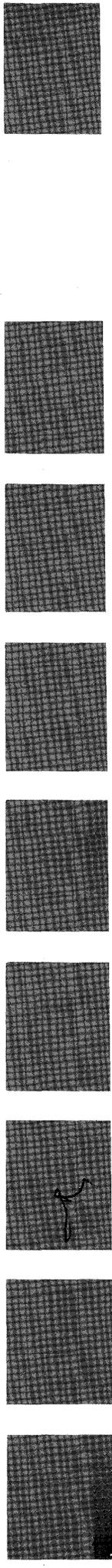




Systems Reference Library

IBM 1130 Card/Paper Tape Programming System Operators Guide

This publication provides the information necessary to operate the IBM 1130 Card and Paper Tape Programming Systems: FORTRAN Compiler, Assembler and Compressor, Subroutine Library, and Utility Programs. Primarily intended for the machine operator, the manual describes the loading and restart procedures for the programming systems in detail.



PREFACE

The 1130 Card/Paper Tape Programming System Operator's Guide is divided into four sections:

1. Assembler and Compressor
2. FORTRAN
3. Subroutine Library
4. Utility

Each section gives a brief description of the programming system and describes the loading procedures in detail. In addition, the manual provides information on card formats, user options, error waits, and program restarts.

Throughout this publication all references to locations in storage are in hexadecimal; therefore, the subscript 16 has been omitted.

The reader should be familiar with the following publications:

IBM 1130 Functional Characteristics
(Form A26-5881)

IBM 1130 Computing System Input/Output Units
(Form A26-5890)

IBM 1130 Subroutine Library
(Form C26-5929)

IBM 1130 FORTRAN Language
(Form C26-5933)

IBM 1130 Assembler Language
(Form C26-5927)

Machine Requirements

The minimum machine configuration required for operation of the programming systems are:

1. IBM 1131 CPU Model 1 with a minimum of 4096 words of core storage
2. IBM 1442 Card Read Punch, or IBM 1134 Paper Tape Reader and IBM 1055 Paper Tape Punch.

This publication supersedes and makes obsolete the publication, IBM 1130 Utility Routines (Form C26-5931).

Copies of this and other IBM publications can be obtained through IBM Branch Offices. A form has been provided at the back of this publication for readers' comments. If the form has been detached, comments may be directed to: IBM, Programming Publications Dept. 234, San Jose, Calif. 95114

CONTENTS

ASSEMBLER AND COMPRESSOR PROGRAMS	1	SUBROUTINE LIBRARY	17
Card Assembler Programs	2	ISS Pre-Operative Errors	17
Pass 1 Procedure	2	Card Subroutine (CARD0 and CARD1) Errors	17
Pass 2 Procedure	2	Console Printer Subroutine (TYPE0 and WRTY0) Errors	19
Assembler Card Formats	3	Keyboard Subroutine (TYPE0) Functions	19
Card Compressor Program	4	Paper Tape Subroutines (PAPT)	19
Procedure	4	Adding and Removing Subroutines	19
Compressed Deck Format	4	1130 Subroutine Library Listing	20
Card System Options	4	UTILITY ROUTINES	26
Uninterrupted Assembly	4	Loading Routines	26
Consecutive Assembly	4	Card System Relocating Loader	26
Error Detection	4	Card System Core Image Converter	28
PACH	4.1	Card System Core Image Loader	31
Compression of Oversize Decks	4.1	Paper Tape System Relocating Loader	31
Card System Waits and Error Conditions for Assembler and Compressor	5	Paper Tape System Core Image Converter	31.1
Restart Procedures	5	Paper Tape System Core Image Loader	31.2
Error Detection Codes	6	Utility Input/Output Routine	31.2
Paper Tape Assembler Program	6	I/O Character and Record Format	31.2
Assembly Procedures	6	Control Card Format	32
Assembler Tape Format	7	User's Exit Option--Card and Paper Tape	32
Paper Tape Compressor Program	7	I/O Routine Error Checks	34
Compressor Procedure	7	Operating Instructions -- Card and Paper Tape	34
Compressed Tape Format	8	Dump Routines	34
Paper Tape Waits and Error Conditions for Assembler and Compressor	8	Keyboard Routine	36
Error Codes	8	Card Reproducing Routine	37
Error Conditions	8	Console Routine	37
FORTTRAN	9	Disk Pack Initialization Routine (DPIR)	38
Card System Compiler Loading	9	Construct Paper Tape Routine	39
Card Load Instructions	9	Operating Notes	41
Card Load Errors	9	APPENDIX A. CORE MAP	42
Control Record Options	10	APPENDIX B. CORE REQUIREMENTS AND PROGRAM TIMES	43
1800 System Control Record Options	12	APPENDIX C. SAMPLE PROGRAMS	45
Optional Tracing	12	APPENDIX D. 1130 CARD DECK AND PAPER TAPE IDENTIFICATION	49
FORTRAN Printouts	12	APPENDIX E. RELOCATABLE CARD AND PAPER TAPE FORMATS	51
Object Deck Loading Procedures	14	APPENDIX F. BINARY CARD	54
Card Load Instructions	14	APPENDIX G. ERROR WAITS AND PROGRAM WAITS	55
Keyboard Input of Data Records	15	APPENDIX H. PAPER TAPE READER ERROR (CHECKSUM) RESTART	57
Object Program Data Record Format	15	APPENDIX I. CONSOLE AND INTERRUPT RESTART PROCEDURES	59
FORTRAN I/O Errors	15	INDEX	64
Paper Tape System Compiler Loading	16		
Compiler Load Instructions	16		
Paper Tape Load Errors	16		
Source Record Format	16.1		
Paper Tape Object Program Loading Procedures	16.1		
Object Program Load Sequence	16.1		
Object Program Load Instructions	16.1		

The primary input to the 1130 computing system is from cards or paper tape. The handling of data

from either source is basically the same. A typical card processing system is shown in Figure 1.

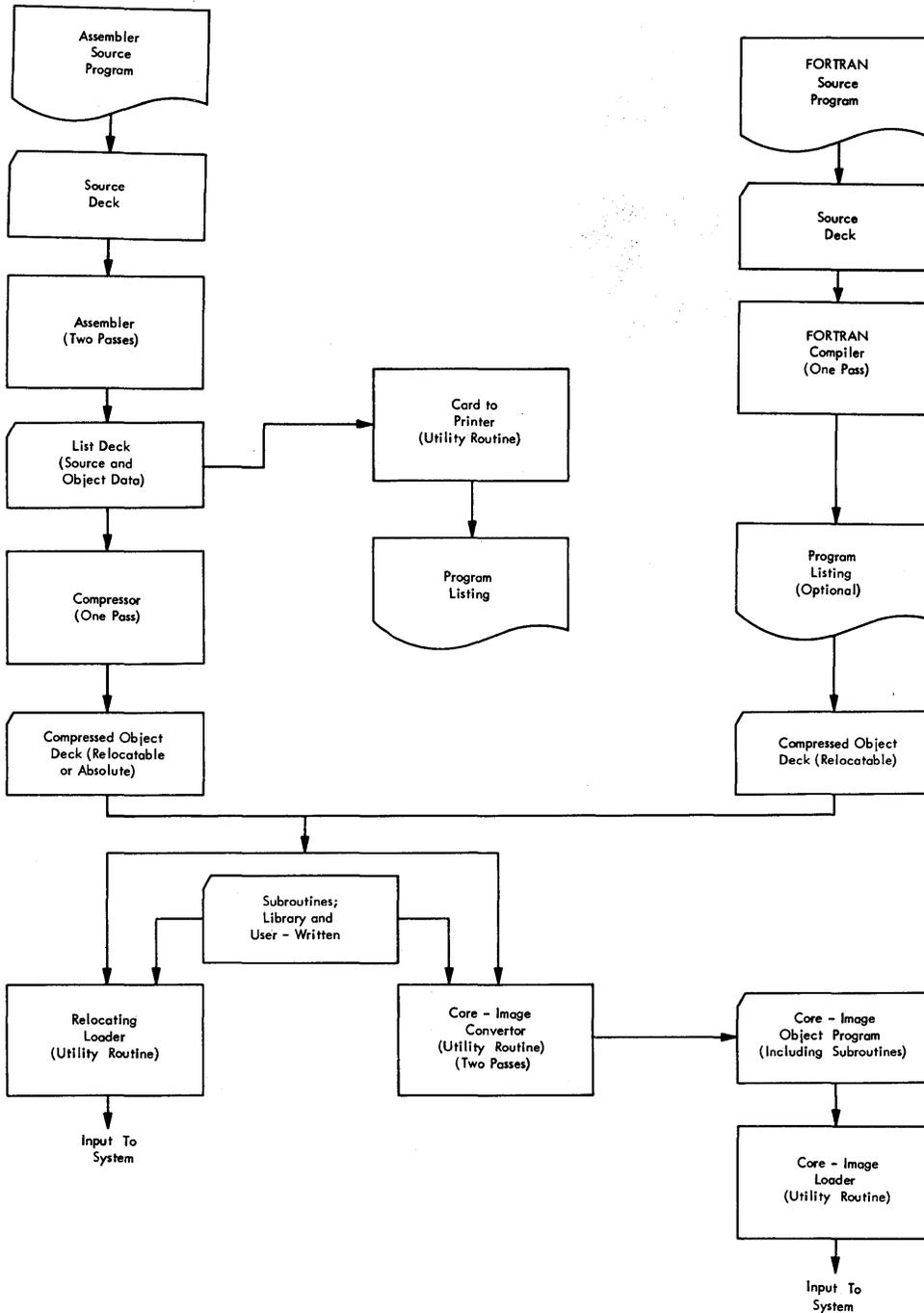


Figure 1. Typical Card Processing System

The IBM 1130 Assembler, available in both paper tape form and card form, is the medium by which a symbolic source program is converted (assembled) into a machine language program. Mainline programs and subroutines intended for use with the 1800 card/paper tape system can also be assembled with this assembler. The assembler is a two-pass program. In addition, after the second pass, a compressor program must compress the object program before it can be loaded into the CPU for execution. The operation (using the card system) is as follows:

- Pass 1 is initiated by loading the assembler deck, followed by the source program (see Figure 2). During pass 1, the assembler generates a symbol table for use in the second pass. The maximum size of the symbol table and, hence, the maximum number of symbols that can be defined in a program is determined by the size of core storage, thus:

<u>Size of Core Storage (words)</u>	<u>Maximum Number of Symbols in Table</u>
4096	550
8192	1915

- At the end of the first pass the assembler waits for the user to reload the source program for the second pass. During pass 2, object data in the form of hexadecimal digits and error codes is punched into the first 19 columns of each source card. The resulting deck is a list deck only; it cannot be loaded directly to core storage (see Figure 3). If the program will be reassembled later, either another copy of the source deck must be available or columns 21-80 of the list deck must be reproduced to make a new source deck (see Card System Options).
- At the end of the second pass, the list deck can be listed to determine if any errors were detected. If a loadable deck is to be made, the compressor program is loaded, followed by the

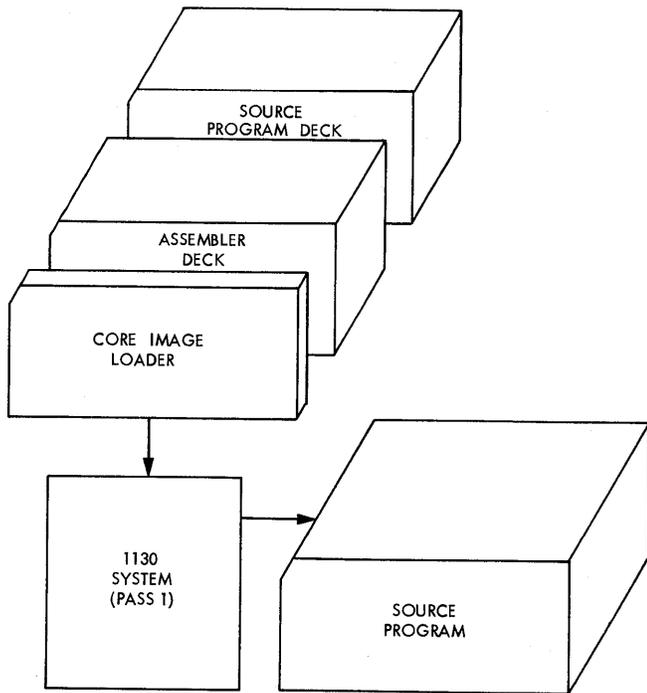


Figure 2. Card Assembler, Pass 1

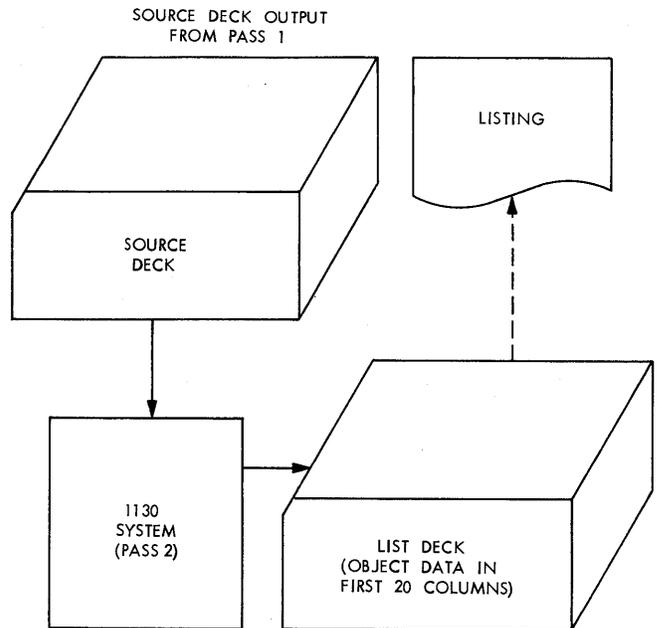


Figure 3. Card Assembler, Pass 2

list deck and a number of blank cards. The data from the list deck is read, converted to compressed binary form, and stored in core storage. After the list deck has passed through the card reader, the compressed data is punched into the blank cards that follow the list deck (see Figure 4).

- The resulting binary object program deck is acceptable as input to the relocating loader or to the core image converter.

