if referenced)
*M	Any EXTRN or ENTRY that became multiply defined

APPENDIX B

The following is a listing of the Interdata pre-processor which produces Interdata relocatable files from CROSSI's zoned ASCII paper tapes. It is also an example of a listing produced by CROSSI.

INTERDATA PREPROCESSOR FOR ASCII RELOCATABLE FILES

```
             * PRE-PROCESSOR FOR INTERDATA ASCII RELOCATABLE
             * PAPER TAPE FILES PRODUCED BY CROSSI WITH R OPTION.
             * ASCII RELOCATABLE FILE READ FROM LU 1.
             * INTERDATA RELOCATABLE FILE WRITTEN TO LU 2.
000A           A       EQU    10
000B           B       EQU    11
000C           C       EQU    12
000D           D       EQU    13
0006           PCNT    EQU    6
4801           RACR    EQU    X'4801'
3802           WBNR    EQU    X'3802'
0000R          INBUF   DS     216
00D8R          OUTBUF  DS     108
0144R 4801     RASCII  DC     RACR            READ ASCII FROM LU 1
0146R 0000             DC     0
0148R 0000R            DC     A(INBUF)
014AR 00D7R            DC     A(INBUF+215)
014CR 3802     WRBIN   DC     WBNR            WRITE BINARY ON LU 2
014ER 0000             DC     0
0150R 00D8R            DC     A(OUTBUF)
0152R 0143R            DC     A(OUTBUF+107)
0154R 0006     UPK     DC     6
0156R 0166R            DC     A(ERR)
0158R 0007     MSG     DC     7
015AR 000E             DC     14
015CR 492F             DC     C'I/O ERROR '
      4F20
      4552
      524F
      5220
0166R          ERR     DS     4
016AR 0007     PMSG    DC     7
016CR 0018             DC     24
016ER 4153             DC     C'ASSEMBLER PRE-PROCESSOR '
      5345
      4D42
      4C45
      5220
      5052
      452D
      5052
      4F43
      4553
      534F
      5220
0186R E120    START   SVC    2,PMSG           SIGN ON MESSAGE
      016AR
             *
             * INITAILIZE REGISTERS FOR BXLE CONTROL
             *
018AR 2482            LIS    8,2              INC FOR OUTBUF
018CR C890            LHI    9,OUTBUF+107     LAST BYTE IN OUTBUF
      0143R
0190R 24B1            LIS    B,1              INC FOR INBUF
0192R C8C0            LHI    C,INBUF+215      LAST BYTE IN INBUF
      00D7R
0196R C870    START1  LHI    7,OUTBUF         ADDRESS OF OUTBUF
      00D8R
019AR 0700            XHR    0,0
019CR 4007    START2  STH    0,0(7)           ZERO OUTBUF
      0000
01A0R C170            BXLE   7,START2
      019CR
01A4R C870            LHI    7,OUTBUF
      00D8R
01A8R 2464            LIS    PCNT,4           INIT COUNTER TO 4
01AAR 0722            XHR    2,2
01ACR E110    READ    SVC    1,RASCII         READ FIRST RECORD
      0144R
01B0R 4800            LH     0,RASCII+2       WAS RECORD GOOD?
      0146R
01B4R 4230            BNZ    ERROR            NO,PRINT MESSAGE,STOP
      0212R
01B8R C8A0            LHI    A,INBUF          ADDRESS OF INBUF
      0000R
01BCR D33A    PACK    LB     3,0(A)           FETCH ONE CHAR
      0000
01C0R C530            CLHI   3,X'0A'          IS CHAR A LINE FEED?
      000A
01C4R 4330            BE     PACK1            YES, IGNORE
      01ECR
01C8R C530            CLHI   3,X'20'          IS CHAR A SPACE?
      0020
01CCR 4330            BE     PACK1            YES, IGNORE
      01ECR
01D0R C530            CLHI   3,X'0D'          IS CHAR A CR?
      000D
01D4R 4330            BE     READ             READ NEXT RECORD
      01ACR
01D8R C530            CLHI   3,X'2E'          IS CHAR A PERIOD?
      002E
01DCR 4330            BE     DONE             YES, END OF FILE
      021ER
01E0R 913C            SLLS   3,12             PACK 4 BITS OF CHAR
```

```
01E2R ED20          SLL   2,4
      0004
01E6R 2761          SIS   PCNT,1            DECREMENT COUNTER
01E8R 4330          BZ    PACK2             IS OUTPUT WORD FULL
      01F4R
01ECR C1A0  PACK1   BXLE  A,PACK            INC INPUT POINTER
      01BCR
01F0R 4300          B     READ              READ NEXT RECORD
      01ACR
01F4R 4027  PACK2   STH   2,0(7)            STORE PACKED BINARY DATA
      0000
01F8R 0722          XHR   2,2