CARD ASSEMBLER PROGRAMS

The assembly procedures for pass 1 and pass 2 are as follows:

PASS 1 PROCEDURE

1. Place cards in the 1442 Card Read Punch in the following order:
 - Core image loader
 - Assembler deck
 - Source program deck (including END card)
2. Set the console Mode switch to RUN and press the following keys:
 - IMM STOP and RESET on the console
 - START on the card reader
 - PROGRAM LOAD on the console

The loader feeds the assembler deck into the inside stacker (stacker 1) and the source program deck into the outside stacker (stacker 2).

3. Press reader START to process the last two cards (see Card System Options). The card system assembler checks card columns 1-20 on pass 1 and pass 2, immediately after reading a source card. At this time these columns should be blank. If a punch is detected the assembler waits with AAAA in the accumulator.
4. To continue the same assembly, remove the deck from the hopper and clear the reader with the NPRO key. Remove the two ejected cards from the stacker and replace the first card with a duplicate, except for columns 1-20 which must be blank. Place these two cards in proper sequence in front of the unprocessed portion of the deck. Reload this deck in the reader and press reader START. Press PROGRAM START on the console to continue assembly.
5. To start a new assembly, remove the source deck in process from the reader, and clear the reader with the NPRO key. Press RESET and PROGRAM START on the console. The reader is now ready to accept a new deck for assembly.

PASS 2 PROCEDURE

1. Remove the source program deck from stacker 2 and place it in the reader hopper. Press

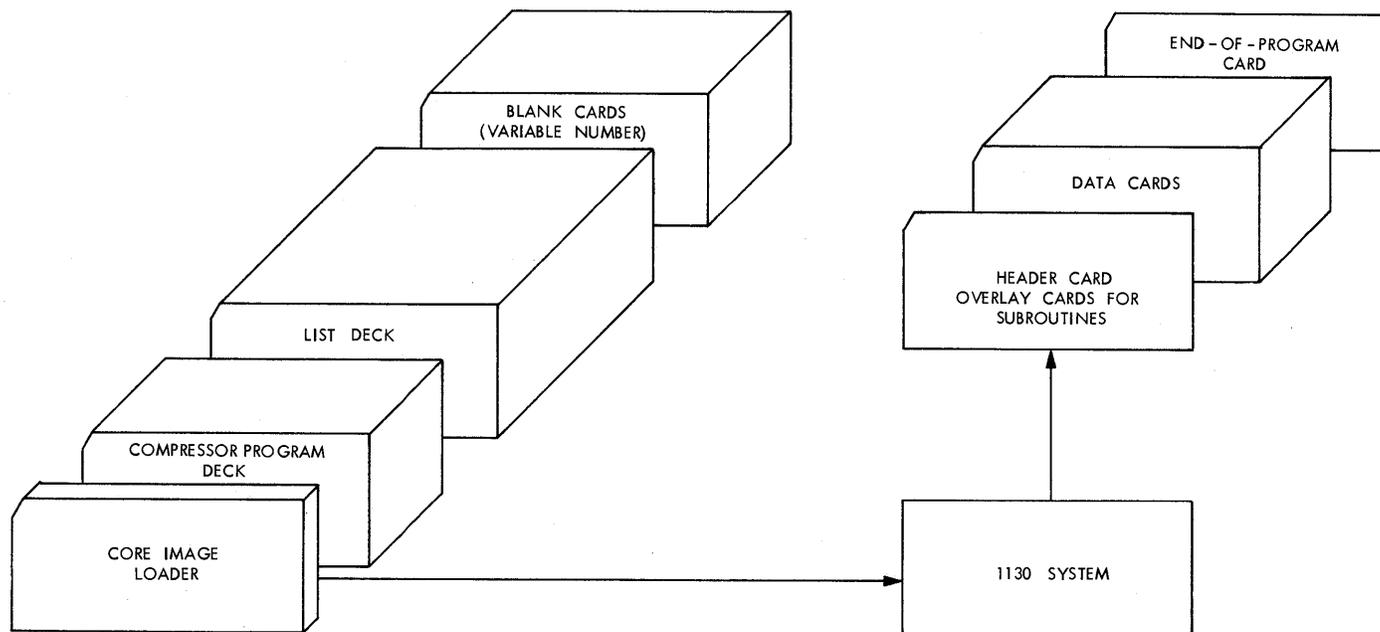


Figure 4. Compressor Program Operation

START on the reader. The assembler now completes assembly of the source program, selecting the cards into stacker 1.

2. Press START on the reader to process the last two cards. The output of pass 2 is a list deck. The list deck can be listed offline, or on-line through the use of a utility routine.

Columns 1-16	Hexadecimal data
18-30, 35-71	IBM card code or EBCDIC
32-33	Only the indicated characters should be used
73-77	ID field
78-80	Sequence number

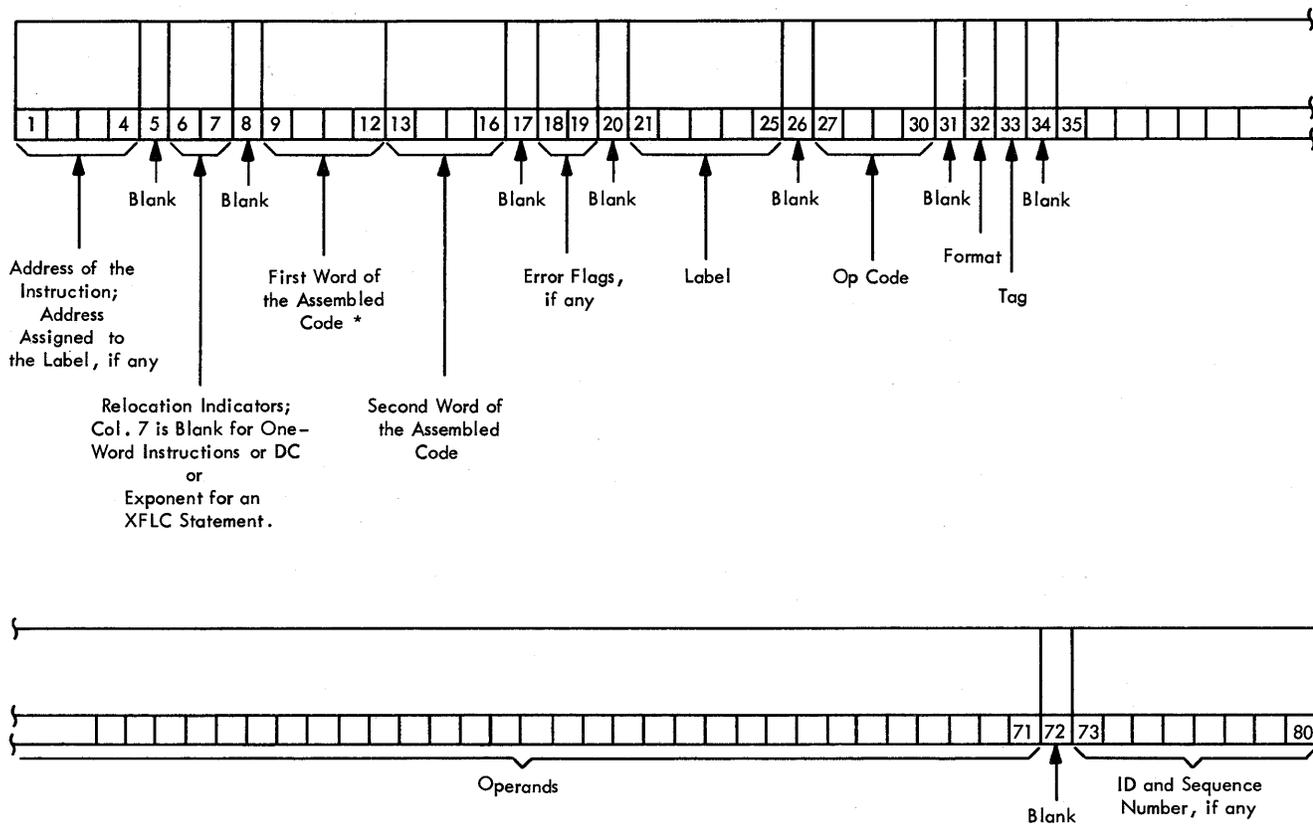
ASSEMBLER CARD FORMATS

The list deck card format is shown in Figure 5.

The IBM Symbolic Assembly Program Card

(Electro No. J76378) format is shown in Figure 6.

The EBCDIC special characters, illustrated on the card, are generated by an IBM 029 Card Punch and the IBM card code special characters by an IBM 026 Card Punch. The blocks across the 12 row of the card are for 80 column interpreters, and those on the 11 row are for 60 column interpreters.



* For EBC Statements, Col. 9-12 Contains the Number of EBC Characters

For BSS and BES Statements, Col. 9-12 Contains the Number of Words Reserved for the Block.

Figure 5. List Deck Format

IBM 1130-1800 SYMBOLIC ASSEMBLY PROGRAM CARD										OPERANDS AND REMARKS										ID. AND SEQ. NO.
ADDR.	REL.	WORD 1	WORD 2	ERR.	LABEL	OP CODE	FT	OPERANDS AND REMARKS										ID. AND SEQ. NO.		
0000	00	00000000	0 0			0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	000				
1A1A	1A	1A1A1A1A	1 1 1 1 1 1 1 1			1	1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	111				
2B2B	2B	2B2B2B2B	2 2 2 2 2 2 2 2			2	2	2 2 2 2 2 2 2 2	2 2 2 2 2 2 2 2	2 2 2 2 2 2 2 2	2 2 2 2 2 2 2 2	2 2 2 2 2 2 2 2	2 2 2 2 2 2 2 2	2 2 2 2 2 2 2 2	2 2 2 2 2 2 2 2	222				
3C3C	3C	3C3C3C3C	3 3 3 3 3 3 3 3			L3	3	3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3	333				
4D4D	4D	4D4D4D4D	4 4 4 4 4 4 4 4					4 4 4 4 4 4 4 4	4 4 4 4 4 4 4 4	4 4 4 4 4 4 4 4	4 4 4 4 4 4 4 4	4 4 4 4 4 4 4 4	4 4 4 4 4 4 4 4	4 4 4 4 4 4 4 4	4 4 4 4 4 4 4 4	444				
5E5E	5E	5E5E5E5E	5 5 5 5 5 5 5 5					5 5 5 5 5 5 5 5	5 5 5 5 5 5 5 5	5 5 5 5 5 5 5 5	5 5 5 5 5 5 5 5	5 5 5 5 5 5 5 5	5 5 5 5 5 5 5 5	5 5 5 5 5 5 5 5	5 5 5 5 5 5 5 5	555				
6F6F	6F	6F6F6F6F	6 6 6 6 6 6 6 6					6 6 6 6 6 6 6 6	6 6 6 6 6 6 6 6	6 6 6 6 6 6 6 6	6 6 6 6 6 6 6 6	6 6 6 6 6 6 6 6	6 6 6 6 6 6 6 6	6 6 6 6 6 6 6 6	6 6 6 6 6 6 6 6	666				
7777	77	77777777	7 7 7 7 7 7 7 7			X	7	7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7	777				
8888	88	88888888	8 8 8 8 8 8 8 8					8 8 8 8 8 8 8 8	8 8 8 8 8 8 8 8	8 8 8 8 8 8 8 8	8 8 8 8 8 8 8 8	8 8 8 8 8 8 8 8	8 8 8 8 8 8 8 8	8 8 8 8 8 8 8 8	8 8 8 8 8 8 8 8	888				
9999	99	99999999	9 9 9 9 9 9 9 9			I	9	9 9 9 9 9 9 9 9	9 9 9 9 9 9 9 9	9 9 9 9 9 9 9 9	9 9 9 9 9 9 9 9	9 9 9 9 9 9 9 9	9 9 9 9 9 9 9 9	9 9 9 9 9 9 9 9	9 9 9 9 9 9 9 9	999				

● Figure 6. Symbolic Assembly Card

CARD COMPRESSOR PROGRAM

PROCEDURE

The procedure for compression is as follows:

- Place cards in the reader in the following order:
 Core image loader
 Compressor program deck
 List deck from pass 2
 Blank cards
- Set the console Mode switch to RUN and press the following keys:
 IMM STOP and RESET on the console
 START on the card reader
 PROGRAM LOAD on the console
 The list deck is read in by the compressor program and a compressed binary object deck is punched into the blank cards and fed into stacker 2. The list deck and any excess blank cards are fed into stacker 1.

In the event that a card with an assembler error code is detected during compression, the system WAITS with 9696 in the accumulator.

If this error halt occurs, the user can terminate or continue the compression by pressing PROGRAM START (see PACH). The erroneous card is ignored. A new compression can begin after the END card is read.

COMPRESSED DECK FORMAT

Each card in the compressed deck contains up to 54 16-bit words, excluding the ID (identification) and

sequence fields. The cards and their contents are shown in Appendix E.

CARD SYSTEM OPTIONS

Card system options include the following:

UNINTERRUPTED ASSEMBLY

- The output from pass 1 of an assembly (in stacker 2) can be placed directly behind the END card to allow the assembler to immediately begin pass 2. The output of pass 2 is stored in stacker 1.

CONSECUTIVE ASSEMBLY

- At the end of pass 2, another program can begin assembly immediately. This allows the user to place the first card of the second program immediately after the END card of the first program when the first program is being read and punched in pass 2.

ERROR DETECTION

- If Console Entry switch 15 is ON before pass 2 begins, the only data punched will be error codes for erroneous statements. This allows a program to be checked for errors during an assembly without requiring the entire source deck to be

duplicated for the next assembly. If errors are detected with switch 15 on, only cards containing errors need be reproduced. Columns 18-19 must be blanked.

PACH

- If erroneous source cards were ignored during compression, the compressed object deck can be corrected by patching. The user can also alter an object deck by patching. The card compressor accepts the mnemonic op code PACH followed by corrected hexadecimal source record cards and produces compressed binary object patch cards with a zero checksum.

A patch deck is produced as follows:

1. Set the incomplete object deck aside.
2. Make a patch header card by punching PACH in columns 27-30 of a blank card.
3. Consult the assembler listing for source errors.
4. Punch corrected source cards: enter corrected data in columns 1-16 as indicated in Figure 5--

List Deck Format. The remaining card columns can be blank.

5. While under the control of the compressor program, place the PACH card, corrected source cards, and blank cards in the card reader.
6. Press reader START and PROGRAM START (PROGRAM LOAD if the compressor is being reloaded).
7. A compressed binary object patch deck is produced by the compressor.
8. Remove the END card from the incomplete object deck set aside in step 1 and place the patch deck behind the object deck.
9. Place the END card behind the patched object deck. The corrected binary object deck is acceptable as input to the relocating loader or to the core image converter.

COMPRESSION OF OVERSIZE DECKS

If the program being compressed is larger than the compressor can store before punching (see Card

System Waits and Error Conditions for Assembler and Compressor), the user may place a number of blank cards in the list deck at some point after the first several hundred cards of the list deck.

When these blank cards are sensed, the compressor punches out the portion of the binary object deck accumulated this far. Any remaining blank cards are passed on into stacker 1, and compression begins again when the next card in the list deck is sensed (this procedure may leave some blank cards in the list deck).

If not enough blank cards have been inserted in the list deck to hold the portion that has been read at that point, compression continues when list cards are again sensed; the unpunched portion of the list deck is punched when the next blank cards are sensed.

Note that decks may be compressed together. Batching is allowed with no blanks between decks.

CARD SYSTEM WAITS AND ERROR CONDITIONS FOR ASSEMBLER AND COMPRESSOR

RESTART PROCEDURES

In addition to the errors listed below, there are other WAITS described in Appendix G.

Symbol Table Overflow

- When the symbol table is full, the assembler halts with F0F0 in the accumulator. If this occurs, either of two corrective procedures can be used:
 1. Divide the program into segments and assemble each segment separately, or
 2. Reduce the number of symbols by using relative addressing. For example, the following sequence of instructions

Label	Operation	F	T
21	25	27	30
32	33	35	40
45			
B,E,G,I,N	L,D,		O,N,E
	S,T,O,		B,U,F,E,R,
	L,D,		H,I,F,O,
	A,		B,U,F,2,
	S,T,O,		B,U,F,3,
O,N,E	D,C,		1,
B,U,F,E,R	D,C,		0,
H,I,F,O	D,C,		/F,O,
B,U,F,2	D,C,		0,
B,U,F,3	D,C,		0,

can also be written

Label	Operation	F	T
21	25	27	30
32	33	35	40
45			
B,E,G,I,N	L,D,		B,G,C,O,N
	S,T,O,		B,G,C,O,N,+1,
	L,D,		B,G,C,O,N,+2,
	A,		B,G,C,O,N,+3,
	S,T,O,		B,G,C,O,N,+4,
B,G,C,O,N	D,C,		1,
	D,C,		0,
	D,C,		/F,O,
	D,C,		0,
	D,C,		0,

and eliminate four symbols.

Oversize Program

If the compressor encounters more list deck cards than it can handle before blank cards are sensed, the compressor waits with FFFF in the accumulator. The following procedure is used to restart.

1. Remove the unprocessed portion of the list deck.
2. Clear the reader with the NPRO key.
3. Place the two non-processed cards ahead of the remaining portion of the list deck.
4. Insert blank cards into the reader, followed by the remainder of the list deck.
5. Press reader START.

The compressor then punches binary cards for the part of the list deck that has been read and passes the remaining blank cards. When the first card of the unprocessed portion of the list deck is sensed, the compression continues. This process can be repeated as necessary to compress the entire program.

Read Error

If an error occurs during reading the reader stops in a not-ready status.

1. Remove the remainder of the deck from the hopper.
2. Press the NPRO key.
3. Place the two non-processed cards ahead of the unused portion of the deck.
4. Reload the deck in the hopper.
5. Press reader START.

Punch Error

If an error occurs during punching the reader stops in a not-ready status.

1. Remove the deck.
2. Press the NPRO key to clear the two cards remaining in the reader.
3. Duplicate the first card, except for columns 1-20 which must be blank, and discard the erroneously punched card.
4. Place the two cards ahead of the unused portion of the deck.
5. Reload the deck in the hopper.
6. Press reader START.

The Punch Error procedure applies to the assembler only. For the compressor:

1. Press NPRO.
2. Discard the mispunched card.
3. Insert more blank cards in the hopper.
4. Press reader START.

Errors occurring on the last card of a deck are not noted until after reader START is pressed.

ERROR DETECTION CODES

Error detection codes for the 1130 assembler are listed in Table 1. For the first error detected in each statement the assembler stores and then punches the code in column 18; the code for a second error is stored, overlaid by any subsequent errors, and punched in column 19. Thus, if more than two errors are detected in the same statement, only the first and last are indicated.

PAPER TAPE ASSEMBLER PROGRAM

The paper tape assembler is supplied to the user as self-loading ~~PTTC/8 (perforated tape transmission code/8 channel)~~ ^{binary} tape, approximately 40 feet in length. The beginning of the tape is a leader, several feet long, that contains the program ID, preceded and followed by delete code punching. Following this leader are several feet of tape, punched in the IPL (initial program load) mode (i. e., four bits per frame); the remainder of the tape is binary.

ASSEMBLY PROCEDURES

Loading Assembler Program

The assembler tape is loaded as follows:

Table 1. Assembler Error Detection Codes

Flag	Cause	Assembler Action
A	Address Error Attempt made to specify displacement field, directly or indirectly, outside range of -128 to +127.	Displacement set to zero
C	Condition Code Error Character other than +, -, Z, E, C, or O detected in first operand of short branch or second operand of long BSC, BOSC, or BSI statement.	Displacement set to zero
F	Format Code Error Character other than L, I, X, or blank detected in col. 32, or L or I format specified for instruction valid only in short form.	Instruction processed as if L format were specified, unless that instruction is valid only in short form, in which case it is processed as if the X format were specified
L	Label Error Invalid symbol detected in label field.	Label ignored
M	Multiply Defined Label Error Duplicate symbol encountered in label field or in operand.	First occurrence of symbol in label field defines its value; subsequent occurrences of symbol in label field are ignored. Multiply defined indicator inserted in symbol table entry (Bit 0 of first word).
R	Relocation Error Expression does not have valid relocation. Non-absolute displacement specified. Absolute origin specified in relocatable program. Non-absolute operand specified in BSS or BES. Non-relocatable operand in END statement of relocatable mainline program. ENT operand non-relocatable.	Expression set to zero Displacement set to zero Origin ignored Operand assumed to be zero Card columns 9-12 left blank; entry assumed to be relative zero Statement ignored
S	Syntax Error Invalid expression (e.g., invalid symbol, adjacent operators, illegal constant) Illegal character in record.	Expression set to zero If illegal character appears in expression, label, op code, format, or tag field, additional errors may be caused. Card columns 9-12 left blank; entry assumed to be relative zero. Card columns 9-12 not punched; address counter incremented by 17.
T	Tag Error Card column 33 contains character other than blank, 0, 1, 2, or 3 in instruction statement.	Tag of zero assumed
U	Undefined Symbol Undefined symbol in expression	Expression set to absolute zero
O	Op Code Unrecognized ISS, ILS, ENT, LIBR, SPR, EPR, or ABS incorrectly placed.	Statement ignored and address counter incremented by 2. Statement ignored

1. Set the console Mode switch to RUN; press IMM STOP and RESET on the console.
 2. Place the tape in the reader so that the read starwheels are over one of the frames of delete code in the leader beyond the program ID.
 3. Press PROGRAM LOAD on the console.
2. Mount the source tape again and press PROGRAM START. A list tape is generated during pass 2.
 3. At the end of pass 2, simultaneously press FEED and DELETE to produce a few inches of trailer for the list tape that has been generated. Again, release FEED first.

Error Check

At the end of the tape are several inches of delete code, which constitute a trailer. When the last data character of the tape is read, the loading routine transfers control to the assembler. The assembler performs a checksum of the data read in from the tape and, if there is no read error, attempts to read a program tape. In this attempt, the last few inches of delete code are read and the reader stops with the not-ready code showing in the accumulator (see Error Codes).

If, however, the checksum test at the end of loading of the assembler tape is not satisfied, the assembler waits without reading past the trailer. The appearance of the checksum error code in the accumulator, in conjunction with this condition, indicates an error occurred in reading the assembler. The assembler should be reloaded. Repeated errors of this type indicate a defective assembler tape or a machine malfunction.

Pass 1 Procedure

In the absence of a checksum error, the program to be assembled can be entered when the assembler halts and the not-ready code shows in the accumulator. The input tape should have several inches of delete code leader preceding the first source record and should have a similarly punched trailer.

1. Mount the source program tape on the reader.
2. Press PROGRAM START to begin pass 1 of the assembly.

The tape reader reads the source program tape and waits at the end; the not-ready code is displayed in the accumulator.

Pass 2 Procedure

1. Before beginning pass 2, simultaneously press FEED and DELETE on the paper tape punch to produce a leader punched with the delete code. Release FEED before releasing DELETE to prevent the possibility of producing feed codes (00), which are not valid PTTC/8 characters.

ASSEMBLER TAPE FORMAT

The symbolic program input to the assembler is punched on PTTC/8 tape, one frame per character. The format of the tape records is the same as the card system except for the following:

1. The tape does not contain preceding blanks corresponding to card columns 1-20.
2. The tape does not contain blanks or data corresponding to card columns 72-80.
3. Trailing blanks need not be punched. Therefore, up to 51 characters (corresponding to card columns 21-71) can appear in the tape record.

Tape records are separated by NL (new line) characters (code DD). The delete character (code 7F) is ignored whenever it is read, but the reader stop character (RS, code 0D) causes the program reading the tape to wait and start reading again when PROGRAM START is pressed. The case shift characters (codes 0E, 6E) are allowed, but are not considered to occupy a space in the format.

The output from the assembler is a list tape, similar to the input tape, but with 20 frames added to the beginning of each record corresponding to card columns 1-20. The list tape is the input to the compressor program.

PAPER TAPE COMPRESSOR PROGRAM

COMPRESSOR PROCEDURE

The operating instructions for the paper tape compressor are the same as those for the assembler with the following exceptions:

- The compressor tape is somewhat shorter than the assembler tape.
- The input tape is the list tape generated during pass 2 of the assembler program.
- The output is a binary tape in standard system relocatable format (although the program may be

absolute). This tape is suitable for input to the relocating loader or core image converter.

- There is no second pass; therefore, the leader for the binary output tape should be punched before mounting the list tape.
- A listing on the Console Printer can be obtained by turning on Console Entry switch 15. Page skipping is implemented by means of a line counter for standard 11-inch-depth paper and is controlled by Console Entry switch 14. Fifty-eight lines are printed, then eight lines are skipped when switches 14 and 15 are on. The setting of the Console Entry switches can be altered at any time during the compression. A carrier return takes place before the first line is printed.

COMPRESSED TAPE FORMAT

The output from the compressor is a binary tape with records identical, word-for-word, to the corresponding card system cards. (ID and sequence numbers are omitted; refer to Appendix E for tape formats, and to Appendix H for illustration of typical compressed tape record.) Each binary record is preceded by a one-frame word count that gives the total number of words in the record (not counting the word count). Trailing zeros are deleted from all non-data records.

At the beginning and end of the tape and between records the delete codes are recognized.

All tapes should contain leaders and trailers of delete code.

PAPER TAPE WAITS AND ERROR CONDITIONS FOR ASSEMBLER AND COMPRESSOR

ERROR CODES

The following waits requiring manual intervention may occur while assembling or compressing:

<u>Accumulator Display</u>	<u>Cause</u>
7001	Reader not ready during loading
A001	Punch not ready
9001	Reader not ready during pass 1 processing
9002	Reader not ready during pass 2 processing
FF00	Checksum error (incorrect data read during loading)

ERROR CONDITIONS

- A checksum error indicates a defective tape or a machine malfunction in reading the processor tape. If several attempts to read the processor fail, the tape should be replaced. ~~See Appendix H for checksum restart procedure.~~
- If a character is read incorrectly, or an invalid character is in the input tape, the paper tape processors store a dollar sign (\$-code 5B) in place of the character that could not be translated during the input code conversion. In the assembler, a syntax error is indicated.

FORTRAN

The IBM 1130 FORTRAN Compiler is a program, supplied by IBM, that translates source program statements into a form suitable for execution on the IBM 1130 Computing System. The translated statements are known as the object program. The compiler detects certain errors in the source program and writes appropriate messages on the console printer or 1132 Printer. The compiler also produces a listing of the source programs and storage allocations. Programs intended for use with the 1800 card/paper tape system can be compiled with this compiler.

For 1130 FORTRAN I/O logical unit definitions, the I/O unit numbers are permanently set as described in Table 2.

• Table 2. 1130 FORTRAN I/O Logical Unit Definitions

Logical Unit Number	Device	Kind of Transmission	Record Size Allowed
1	Console printer	Output only	120
2	1442 Card Read Punch	Input/output	80
3	1132 Printer	Output only	1 carriage control + 120
4	1134-1055 Paper Tape Reader/Punch	Input/output	80, plus max. of 80 case shifts for PTT/C/8 code, plus NL code.
6	Keyboard	Input only	80
7	Plotter	Output only	120

CARD SYSTEM COMPILER LOADING

The FORTRAN compiler is divided into two logical parts: an input phase and a series of compilation phases. The loading sequence for compilation is shown in Figure 7.

CARD LOAD INSTRUCTIONS

1. Place the deck in the 1442 Card Read Punch in the order shown in Figure 7. (Sufficient blank cards should be placed behind the compiler to allow object deck punching.)
2. Press IMM STOP and RESET on the console.
3. Press START on the 1442.
4. Press PROGRAM LOAD on the console.

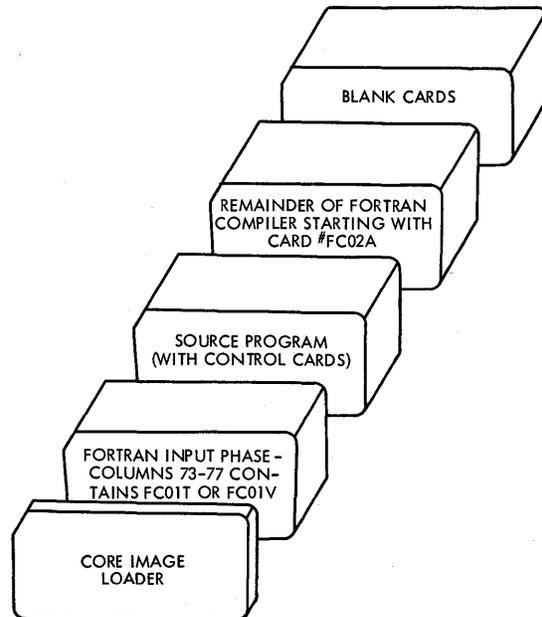


Figure 7. Loading Sequence for FORTRAN Compilation

The source deck and the relocatable object program deck will be in the outside stacker (stacker 2) when the compilation is completed.