01FAR C860          LHI   PCNT,4            REINITIALIZE COUNTER
      0004
01FER C170          BXLE  7,PACK1           INC OUTPUT POINTER
      01ECR
0202R E110  WRITE   SVC   1,WRBIN           WRITE FULL OUTBUF
      014CR
0206R 4800          LH    0,WRBIN+2         IS WRITE GOOD?
      014ER
020AR 4230          BNZ   ERROR             NO,PRINT MESSAGE,STOP
      0212R
020ER 4300          B     START1            YES, CONT
      0196R
0212R E120  ERROR   SVC   2,UPK             ERROR CODE
      0154R
0216R E120          SVC   2,MSG             PRINT ERROR MESSAGE
      0158R
021AR E130          SVC   3,0               RETURN CONTROL TO OS
      0000
021ER 0822  DONE    LHR   2,2               PACK LAST DATA
0220R 4330          BZ    DONE1             IS OUTPUT WORD EMPTY?
      0246R
0224R C320          THI   2,X'FFF0'         WORD CONTAIN 1 HEX DIGET?
      FFF0
0228R 4230          BNZ   DONE2             NO, CONT
      0230R
022CR 912C          SLLS  2,12              YES, SHIFT LEFT
022ER 230C          BS    DONE1             PACKING DONE
0230R C320  DONE2   THI   2,X'FF00'         WORD CONTAIN 2 HEX BITS?
      FF00
0234R 4230          BNZ   DONE3             NO, CONT
      023CR
0238R 9128          SLLS  2,8               YES, SHIFT LEFT
023AR 2306          BS    DONE1             PACKING DONE
023CR C320  DONE3   THI   2,X'F000'         WORD CONTAIN 3 HEX DIGITS?
      F000
0240R 4230          BNZ   DONE1             NO, WORD HAS 4 HEX DIGITS
      0246R
0244R 9124          SLLS  2,4               YES, SHIFT LEFT
0246R 4027  DONE1   STH   2,0(7)            STORE WORD IN OUTBUF
      0000
```

```
   024AR E110            SVC   1,WRBIN         WRITE LAST RECORD
         014CR
   024ER 4800            LH    0,WRBIN+2       IS LAST RECORD GOOD?
         014ER
   0252R 4230            BNZ   ERROR           NO,PRINT MESSAGE,STOP
         0212R
   0256R E130            SVC   3,0             YES, ALL DONE
         0000
   025AR                 END   START
NO ERRORS
  025AR               END  START
  A       000A
  B       000B
  C       000C
  D       000D
  DONE    021ER
  DONE1   0246R
  DONE2   0230R
  DONE3   023CR
  ERR     0166R
  ERROR   0212R
  INBUF   0000R
  MSG     0158R
  OUTBUF  00D8R
  PACK    01BCR
  PACK1   01ECR
  PACK2   01F4R
  PCNT    0006
  PMSG    016AR
  RACR    4801
  RASCII  0144R
  READ    01ACR
  START   0186R
  START1  0196R
  START2  019CR
  UPK     0154R
  WBNR    3802
  WRBIN   014CR
  WRITE   0202R
END CROSSI
```

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR77-1230 (R)	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

CROSSI - A UNIVAC PROCESSOR WHICH ASSEMBLES
CODE FOR INTERDATA MINICOMPUTERS

5. Publication Date

6. Performing Organization Code

7. AUTHOR(S)

Carol V. Young

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. Sponsoring Organization Name and Complete Address *(Street, City, State, ZIP)*

13. Type of Report & Period Covered

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

Instructions are given for the use of CROSSI, a computer program which assembles Interdata assembler code on the UNIVAC 1108 producing an assembler listing and relocatable code for the minicomputer. The assembler statements, relocatable output formats and the structure of the processor are described. A preprocessor is given for converting CROSSI's zoned relocatable format to Interdata's zoned relocatable format. A method of reconfiguring the CROSSI processor to assemble code for newer upward compatible minicomputer models is explained.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Assembler; cross-assembler; machine code; relocatable code

18. AVAILABILITY ☒ Unlimited

□ For Official Distribution. Do Not Release to NTIS

□ Order From Sup. of Doc., U.S. Government Printing Office
Washington, D.C. 20402, SD Cat. No. C13

☒ Order From National Technical Information Service (NTIS)
Springfield, Virginia 22151

19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED	21. NO. OF PAGES 32
20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	22. Price $4.00