CARD LOAD ERRORS

In addition to the card load errors described below, there are other program WAITS described under Core Image Loader in Appendix G.

Program Loop

- If the program loops after any phase of the compiler has been read in, check the system I/O units for a "not ready" condition.

1442 Errors

Read Errors. Clear the 1442 with the NPRO key. Place the two nonprocessed cards in front of the unprocessed portion of the deck in the reader hopper and press reader START. If the error occurs during

the core image loader program load it is also necessary to press PROGRAM START.

Feed Errors. Feed check errors should be treated in the same manner as read errors. Observe whether one or two cards are passed out on NPRO, and place them in front of the unprocessed portion of the deck.

Punch Errors. Clear the 1442 with the NPRO key, discard the erroneously punched card, replace the blank cards in the hopper, and press reader START to continue.

CONTROL RECORD OPTIONS

Option Control Cards

When using the FORTRAN compiler, the user can specify certain options by means of control cards. Control cards must precede the source program. They need not be in any specific order, but each must have an asterisk in column 1. The control record name can appear free-form, any place between columns 2 and 72, inclusive. Comments are not permitted on these control cards.

The *IOCS, *NAME, and *SAVE LOADER cards can be used only in mainline programs; the others may be used in both mainline programs and subroutines. Every card having an asterisk in column 1 will be listed on the output device assigned by the program.

Any unrecognizable FORTRAN control record is considered to be a comments record and the option is not performed. No error occurs during compilation.

The effect of each control card on compilation is described in the paragraphs that follow.

***IOCS (CARD, TYPEWRITER, KEYBOARD, 1132 PRINTER, PAPER TAPE, PLOTTER)** This card must be used to specify any I/O device required for execution of the program; however, only the devices required should be included. Because the *IOCS card can appear only in the mainline program, it must include all the I/O devices used by all FORTRAN subprograms that will be called. The device names must be in parentheses with a comma between each name.

Assembly language subroutines referenced by a FORTRAN mainline program can use any I/O subroutines for any device that is not mentioned in *IOCS and that is not on the same interrupt level as a device in *IOCS. Otherwise, the subprograms must use FORTRAN I/O routines (CARDZ, PAPTZ, PRNTZ, WRTYZ, TYPEZ, PLOTX).

***LIST SOURCE PROGRAM** The source program is listed as it is read in.

***LIST SUBPROGRAM NAMES** The names of all subprograms (including EXTERNAL subprograms) called directly by the compiled program are listed.

***LIST SYMBOL TABLE** The following are listed:

- Variable names and their relative addresses
- Statement numbers and their relative addresses
- Statement function names and their relative addresses
- Constants and their relative addresses

***LIST ALL** The source program, subprogram names, and symbol table are listed. If this card is used, the other *LIST control cards are not required.

***EXTENDED PRECISION** Variables and real constants are stored in three words instead of two, and the compiler generates linkage to extended precision routines. When this control card is used, the program does not conform to the ASA basic FORTRAN standard for data storage; it may require modification in order to be used with other FORTRAN systems.

***ONE WORD INTEGERS** Integer variables are allocated one word of storage rather than the same allocation used for real variables. Whether this control card is used or not, integer constants are always contained in one word. When this control card is used, the program does not conform to the ASA basic FORTRAN standard for data storage; it may require modification in order to be used with other FORTRAN systems.

***NAME XXXXX** The program name represented by XXXXX is punched in columns 73-77 of the object deck (columns 78-80 are sequence numbered). This control card is used only on mainline programs since subprogram names are automatically taken from the FUNCTION or SUBROUTINE statement and punched into the deck.

This Control record does not cause any punching in the object paper tape but does give a listable name to the mainline program.

***ARITHMETIC TRACE** The compiler generates linkage to trace routines whenever a value is assigned to a variable on the left of an equal sign. If Console Entry switch 15 is on at execution time, the trace

printout routine prints the value of the assigned variable (see Optional Tracing).

*TRANSFER TRACE. The compiler generates linkage to trace routines whenever an IF statement or computed GO TO statement is encountered. If Console Entry switch 15 is on at execution time, the trace printout routine prints the value of the IF expression or the value of the computed GO TO index (see Optional Tracing).

If tracing is requested, an *IOCS control card must also be present to indicate that either the console printer or the 1132 Printer is needed. If both the console printer and the 1132 Printer are indicated in the *IOCS card, the 1132 Printer is used for tracing.

The traced value for the assignment of a variable on the left of an equal sign of an arithmetic statement is printed with one leading asterisk. For the expression of an IF statement, the traced value is printed with two leading asterisks. The traced value for the index of a computed GO TO statement is printed with three leading asterisks.

*SAVE LOADER. If this control card is used, the source statement CALL LOAD can be included in the FORTRAN program. The loader will not be overlaid by variables in storage, and the CALL LOAD statements will cause the next program in the reader to be loaded, if it is in core image form. The CALL LOAD feature can be used only in card programs that have been converted to core image.

This feature is not available in the paper tape system.

** Header Information If the 1132 Printer is used for output, the information contained in card columns 3-72 is printed at the top of each page printed out during compilation.

Operating Notes - *LIST Control Cards

A constant in a STOP or PAUSE statement is treated as a hexadecimal number. This number and its decimal equivalent appear in the list of constants.

Variables and constants that require more than one word of storage have the address of the word nearest the zero address of the machine. In the case of arrays, the given address refers to the addressed word of the first element. In the case of a two- or three-word integer, the integer value is contained in the addressed word. The first variable listed might not be addressed at 0000 because room may be required for generated temporary storage locations.

The relative address for variables not in COMMON would be the actual address if the program started at storage location zero. The relative address for variables in COMMON would be the actual address if the machine had 32K storage. The relocating loader or core image converter makes any necessary adjustments. Variables in COMMON are adjusted to reside in the high-order core location of the machine being used (e.g., first COMMON variable will be loaded to 8191 on an 8K machine).

The actual storage location at which loading begins is variable and can be obtained as follows:

- When the relocating loader is used with a FORTRAN mainline deck that uses the *SAVE LOADER control card, relative address zero of the mainline program is at absolute location 027C.

This configuration can be used to debug the program up to, but not including, any CALL LOAD statements. The CALL LOAD statement can be executed only with the core image loader.

- When the relocating loader is used without the *SAVE LOADER control card, relative address zero must be computed as follows:

If the number of words for variables not in COMMON is equal to or greater than 454, relative address zero is at absolute location 00B6.

If the number of words for variables not in COMMON is less than 454, relative address zero is at absolute location 027C, minus the number of words for variables.

- When the core image loader is used with a FORTRAN mainline deck that uses the *SAVE LOADER control card, relative address zero of the mainline is at absolute location 00D6.
- When the core image loader is used without the *SAVE LOADER control card, relative address zero must be computed as follows:

If the number of words for variables not in COMMON is equal to or greater than 32, relative address zero is at absolute location 00B6.

If the number of words for variables not in COMMON is less than 32, relative address zero is at absolute location 00D6, minus the number of words for variables.

1800 SYSTEM CONTROL RECORD OPTIONS

The following control records are valid only for programs that are to be executed on an 1800 System.

***IOCS (1443 PRINTER, MAGNETIC TAPE)** The 1443 PRINTER or MAGNETIC TAPE options must be specified in an IOCS record for programs requiring those devices. Programs using either the 1443 PRINTER or MAGNETIC TAPE options can be executed only on an 1800 System.

***MULTIPLE DEVICE or *MULTIPLE DEVICE (CARD, TYPEWRITER, KEYBOARD)** This control record, which is valid for 1800 System programs only, indicates that more than one of the same type I/O device is attached to the system (for the purpose of this control record a 1053 and an 1816 are considered to be the same type).

The multiple device control card causes the compiler to generate linkage to the multiple device FORTRAN I/O subroutines. If a multiple device control card is not present in a mainline program, linkage to the single device (type Z) subroutines will be generated. A multiple device control card must also be present in any subprogram requesting a device specified in the mainline *IOCS control card.

Without the parentheses, the multiple device control card causes the 0 version (no error parameter) of the CARD, WRTY, and TYPE subroutines to be selected. This selection is accomplished via an A version of these subroutines.

With the parentheses, the multiple device control card signifies that multiple devices are being used and the 1 version (error parameter) of the subroutines are requested. CARD, TYPEWRITER, and KEYBOARD are the only names which will have any effect when used within the parentheses of this control card.

Table 2.1 lists the subroutine options.

OPTIONAL TRACING

The user can elect to trace only selected parts of the program by placing statements in the source program logic flow to start and stop tracing. This is done by executing a CALL to either subroutine:

- CALL TSTOP (to stop tracing)
- CALL TSTRT (to start tracing)

● Table 2-1. Subroutine Called for I/O Device in *IOCS

I/O Device Specified in *IOCS	SUBROUTINE CALLED		
	No Multiple Device Control Card	*Multiple Device Control Card	*Multiple Device (with units named) Control Card
Card	CARDZ	CARD0 via CARDA	CARD1 (CARD0 via CARDA if called but not specified in parentheses)
Magnetic Tape	MAGTZ	MAGT	MAGT
Typewriter	TYPEZ	TYPE0 via TYPEA	TYPE1 (TYPE0 via TYPEA if called but not specified in parentheses)
Keyboard	WRTYZ	WRTY0 via WRTYA	WRTY1 (WRTY0 via WRTYA if called but not specified in parentheses)
1443 Printer	PRNTZ	PRNT1	PRNT1
Paper Tape	PAPTZ	PAPT1	PAPT1
Plotter	PLOTX	PLOTX	PLOTX

Thus, tracing occurs only if:

- The trace control records were compiled with the source program.
- Console Entry switch 15 is on (can be turned off at any time).
- A CALL TSTOP has not been executed, or a CALL TSTRT has been executed since the last CALL TSTOP.

FORTRAN PRINTOUTS

Compilation Messages

Near the end of the compilation, core usage information and the features supported (control cards in deck) are printed out as follows:

FEATURES SUPPORTED
EXTENDED PRECISION
ONE WORD INTEGERS
TRANSFER TRACE
ARITHMETIC TRACE
IOCS
SAVE LOADER
CORE REQUIREMENTS FOR XXXXX
COMMON YYYYY VARIABLES YYYYY PROGRAM YYYYY

where XXXXX is the name of the program designated in the *NAME control record or in the SUBROUTINE or FUNCTION statement, and YYYYY is the number of words allocated for the specified parts of the program. Unreferenced statements are considered as possible errors and listed unconditionally.

Compilation Error Messages

During compilation, a check is made to determine if certain errors have occurred. If one or more of these errors is detected, the error indications are

printed at the conclusion of compilation and no object program is punched. Only one error is detected for each statement. In addition, because of the interaction of error conditions, the occurrence of some errors might prevent the detection of others until the detected errors are corrected.

The error message appears in the following format:

XXXXX + YYY ERROR TYPE AA

where XXXXX is the last encountered valid statement number, YYY is the count of statements from statement XXXXX, and AA is the error code. Declarative statements are not counted unless they contain errors. Statement numbers on declarative statements are ignored.

Error Code Definitions

1. Non-numeric character in statement number
2. More than five continuation cards, or a con-

- tinuation card encountered in an invalid position in the program
3. END statement missing
 4. Undeterminable, misspelled, or incorrectly formed statement
 5. Statement out of sequence
 6. Statement following transfer statement or STOP statement does not have statement number
 7. Name longer than five characters, or name not starting with an alphabetic character
 8. Incorrect or missing subscript within dimension information (DIMENSION, COMMON, or type)
 9. Duplicate statement number
 10. Syntax error in COMMON statement
 11. Duplicate name in COMMON
 12. Syntax error in FUNCTION or SUBROUTINE statement
 13. Formal parameter (dummy argument) appears in COMMON
 14. Name appears twice as a formal parameter in the argument list of a SUBROUTINE or FUNCTION statement
 15. *IOCS control record in a subprogram
 16. Syntax error in DIMENSION statement
 17. Subprogram name in DIMENSION statement
 18. Name dimensioned more than once, or not dimensioned on first appearance of name
 19. Syntax error in REAL, INTEGER, or EXTERNAL statement
 20. Subprogram name in REAL or INTEGER statement
 21. Name in EXTERNAL statement that is also in COMMON or DIMENSION statement
 22. IFIX or FLOAT in EXTERNAL statement
 23. Invalid real constant
 24. Invalid integer constant
 25. More than 15 names, or duplicate names, in statement function argument list
 26. Right parenthesis missing from a subscript expression
 27. Syntax error in FORMAT statement
 28. FORMAT statement without statement number
 29. Field width specification greater than 145
 30. In a FORMAT statement specifying E or F conversion, w greater than 127, d greater than 31, or d greater than w
 31. Subscript error in EQUIVALENCE statement
 32. Subscripted variable in a statement function
 33. Incorrectly formed subscript expression
 34. Undefined variable in subscript expression
 35. Number of subscripts in a subscript expression does not agree with the dimension information
 36. Invalid arithmetic statement or variable; or, in a FUNCTION subprogram, the left side of an arithmetic statement is a dummy argument or is in COMMON
 37. Syntax error in IF statement
 38. Invalid expression in IF statement
 39. Syntax error or invalid simple argument in CALL statement
 40. Invalid expression in CALL statement
 41. Invalid expression to the left of an equal sign in a statement function
 42. Invalid expression to the right of an equal sign in a statement function
 43. If an IF, GOTO, or DO statement, a statement number is missing, invalid, incorrectly placed, or is the number of a FORMAT statement
 44. Syntax error in READ or WRITE statement
 45. *IOCS card missing with a READ or WRITE statement (mainline program only)
 46. FORMAT statement number missing or incorrect in a READ or WRITE statement
 47. Syntax error in input/output list or an invalid list element; or, in a FUNCTION subprogram, an input list element is a dummy argument or is in COMMON
 48. Syntax error in GOTO statement
 49. Index of a computed GOTO missing, invalid, or not preceded by a comma
 50. *TRANSFER TRACE or *ARITHMETIC TRACE control record, but no *IOCS control record in a mainline program
 51. Incorrect nesting of DO statements, or the terminal statement of the associated DO statement is a GOTO, IF, RETURN, FORMAT, STOP, PAUSE, or DO statement
 52. More than 25 nested DO statements
 53. Syntax error in DO statement
 54. Initial value in DO statement is zero
 55. In a FUNCTION subprogram, the index of DO is a dummy argument or in COMMON
 56. Syntax error in BACKSPACE statement (1800 only)
 57. Syntax error in REWIND statement (1800 only)
 58. Syntax error in END FILE statement (1800 only)
 59. Syntax error in STOP statement
 60. Syntax error in PAUSE statement
 61. Integer constant in STOP or PAUSE statement greater than 9999
 62. Last executable statement before END statement is not a STOP, GOTO, IF, or RETURN statement
 63. Statement contains more than 15 different subscript expressions
 64. Statement too long to be scanned due to compiler expansion of subscript expressions or compiler addition of generated temporary storage locations
 65. *All variables undefined in an EQUIVALENCE list
 66. *Variable made equivalent to an element of an array in such a manner as to cause the array

- to extend beyond the origin of the COMMON area
67. *Two variables or array elements in COMMON are equated, or the relative locations of two variables or array elements are assigned more than once (directly or indirectly)
 68. Syntax error in EQUIVALENCE statement, or an illegal variable name in an EQUIVALENCE list
 69. Subprogram does not contain a RETURN statement, or a mainline program contains a RETURN statement

*The detection of a code 65, 66, or 67 error blocks any subsequent error with one of these three codes.

In addition, undefined variables are listed by name at the end of compilation. A variable is not defined unless it is on the left side of an assignment statement, is in the list of a READ statement, is an index variable of a DO statement, is in COMMON, is a formal parameter of a subprogram, or is equivalent to a defined variable. Undefined variables inhibit the output of the object program.

Undefined variables appearing only in subscript expressions generate error code 34, and are not listed here.

If the output of the object program has been inhibited, the following message is printed:

OUTPUT HAS BEEN SUPPRESSED

If an overflow condition results during compilation, the message,

PROGRAM LENGTH EXCEEDS CAPACITY

is printed.

OBJECT DECK LOADING PROCEDURES

After an object deck has been punched, it can be left in relocatable form or converted to core image form. Figure 8 shows the loading sequence for an object program in relocatable form; Figure 9 shows the loading sequence for an object program in core image form.

The subroutine library might have been divided into standard and extended precision decks. If this is the case, the precision deck that corresponds to the compilation precision should be used. If trace control cards were included at compilation time, Console Entry switch 15 must be turned on whenever the trace printout is desired.

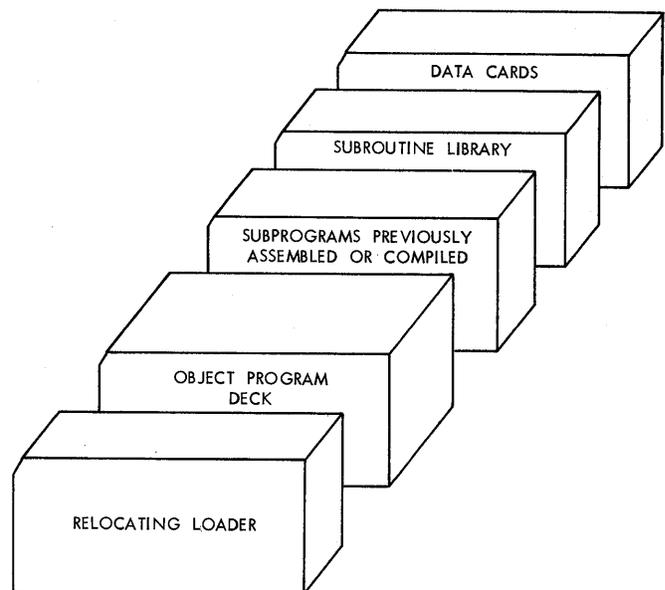


Figure 8. Loading Sequence for Object Program in Relocatable Form

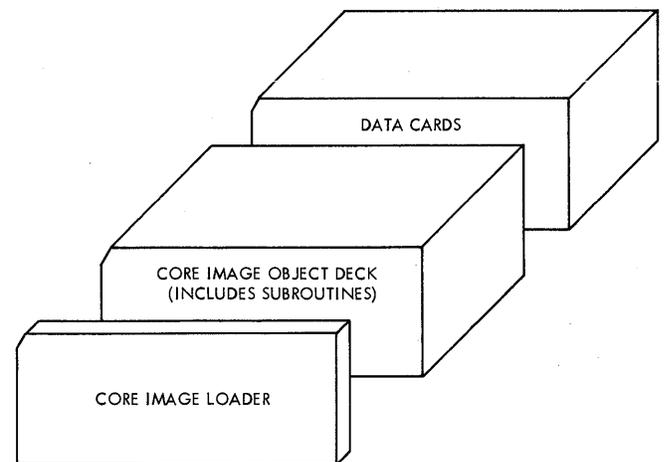


Figure 9. Loading Sequence for Object Program in Core Image Form

CARD LOAD INSTRUCTIONS

The object program operating procedure is as follows:

1. Place the deck in the 1442 Card Read Punch.
2. Press IMM STOP and RESET on the console.
3. Press START on the 1442.
4. Press PROGRAM LOAD on the console.

If the first I/O instruction is a WRITE to the 1442, no cards should be stacked behind the sub-

routine library deck. The library deck should be run out with the NPRO key before blank cards are placed in the reader. Ready the reader and press PROGRAM START to continue.

To restart a FORTRAN program that has been loaded in relocatable form, the procedure is as follows:

1. Set the Mode switch to LOAD.
2. Set 0198 into the Console Entry switches.
3. Press LOAD IAR.
4. Set the Mode switch to DISP.
5. Press PROGRAM START.
6. Set contents of the storage buffer register into the Console Entry switches.
7. Set the Mode switch to LOAD.
8. Press LOAD IAR.
9. Set the Mode switch to RUN.
10. Ready any I/O devices used with data repositioned for the first READ statement.
11. Press PROGRAM START.

The above procedure may not work if the program was stopped during an I/O operation. To continue, reload the object deck.

KEYBOARD INPUT OF DATA RECORDS

Data records of up to 80 characters can be read from the keyboard by a FORTRAN READ statement. Data values must be right-justified in their respective fields.

Keyboard Operation

If it is desirable to key in less than 80 characters, the EOF key can be pressed to stop transmittal. Also, the ERASE FIELD or BACKSPACE key can be pressed to restart the record transmittal if an error is detected while entering data. If the keyboard appears to be locked up, press REST KB to restore the keyboard. The correct case shift must be selected before data is entered.

Buffer Status After Keyboard Input

When the EOF key is pressed prior to completing a full buffer load of 80 characters, blanks are inserted in the remainder of the buffer. If more data is necessary to satisfy the list items, the remaining numeric fields (I, E, or F) are stored in core as zeros and remaining alphameric fields (A or H) are stored as blanks. Processing is continuous and no errors result from the above condition.

OBJECT PROGRAM DATA RECORD FORMAT

Data records of up to 80 EBCDIC characters in PTTC/8 code can be read or written by the FORTRAN object programs. The delete and new-line codes are recognized. Delete codes and case shifts are not included in the count of characters. If a new-line code is encountered before the 80th character is read, the record is terminated. If the 80th character is not a new-line code, the 81st character is read and assumed to be a new-line code. A new-line code is punched at the end of each output record.

FORTRAN I/O ERRORS

If input/output errors are detected during execution, the program stops and cannot be continued. The error is indicated by a display in the accumulator.

The error displays and their meanings are listed in Table 3.

When the output field is too small to contain the number, the field is filled with asterisks and execution is continued.

The input/output routines used by FORTRAN (PAPTZ, CARDZ, PRNTZ, WRTYZ, TYPEZ) wait on any I/O device error or device not in a ready condition. When the devices are ready, press PROGRAM START to execute the I/O operation.

Error detection in functional and arithmetic subroutines is possible by the use of source program statements. Refer to "FORTRAN Machine and Program Indicator Tests" in the manual, IBM 1130 FORTRAN Language (Form C26-5933).

Table 3. FORTRAN I/O Error Indications

Accumulator Display (Hexadecimal)	Error Definition
F001	Logical unit defined incorrectly, or no IOCS control record for I/O device specified
F002	Requested record exceeds allocated buffer
F003	Illegal character encountered on input
F004	Exponent too large or too small in input
F005	More than one E encountered on input
F006	More than one sign encountered on input
F007	More than one decimal point encountered on input
F008	Read on output-only device, or write on input-only device

PAPER TAPE SYSTEM COMPILER LOADING

In the paper tape FORTRAN system, the compiler loading sequence should be as follows:

1. Core image loader
2. FORTRAN input phase
3. Source program
4. Balance of FORTRAN compiler

COMPILER LOAD INSTRUCTIONS

The following procedure should be followed to initiate the program load:

1. Set the Mode switch to RUN and press IMM STOP and RESET on the console.
2. Ready the paper tape reader. Place the tape in the reader, positioning the tape so that the read starwheels are over one of the frames of delete code in the leader beyond the program ID.
3. Ready the paper tape punch. Create a leader of delete codes by simultaneously pressing DELETE and FEED. Release FEED before releasing DELETE.
4. Press PROGRAM LOAD on the console.

Any tape information can be loaded from separate strips of tape (i. e., control records, source statements). Each segment must be preceded by and followed by two or more inches of tape delete characters.

Program printouts and program options are the same as described in the preceding sections on the card system.

PAPER TAPE LOAD ERRORS

In addition to the paper tape load errors described below, there are other program WAITs described under Core Image Loader in Appendix G.

Character Other Than Delete Read After Source Tape

Phase 2 of the FORTRAN compiler is preceded by two records, each containing a single dollar sign character. Detection of the first of these records causes the compiler to pass the second record

without processing and start loading phase 2 of the compiler when the first valid record containing program data is read.

If any character (other than DELETE) is read between the last character of the source tape (always a new line character) and the first dollar sign record of the second FORTRAN compiler tape, the compiler will not load correctly. A checksum error will occur. If a listing is being made and this error occurs, a dollar sign appears as the last character on the last printed line. In order to prevent this error from occurring, the user should be certain that the end of the last source tape is trimmed so that only DELETE characters are read. Do not open the read head while reading the trailer on this tape, as a NON-DELETE character can be read.

Parity Error

A parity error during reading will cause the system to WAIT at 01E2 during phase 01W (1132 Printer output) and at 01FA during phase 01U (Console Printer output). The WAIT address is displayed in the storage address register.

To continue, back the tape one frame and press PROGRAM START.

Reader Not Ready

A reader not-ready condition will cause the system to WAIT at 0190 during phase 01W (1132 Printer output) and at 0194 during phase 01U (Console Printer output). The WAIT address is displayed in the storage address register.

To continue, clear the cause of the not-ready condition; check that the tape tension switch is closed, check that the tape is loaded properly and is feeding freely. When the tape reader is ready, press PROGRAM START.

Punch Not Ready

A punch not ready condition will cause the system to WAIT with 050A in the storage address register. To continue, clear the cause of the not-ready condition; check that the tape tension switch is closed; check that the tape is loaded properly and feeding freely. When the tape punch is ready, press PROGRAM START.

SOURCE RECORD FORMAT

Paper tape records must be in pseudo card image format as follows:

- Control records must start with an asterisk in the first character of the paper tape record.
- Source records must have a five-character field corresponding to a statement number, whether a statement number is present or not.
- The sixth character of the source record is a continuation character.
- The source records must not be more than 72 characters long (excluding case shifts, deletes, and new line characters), but can be shorter. If record IDs are desired for listing purposes, they can appear in character positions 73-80.

Characters must be in PTTC/8 code and each record must be terminated by a new line character.

Paper tapes can be produced by using the Key-board Utility Routine described in the Utility Routines section of this manual.

PAPER TAPE OBJECT PROGRAM LOADING PROCEDURES

After an object program tape has been punched, it can be left in relocatable form or converted to core image.

OBJECT PROGRAM LOAD SEQUENCE

The following sections describe the loading sequence for the relocating loader and for the core image loader. If TRACE control records were included when the object program was compiled, Console Entry switch 15 must be on to execute the trace.

Loading Sequence, Relocating Loader

1. Relocating loader
2. Object program
3. FORTRAN or assembler subprograms
4. Subroutines. The precision tape that corresponds to the compiler precision should be used.
5. Data records

Loading Sequence, Core Image Loader

1. Core image loader
2. Object Program
3. Data records

OBJECT PROGRAM LOAD INSTRUCTIONS

1. Set the Mode switch to RUN and press IMM STOP and RESET on the console.
2. Ready the paper tape reader and punch.
3. Press PROGRAM LOAD on the console.

their timing & core requirements

The 1130 Subroutine Library consists of a group of subroutines that aid the programmer in making efficient use of the IBM 1130 Computing System. Descriptions of the subroutines and methods for programming them are contained in the publication, IBM 1130 Subroutine Library (Form C26-5929); therefore, these operating procedures concern only the layout of the subroutine library and the methods by which the user prepares the library for loading and adds or deletes portions of it to suit his needs.

The following paragraphs describe the use of the card and paper tape subroutine libraries and discusses pre-operative errors and I/O error re-starts where special handling is required.

ISS PRE-OPERATIVE ERRORS

A pre-operative error is an error condition detected before an I/O operation is started. It denotes either an illegal LIBF parameter, an illegal specification in I/O area, or a device not-ready condition.

This error causes a branch to location 0029 and the following conditions:

- The ~~I-counter~~ ^{IAR} displays the address 002A.
- The accumulator displays an error code represented by four hexadecimal digits.
 - Digit 1 identifies the ISS subroutine called:
 - 1 - CARD0 or CARD1
 - 2 - TYPE0 or WRTY0
 - 3 - PAPT1 or PAPTN
 - 5 - DISK0, DISK1, or DISKN
 - 6 - PRNT1
 - 7 - PLOT1
 - Digits 2 and 3 are not used.
 - Digit 4 identifies the error:
 - 0 - Device not ready
 - 1 - Illegal LIBF parameter or illegal specification in I/O area
- Location 0028 contains the address of the LIBF in question.

Since the relocating loader stores a WAIT instruction in location 0029 and a BSC I 0028 in locations 002A and 002B, the LIBF can be executed again by pressing PROGRAM START.

When a pre-operative error is encountered the operator can:

- Correct the error condition, if possible, and press PROGRAM START, or
- Note the contents of the accumulator and location 0028, dump core storage, and proceed with the next job.

CARD SUBROUTINE (CARD0 AND CARD1) ERRORS

Error Parameters

CARD0. There is no error parameter. If an error is detected during processing of an operation-complete interrupt, the subroutine loops internally, with the interrupt level 4 on, until the 1442 becomes ready and then retries the operation.

CARD1. There is an error parameter. If an error is detected during processing of an operation-complete interrupt, the user program can elect to terminate (clear "routine busy" and the interrupt level) or to retry. A retry consists of looping internally, with interrupt level 4 on, until the 1442 becomes ready and then reinitiating the function.

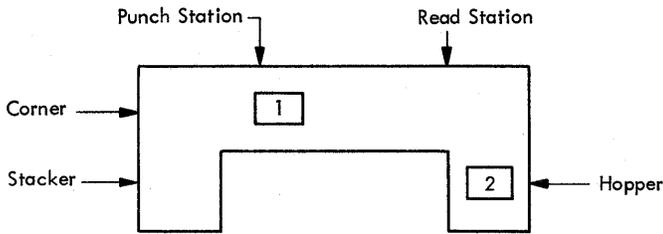
1442 Errors and Operator Procedures

If a 1442 error occurs, the 1442 becomes not ready until the operator has intervened. Unless the stop is caused by a stacker full (no indicator) or chip box indication, the 1442 card path must be cleared before proceeding. The 1442 error indicators and the position of the cards in the feed path should be used to determine which cards must be placed back in the hopper.

As far as the card subroutines are concerned, a retry consists of positioning the cards (i.e., skipping the first card in the hopper, if necessary, on a read or feed operation) and reinitiating the function whenever the card reader becomes ready.

Hopper Misfeed. Indicates that card 2 failed to pass properly from the hopper to the read station during the card 1 feed cycle.

Card positions after error:

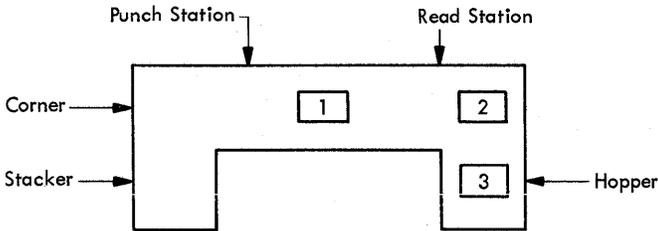


Error indicator: HOPR

Operator procedure: When program halts, empty hopper, press NPRO to eject card 1, place card 1 in deck in front of card 2, return deck to hopper, and ready the 1442.

Feed Check (punch station). Indicates that card 1 is improperly positioned in the punch station at the completion of its feed cycle.

Card positions after error:

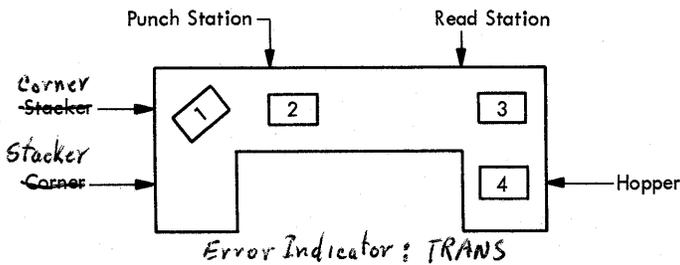


Error indicator: PUNCH STA

Operator procedure: When program halts, empty hopper, clear 1442 card path, place cards 1 and 2 in hopper before card 3, and ready the 1442.

Transport. Indicates that card 1 has jammed in the stacker during the feed cycle for card 2.

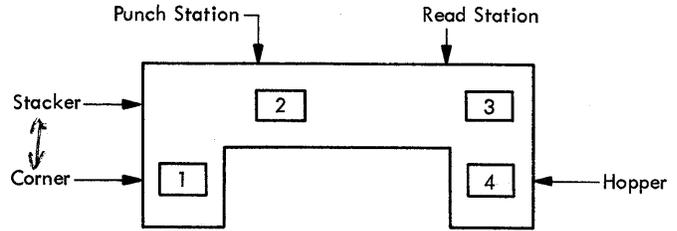
Card positions after error:



Operator procedure: When program halts, empty hopper, clear 1442 card path, place cards 2 and 3 in hopper before card 4, and ready the 1442.

Feed Cycle. Indicates that the 1442 took an un-called for feed cycle and, therefore, cards 1, 2, and 3 are each one station farther ahead in the 1442 card path than they should be.

Card positions after error:

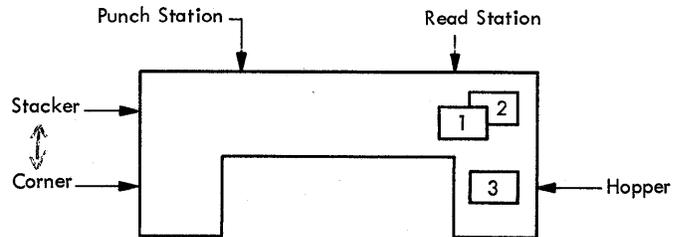


Error indicator: FEED CLU

Operator procedure: When program halts, empty hopper, press NPRO to eject cards 2 and 3, place cards 1, 2, and 3 in hopper before card 4, and ready the 1442.

Feed Check (read station). Indicates that card 1 failed to eject from the read station during its feed cycle.

Card positions after error:

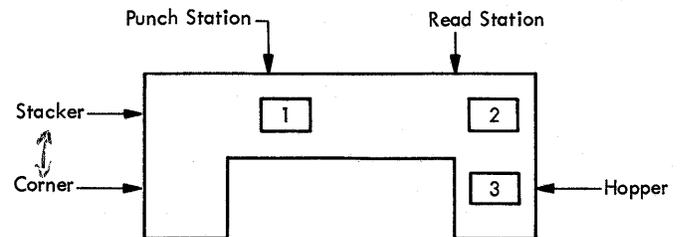


Error indicator: READ STA

Operator procedure: When program halts, empty hopper, clear 1442 card path, place cards 1 and 2 in hopper before card 3, and ready the 1442.

Read Registration. Indicates incorrect card registration or a difference between the first and second reading of a column.

Card positions after error:

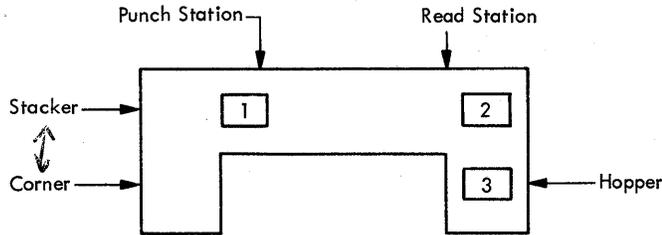


Error indicator: READ REG

Operator procedure: See Feed Check (punch station). Repeated failures of this type might indicate a machine malfunction.

Punch Check. Indicates an error in output punching.

Card positions after error:



Error indicator: PUNCH

Operator procedure: When program halts, empty hopper, check card position and press NPRO to clear 1442 card path. If necessary, correct card 1 to pre-punched state. Place (corrected) card 1 and card 2 in hopper before card 3 and ready the 1442.

CONSOLE PRINTER SUBROUTINE (TYPE0 AND WRTY0) ERRORS

If the carrier attempts to print beyond the manually positioned margins, a carrier restore (independent of the program) occurs.

Subroutine printing begins wherever the carrier is positioned as a result of the previous print operation. There is no automatic carrier return as a result of an LIBF.

If the console printer indicates a not-ready condition when selected, the subroutines loop internally, with interrupt level 4 on, waiting for the console printer to become ready. Operator procedures are as follows:

1. Press IMM STOP on the console.
2. Ready the console printer.
3. Press PROGRAM START on the console.

KEYBOARD SUBROUTINE (TYPE0) FUNCTIONS

Erase Field

When the Erase Field key is pressed a character interrupt signals the interrupt response routine that the previously entered keyboard message is in error and will be re-entered. The routine prints two slashes on the console printer, restores the carrier to a new line, and prepares to replace the

old message in the I/O area with the new message. The operator then enters the new message. The old message in the I/O area is not cleared. The new message overlays the previous message, character by character. If the previous message was longer than the new message, characters from the previous message remain (following the NL character which terminated the new message).

Backspace

When the backspace key is pressed, the last graphic character entered is slashed and the address of the next character to be read is decremented by +1. If the backspace key is pressed twice, consecutively, the character address is decremented by +2, but only the last graphic character is slashed. For example, assume that ABCDE has been entered and the backspace key pressed three times. The next graphic character replaces the C, but only the E is slashed. If the character F had been used for replacement the paper would show ABCD~~E~~FFF, but ABFFFF would be stored in the buffer.

PAPER TAPE (PAPT1 and PAPTn) SUBROUTINES

If the reader or punch becomes not ready during an I/O operation, the subroutines exit to the user via the error parameter. The user can request the subroutine to terminate (clear device busy and interrupt level) or to loop on not ready waiting for operator intervention (interrupt level 4 on).

The following procedure should be used to clear a paper tape not-ready condition:

1. Press IMM STOP on the console.
2. Ready the paper tape unit.
3. Press PROGRAM START on the console.

To load the paper tape reader, place the tape so that the delete characters punched in the leader are under the read starwheels. To begin reading at any point in the tape other than the leader, place the tape so that the frame (character position) preceding the character to be read is under the read starwheels. The first start reader control after tape is loaded or repositioned causes the reader to skip the character under the read starwheels and load the next character into the buffer.

ADDING AND REMOVING SUBROUTINES

Subroutines can be added or removed as desired by the user. However, the following rules should be followed to ensure proper operation:

- A user-written CALL subroutine should be placed in front of the IBM CALL subroutines. This allows the user's subroutine to call any IBM subroutine without having to pass the subroutine deck more than once.
- A user-written LIBF subroutine should be placed directly in back of the EOD-1 cards (immediately preceding the IBM LIBF subroutines), for the same reason. Since LIBF subroutines cannot call CALL subroutines, no problems are created by having user-written LIBF subroutines behind CALL subroutines.
- The user should not remove subroutines that are called by other subroutines left in the deck. (See Table 4 for a list of subroutines called by other subroutines.)

1130 SUBROUTINE LIBRARY LISTING

Table 4 is a list of the subroutines in the library deck provided with the IBM 1130 Subroutine Library

Programming System. The subroutines are listed in the same order as they appear in the library deck.

Cards in the library deck can be identified by the punches in columns 73-80:

Cols. 73-77	Program ID (mnemonics, in table)
Col. 78	Routine modification character: 0-9, A-Z
Cols. 79-80	Card sequence number

Tapes in the subroutine library can be identified by the program name and program ID (BP number, in table) punched in the tape leader.

<i>BP01</i>	BP-09	ISS, ILS, and conversion subroutines
<i>BP02</i>	BP-10	Arithmetic, functional, and FORTRAN-supplied subroutines — standard package
<i>BP03</i>	BP-11	Arithmetic, functional, and FORTRAN-supplied subroutines — extended package

Table 4. Subroutine Library Listing

Subroutines	Names	Other Subroutines Required	Card ID (Cols 73-77)	Tape ID (in leader)
<u>Utility Calls</u>				
Set Pack Initialization Routine for DISK0	SPIR0	DISK0	SPIR0	BP01
Set Pack Initialization Routine for DISK1	SPIR1	DISK1	SPIR1	BP01
Set Pack Initialization Routine for DISKN	SPIRN	DISKN	SPIRN	BP01
Selective Dump on Console Printer	DMTD0, DMTX0	WRTY0	DMT0	BP01
Selective Dump on 1132 Printer	DMPD1, DMPX1	PRNT1	DMP1	BP01
Dump 80 Routine	DMP80	None	DMP80	BP01
<u>Common FORTRAN Calls</u>				
Test Data Entry Switches	DATSW	None	DATSW	BP02, BP03
Divide Check Test	DVCHK	None	DVCHK	BP02, BP03
Functional Error Test	FCTST	None	FCTST	BP02, BP03
Restore Core Image Loader	LOAD	None	LOAD	
Overflow Test	OVERF	None	OVERF	BP02, BP03
Sense Light Control and Test	SLITE, SLITT	None	SLITE	BP02, BP03
FORTRAN Trace Stop	TSTOP	TSET	TSTOP	BP02, BP03
FORTRAN Trace Start	TSTRT	TSET	TSTRT	BP02, BP03
Integer Transfer of Sign	ISIGN	None	ISIGN	BP02, BP03
<u>Extended Arith/Funct Calls</u>				
Extended Precision Hyperbolic Tangent	ETANH, ETNH	EEXP, ELD/ESTO, EADD, EDIV, EGETP	ETANH	BP03
Extended Precision A**B Function	EAXB, EAXBX	EEXP, ELN, EMPY	EAXB	BP03
Extended Precision Natural Logarithm	ELN, EALOG	XMD, EADD, EMPY, EDIV, NORM, EGETP	ELN	BP03
Extended Precision Exponential	EEXP, EXPN	XMD, FARC, EGETP	EEXP	BP03
Extended Precision Square Root	ESQR, ESQRT	ELD/ESTO, EADD, EMPY, EDIV, EGETP	ESQR	BP03
Extended Precision Sine-Cosine	ESIN, ESINE, ECOS, ECOSN	EADD, EMPY, NORM, XMD, EGETP	ESIN	BP03
Extended Precision Arctangent	EATN, EATAN	EADD, EMPY, EDIV, XMD, EGETP, NORM	EATN	BP03
Extended Precision Absolute Value Function	EABS, EAVL	EGETP	EABS	BP03
<u>FORTRAN Sign Transfer Calls</u>				
Extended Precision Transfer of Sign	ESIGN	ESUB, ELD	ESIGN	BP03
Standard Precision Transfer of Sign	FSIGN	FSUB, FLD	FSIGN	BP02

(Continued)

Table 4. Subroutine Library Listing

Subroutines	Names	Other Subroutines Required	Card ID (Cols 73-77)	Tape ID (in leader)
<u>Standard Arith/Funct Calls</u>				
Standard Precision Hyperbolic Tangent	FTANH, FTNH	FEXP FLD/FSTO, FADD, FDIV, FGETP	FTANH	BP02
Standard Precision A**B Function	FAXB, FAXBX	FEXP, FLN, FMPY	FAXB	BP02
Standard Precision Natural Logarithm	FLN, FALOG	FSTO, XMDS, FADD, FMPY, FDIV, NORM, FGETP	FLN	BP02
Standard Precision Exponential	FEXP, FXPN	XMDS, FARC, FGETP	FEXP	BP02
Standard Precision Square Root	FSQR, FSQRT	FLD/FSTO, FADD, FMPY, FDIV, FGETP	FSQR	BP02
Standard Precision Sine-Cosine	FSIN, FSINE, FCOS, FCOSN	FADD, FMPY, NORM, XMDS, FSTO, FGETP	FSIN	BP02
Standard Precision Arc tangent	FATN, FATAN	FADD, FMPY, FDIV, XMDS, FSTO, FGETP	FATN	BP02
Standard Precision Absolute Value Function	FABS, FAVL	FGETP	FABS	BP02
<u>Common Arith/Funct Calls</u>				
Fixed Point (Fractional) Square Root	XSQR	None	XSQR	BP02, BP03
Integer Absolute Function	IABS	None	IABS	BP02, BP03
Floating Binary/EBC Decimal Conversions	FBTD (BIN. TO DEC.) FDTB (DEC. TO BIN.)	None	FBTD	BP02, BP03
End of Deck One (EOD1) Card/Tape Records (2) Type C and D			EOD01	Contained in BP01, BP02, BP03
<u>FORTRAN Extended Trace Routines</u>				
Floating Variable Trace	VARI, VARIX	ESTO, TTEST, VWRT VIOF, VCOMP	VARI	BP03
Fixed Variable Trace	VIAR, VIARX	TTEST, VWRT, VIOI VCOMP	VIAR	BP03
Floating IF Trace	VIF	TTEST, VWRT, VIOF, VCOMP	VIF	BP03
Fixed IF Trace	VIIF	TTEST, VWRT, VIOI, VCOMP	VIIF	BP03
GOTO Trace	VGOTO	TTEST, VWRT, VIOI, VCOMP	VGOTO	BP03
<u>FORTRAN I/O (Ext.)</u>				
Extended FORTRAN Input/Output	VFIO, VIOI, VIOAI, VIOF, VIOAF, VIOFX, VCOMP, VWRT, VRED, VIOIX	FLOAT, ELD/ESTO, IFIX	VFIO0	BP03
<u>FORTRAN Standard Trace Routines</u>				
Floating Variable Trace	WARI, WARIX	FSTO, TTEST, WWRT, WIOF, WCOMP	WARI	BP02
Fixed Variable Trace	WIAR, WIARX	TTEST, WWRT, WIOI, WCOMP	WIAR	BP02
Floating IF Trace	WIF	FSTO, TTEST, WWRT, WIOF, WCOMP	WIF	BP02
Fixed IF Trace	WIIF	TTEST, WWRT, WIOI, WCOMP	WIIF	BP02
GOTO Trace	WGOTO	TTEST, WWRT, WIOI, WCOMP	WGOTO	BP02
<u>FORTRAN I/O (Std.)</u>				
Standard FORTRAN Input/Output	WFIO, WIOI, WIOAI, WIOF, WIOAF, WIOFX, WCOMP, WWRT, WRED, WIOIX	FLOAT, FLD/FSTO, IFIX	WFIO0	BP02

(Continued)

Table 4. Subroutine Library Listing

Subroutines	Names	Other Subroutines Required	Card ID (Cols 73-77)	Tape ID (in leader)
<u>FORTRAN Common LIBF's</u>				
FORTRAN Pause	PAUSE	None	PAUSE	BP02, BP03
FORTRAN Stop	STOP	None	STOP	BP02, BP03
FORTRAN Subscript Displacement Calculation	SUBSC	None	SUBSC	BP02, BP03
FORTRAN Subroutine Initialization	SUBIN	None	SUBIN	BP02, BP03
FORTRAN Trace Test and Set	TTEST, TSET	None	TTEST	BP02, BP03
<u>FORTRAN I/O and Conversion Routines</u>				
FORTRAN Card Routine	CARDZ	HOLEZ	CARDZ	BP02, BP03
FORTRAN Paper Tape Routine	PAPTZ	None	PAPTZ	BP02, BP03
FORTRAN 1132 Printer Routine	PRNTZ	None	PRNTZ	BP02, BP03
FORTRAN Keyboard-Typewriter Routine	TYPEZ	GETAD, EBCTB, HOLEZ	TYPEZ	BP02, BP03
FORTRAN Typewriter Routine	WRTYZ	GETAD, EBCTB	WRTYZ	BP02, BP03
FORTRAN Hollerith to EBCDIC Conversion	HOLEZ	GETAD, EBCTB, HOLTB	HOLEZ	BP02, BP03
FORTRAN Get Address Routine	GETAD	None	GETAD	BP02, BP03
FORTRAN EBCDIC Table	EBCTB	None	EBCTB	BP02, BP03
FORTRAN Hollerith Table	HOLTB	None	HOLTB	BP02, BP03
<u>Extended Arith/Funct LIBF's</u>				
Extended Precision Get Parameter Subroutine	EGETP	ELD	EGETP	BP03
Extended Precision A**I Function	EAXI, EAXIX	ELD/ESTO, EMPY, EDVR	EAXI	BP03
Extended Precision Divide Reverse	EDVR, EDVRX	ELD/ESTO, EDIV	EDVR	BP03
Extended Precision Float Divide	EDIV, EDIVX	XDD, FARC	EDIV	BP03
Extended Precision Float Multiply	EMPY, EMPYX	XMD, FARC	EMPY	BP03
Extended Precision Subtract Reverse	ESBR, ESBRX	EADD	ESBR	BP03
Extended Add-Subtract	EADD, ESUB, EADDX, ESUBX	FARC, NORM	EADD	BP03
Extended Load-Store	ELD, ELDX, ESTQ, ESTOX	None	ELD	BP03
<u>Standard Arith/Funct LIBF's</u>				
Standard Precision Get Parameter Subroutines	FGETP	FLD	FGETP	BP02
Standard Precision A**I Function	FAXI, FAXIX	FLD/FSTO, FMPY, FDVR	FAXI	BP02
Standard Precision Divide Reverse	FDVR, FDVRX	FLD/FSTO, FDIV	FDVR	BP02
Standard Precision Float Divide	FDIV, FDIVX	FARC	FDIV	BP02
Standard Precision Float Multiply	FMPY, FMPYX	XMDS, FARC	FMPY	BP02
Standard Precision Subtract Reverse	FSBR, FSBRX	FADD	FSBR	BP02
Standard Add-Subtract	FADD, FSUB, FADDX, FSUBX	NORM, FARC	FADD	BP02
Standard Load-Store	FLD, FLDX, FSTO, FSTOX	None	FLD	BP02

(Continued)

Table 4. Subroutine Library Listing

Subroutines	Names	Other Subroutines Required	Card ID (Cols 73-77)	Tape ID (in leader)
<u>Standard Arith/Funct LIBFS (Cont)</u>				
Standard Precision Fractional Multiply	XMDS	None	XMDS	BP02
<u>Common Arith/Funct LIBFS</u>				
Fixed Point (Fractional) Double Divide	XDD	XMD	XDD	BP02, BP03
Fixed Point (Fractional) Double Multiply	XMD	None	XMD	BP02, BP03
Sign Reversal Function	SNR	None	SNR	BP02, BP03
Integer to Floating Point Function	FLOAT	NORM	FLOAT	BP02, BP03
Floating Point to Integer Function	IFIX	None	IFIX	BP02, BP03
I**J Integer Function	FIXI, FIXIX	None	FIXI	BP02, BP03
Normalize Subroutine	NORM	None	NORM	BP02, BP03
Floating Accumulator Range Check Subroutine	FARC	None	FARC	BP02, BP03
<u>Interrupt Service Subroutines</u>				
Card Input/Output (No Error Parameter)	CARD0	Interrupt Level Zero/Four	CARD0	BP01
Card Input/Output (Error Parameter)	CARD1	Interrupt Level Zero/Four	CARD1	BP01
One Sector Disk Input/Output	DISK0	Interrupt Level Two	DISK0	BP01
Multiple Sector Disk Input/Output	DISK1	Interrupt Level Two	DISK1	BP01
High-Speed Multiple Sector Disk Input/Output	DISKN	Interrupt Level Two	DISKN	BP01
Paper-Tape Input/Output	PAPT1	Interrupt Level Four	PAPT1	BP01
Simultaneous Paper Tape Input/Output	PAPTN	Interrupt Level Four	PAPTN	BP01
Plotter Output Routine	PLOT1	Interrupt Level Three	PLOT1	BP01
1132 Printer Output Routine	PRNT1	Interrupt Level One	PRNT1	BP01
Keyboard/Console Printer Input/Output	TYPE0	HOLL, PRTY, Interrupt Level Four	TYPE0	BP01
Console Printer Output Routine	WRTY0	Interrupt Level Four	WRTY0	BP01
<u>Conversion Routines</u>				
Binary Word to 6 Decimal Characters (Card Code)	BINDC	None	BINDC	BP01
Binary Word to 4 Hexadecimal Characters (Card Code)	BINHX	None	BINHX	BP01
6 Decimal Characters (Card Code) to Binary Word	DCBIN	None	DCBIN	BP01
EBCDIC to Console Printer Output Code	EBPRT	EBPA, PRTY	EBPRT	BP01
Card Code to EBCDIC-EBCDIC to Card Code	HOLEB	EBPA, HOLL	HOLEB	BP01
Card Code to Console Printer Output Code	HOLPR	HOLL, PRTY	HOLPR	BP01

(Continued)

Table 4. Subroutine Library Listing

Subroutines	Names	Other Subroutines Required	Card ID (Cols 73-77)	Tape ID (in leader)
<u>Conversion Routines (Cont)</u>				
4 Hexadecimal Characters (Card Code) to Binary Word	HXBIN	None	HXBIN	BP01
PTTC/8 to EBCDIC - EBCDIC to PTTC/8	PAPEB	EBPA	PAPEB	BP01
PTTC/8 to Card Code - Card Code to PTTC/8	PAPHL	EBPA, HOLL	PAPHL	BP01
PTTC/8 to Console Printer Output Code	PAPPR	EBPA, PRTY	PAPPR	BP01
Card Code to EBCDIC- EBCDIC to Card Code	SPEED	None	SPEED	BP01
EBCDIC and PTTC/8 Table	EBPA	None	EBPA	BP01
Card Code Table	HOLL	None	HOLL	BP01
Console Printer Output Code Table	PRTY	None	PRTY	BP01
<u>Interrupt Level Subroutines</u>				
Interrupt Level Zero Routine	Entry by Interrupts	None	ILS00	BP01
Interrupt Level One Routine	Entry by Interrupts	None	ILS01	BP01
Interrupt Level Two Routine	Entry by Interrupts	None	ILS02	BP01
Interrupt Level Three Routine	Entry by Interrupts	None	ILS03	BP01
Interrupt Level Four Routine	Entry by Interrupts	None	ILS04	BP01
End of Deck Two (EOD2) Cards/Records (4) Types C, C, C, E.			EOD02	Contained in BP01, BP02, BP03

(Concluded)

UTILITY ROUTINES

This section of the manual provides programmers with specifications and operating procedures for the IBM 1130 Utility Routines. It includes descriptions of the functions of the routines and general requirements for their use. The utility routines are provided in two forms: card and paper tape. Once loaded into storage, the programs are the same. Each routine is contained in a separate card deck or paper tape. All references to control cards, data cards, etc., in this publication are also applicable in terms of control records, data records, etc., for the paper tape system. A typical compressed tape record is illustrated in Appendix H.

The utility routines comprise the following:

1. Loaders
2. Core image converter
3. Input/Output
4. Dump (Dump from 00F0 and the dump between limits on the card read punch are available in card form only)
5. Keyboard
6. Card reproducing (card only)
7. Console
8. Disk pack initialization
9. Construct paper tape (paper tape only)

LOADING ROUTINES

The various user-written programs and routines that make up a complete job are compiled or assembled separately. To execute a job, all the component programs must be loaded in binary form. This operation can be performed by either of two IBM-supplied loading routines: the relocating loader, or the core image loader. The core image loader requires the object program to be in core image format. This conversion is accomplished with the core image converter.

Machine stops and error waits for the card/paper tape relocating loader, core image loader, and core image converter are listed in Appendix G.

CARD SYSTEM RELOCATING LOADER

The relocating loader loads compressed object programs and subroutines that are in relocatable format

(see Appendix E). Only the subroutines required by the main program (mainline) are loaded and, at the completion of the loading process, the loader branches to the execution address of the object program. Figure 10 shows the stacked input arrangement required to load a program with the relocating loader.

Building the Program Deck

EOD1 and EOD2 Loader Cards. The two EOD1 cards (types C, D) and four EOD2 cards (types C, C, C, E), included as part of the IBM subroutine library, must always be used with the relocating loader. The EOD1 cards terminate processing of two-word called (CALL) subroutines thereby making it possible for the relocating loader to recognize and load one-word called (LIBF) subroutines and to process EOD2 cards. The EOD2 cards terminate processing of one-word called (LIBF) subroutines thereby making it possible for the relocating loader to initiate terminating procedures such as the completion of transfer vector processing and loading of the interrupt branch addresses.

With the IBM Subroutine Library Deck. Place the mainline object deck in front of the first two-word, user-written, called subroutine. Place the two-word, IBM-supplied, called subroutines plus the EOD1 cards after the last two-word, user-written, called subroutine.

Place all one-word, user-written, called subroutines after the EOD1 cards and before the first one-word, IBM-supplied, called subroutine. Follow the last one-word, IBM-supplied, called subroutine with the EOD2 cards. The program deck is now ready to be loaded with the relocating loader.

Without the IBM Subroutine Library Deck. Place the mainline object deck in front of the first two-word, user-written, called subroutines. Place the EOD1 cards after the last two-word, user-written, called subroutine. If no two-word called subroutines are needed, place the EOD1 cards immediately after the mainline object deck.

Immediately after the EOD1 cards, place the one-word, user-written, called subroutines, followed by the EOD2 cards. If no one-word called subroutines

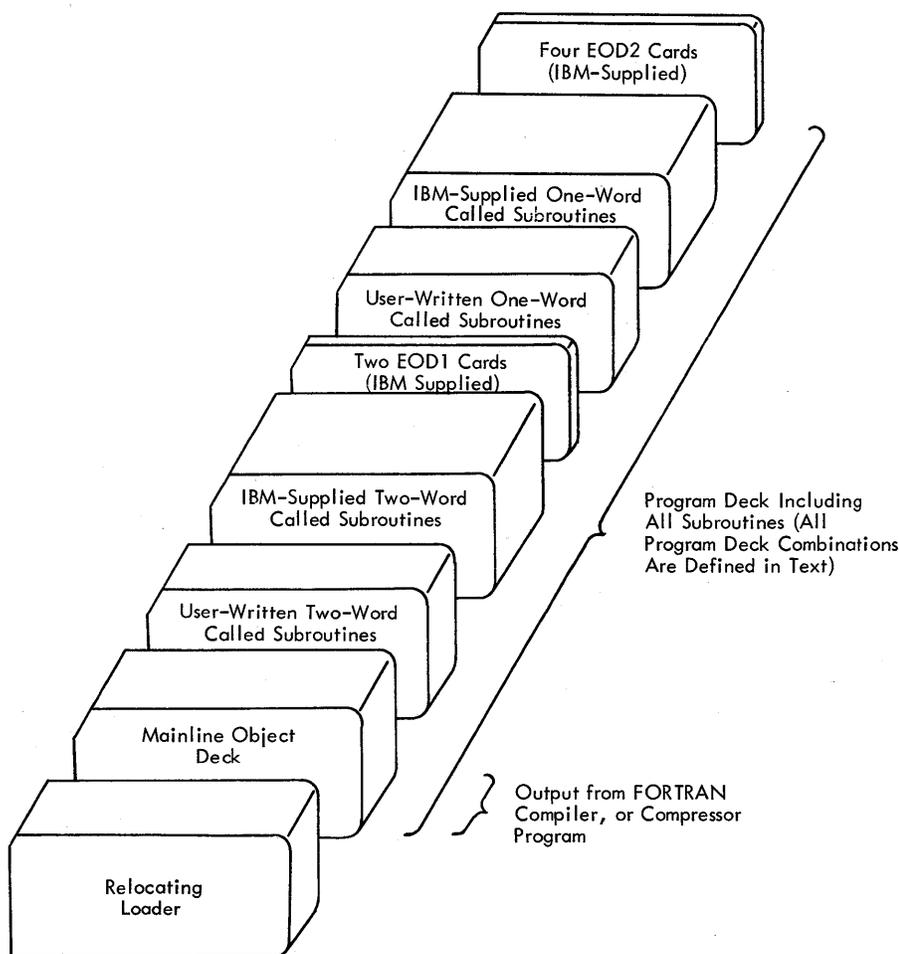


Figure 10. Stacked Input to Load a Relocatable Program

are needed, place the EOD2 cards immediately after the EOD1 cards. The program deck is now ready to be loaded with the relocating loader.

Without Subroutines. Place the mainline object deck in front of the EOD1 cards. Immediately after the EOD1 cards, place the EOD2 cards (if no subroutines are used, the EOD1 and EOD2 cards must still be used). The program deck is now ready to be loaded with the relocating loader.

Operating Procedure

Place the relocating loader, followed by the program deck, in the card reader. With the computer in RUN mode:

1. Press IMM STOP and RESET on the console.
2. Press START on the 1442.
3. Press PROGRAM LOAD on the console.

The relocating loader reads in the program deck. To read in the last card of the program deck, press reader START. If two passes are required, the loader loops from 00A1 to 00A3. If only one pass is required, the loader branches to the execution address specified by the mainline program.

If the loader loops, place the program deck minus the mainline object deck in the card reader and press reader START.

Programming Notes

Causes of Multiple Passes. If a subroutine is placed in the wrong part of the program deck, it might be necessary to remove the loader and the mainline program and place the remainder of the deck in the feed hopper for another pass.

If the relocating loader is used and the six loader cards (EOD1 followed by EOD2) are placed incorrectly, but still in two groups (one group of two and

the other of four), a second pass might be required.

If a routine is called but is not in the program deck, the loader acts as though the routine is misplaced in the program deck. The effect of a second pass is to indicate that still another pass is required.

ALD Card. Any two-word call can be placed in storage by a user punched ALD (automatic load) card. This is useful in program debugging with two-word subroutines, such as trace and dump, that are not called by the mainline program.

The ALD card must be placed behind the mainline header card of the mainline deck being loaded by the relocating loader. Only one ALD card is allowed per program deck.

The format of the ALD card is as follows. Columns 1-4 must be blank. The name of the routines to be loaded are punched, left justified, in IBM card code, starting in column 5. Five columns must be allocated for each name, with a blank column following. Nine names can be punched in a card. The starting columns are 5, 11, 17, 23, 29, 35, 41, 47, 53. Comments can start in column 58.

The ALD card may also be placed in mainline object decks being converted to core image format by the core image converter.

CARD SYSTEM CORE IMAGE CONVERTER

The core image converter changes mainline object programs from relocatable or absolute format to core image format. The core image object deck is then suitable for loading with the core image loader. Figure 11 shows the stacked input required for converting a mainline object program to core image format. Two passes of the program deck are required. On the first pass the program deck is preceded by the core image loader followed by the core image converter.

Building the Program Deck

EOD1 and EOD2 Loader Cards. The two EOD1 cards (types C,D) and four EOD2 cards (types C,C,C,E), included as part of the IBM subroutine library, must always be used with the core image converter. The EOD1 cards terminate processing of two-word called

(CALL) subroutines thereby making it possible for the core image converter to process one-word called (LIBF) subroutines and EOD2 cards. The EOD2 cards terminate processing of one-word called (LIBF) subroutines thereby making it possible for the core image converter to prepare for the second pass or to initiate terminating procedures, such as the completion of transfer vector processing and the setup of the interrupt branch addresses.

With the IBM Subroutine Library Deck. Place the mainline object deck in front of the first two-word, user-written, called subroutine. Place the two-word, IBM-supplied, called subroutines plus the EOD1 cards after the last two-word, user-written, called subroutine.

Place all one-word, user-written, called subroutines after the EOD1 cards and before the first one-word, IBM-supplied, called subroutine. Follow the last one-word, IBM-supplied, called subroutine with the EOD2 cards. The program deck to be converted is now ready for loading.

Without the IBM Subroutine Library Deck. Place the mainline object deck in front of the first two-word, user-written, called subroutines. Place the EOD1 cards after the last two-word, user-written, called subroutine. If no two-word called subroutines are needed, place the EOD1 cards immediately after the mainline object deck.

Immediately after the EOD1 cards, place the one-word, user-written, called subroutines, followed by the EOD2 cards. If no one-word called subroutines are needed, place the EOD2 cards immediately after the EOD1 cards. The program deck to be converted is now ready for loading.

Without Subroutines. Place the mainline object deck in front of the EOD1 cards. Immediately after the EOD1 cards, place the EOD2 cards (if no subroutines are used, the EOD1 and EOD2 cards must still be used). The program deck to be converted is now ready for loading.

Operating Procedure

The core image converter requires two passes of the program deck. The first pass is with the core

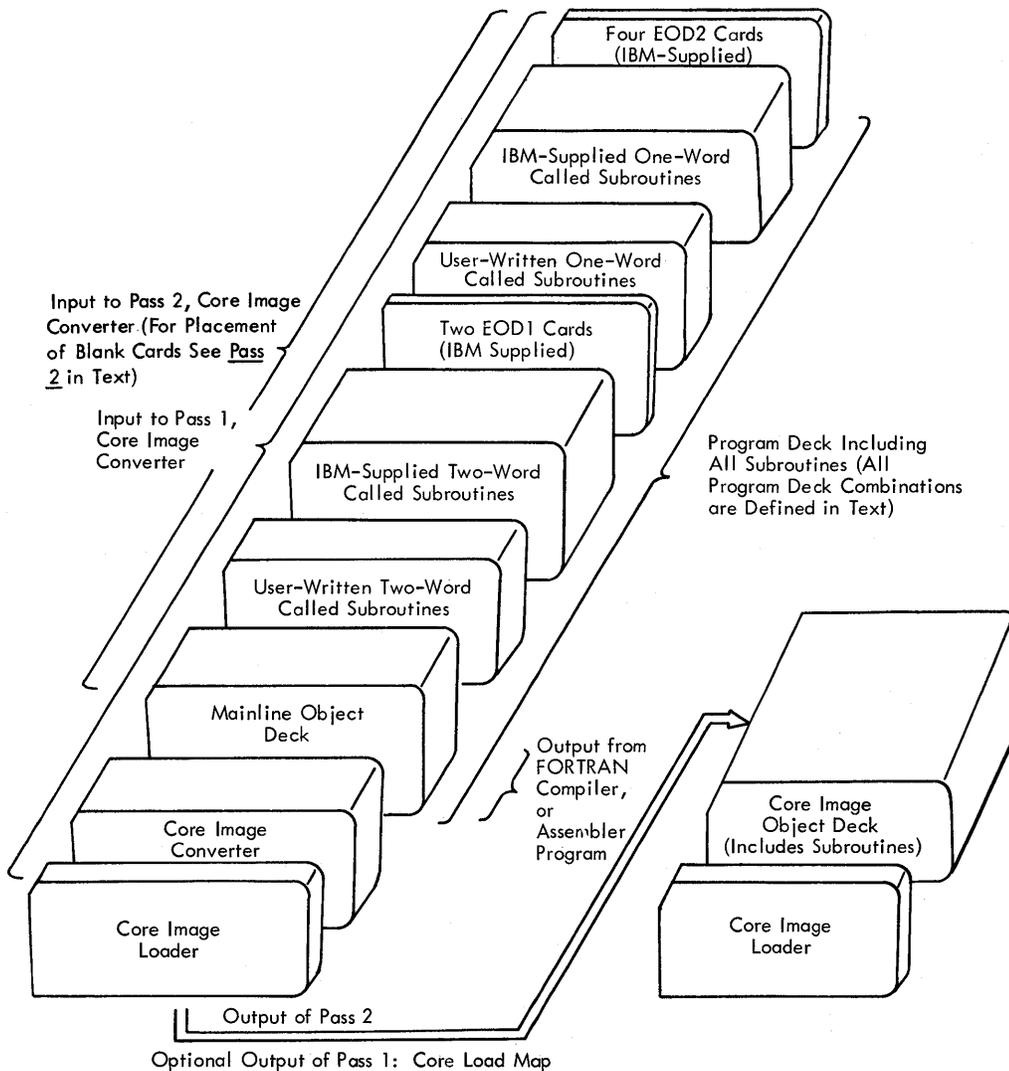


Figure 11. Stacked Input to Produce and Load a Core Image Program

image converter preceded by the core image loader.

Pass 1 creates the transfer vector (with the option to print a core map). Pass 2 creates the core image binary deck.

The core image converter will run correctly with all the loader overlay cards (excluding the EOD1 and EOD2 cards) removed from the program deck. If, in any one program, some overlay cards are removed, all must be removed or a checksum error will occur.

Pass 1. Place the core image loader followed by the core image converter in the card reader. Place the program deck in the reader behind the core image converter. With the computer in RUN mode:

1. Press IMM STOP and RESET on the console.
2. Press START on the 1442.
3. Press PROGRAM LOAD on the console.

After the core image converter is loaded, it waits at location 0D72 if the printout is on the console

printer or at 0DD5 if the printout is on the 1132 Printer. If all Console Entry switches are off and PROGRAM START is pressed, a core map is printed. If any Console Entry switch is on when PROGRAM START is pressed, the map is not printed. See Core Map, Appendix A.

The core image converter reads in and processes the program deck. To read in the last card of the program deck, press START on the card reader. When the first pass is complete, the program waits at location 04A6. Run the last card out of the reader with the NPRO key.

Note that the last card of the program deck remains in the reader only if pass 1 is completed. If pass 1 is not completed, the last card is fed out and the core image converter program loops, waiting for any called routine that has not been loaded. Place the missing routine, or the program deck less the mainline, in the card reader and press reader START.

NOTE: If the missing routine was an ILS, the program waits at 046C. To continue, load the missing ILS followed by the EOD2 cards in the reader. Press reader START and PROGRAM START.

Pass 2. Remove the core image loader and the core image converter from in front of the program deck. The program deck (still including the subroutine library) is now ready to have blank cards inserted. Blank cards can be inserted in the program deck or placed at the end of the deck. If sufficient blank cards are inserted, it is likely that conversion will proceed without further intervention.

Place the program deck, including blank cards in the hopper of the card reader and press reader START. Press PROGRAM START to begin pass 2 of the converter.

If the core storage area for the core-image binary information is too small to contain the entire core-image-converted program, the machine halts at location 0789 before the entire deck has been passed through the reader. At this point, remove the remainder of the program deck from the hopper; run out the cards in the machine by pressing NPRO; place these two cards immediately before the remainder of the program deck; place sufficient blank cards in front of the program deck; and return the deck to the hopper. Press START on the reader and PROGRAM START on the console to continue.

If the program loops at location 0114 (0112 to

0114), the complete program deck has not been read. Place any remaining cards to be processed in the card reader and press reader START. The last card punched is not punched in column 1.

Output Card Formats

The output format of the core image converted deck is as follows.

Data Cards. These cards contain the instructions and data that constitute the core image compressed program. The format (in 16-bit binary code) is:

Word	1	-	Location
	2	-	Checksum
	3	-	Type (left eight bits) 0000 0000; Word count (right eight bits)
	4-54	-	Data Words
	55-60	-	ID and sequence number

Execute Card. A core image execute card is the last card of the program deck to be punched. The format (in 16-bit binary code) is:

Word	1	-	0000 0000 0000 1000
	2	-	Checksum
	3	-	1000 0000 0010 1011
	4-48	-	Data words (interrupt levels, etc.)
	49-52	-	Reserved
	53	-	Midpoint address of TV
	54	-	Execution address
	55-60	-	ID and sequence number

The interrupt level words will be loaded from the execute card.

Programming Notes

An ALD card (see ALD Card, Card System Relocating Loader) can be used to place two-word calls in storage that are not called by the mainline program, e.g., two word subroutines for debugging, such as trace and dump. The ALD card must be placed directly behind the mainline header card of the mainline object deck that is being converted to core image format. Only one ALD card is allowed per program deck.

A relocatable program is assigned the starting address of 00D6. An absolute program is assigned its

absolute address.

Because an illegal CALL statement in an LIBF program will not be detected by the core image converter, it is advisable for the user, as common practice, to run his program first with the relocating loader.

CARD SYSTEM CORE IMAGE LOADER

The core image loader loads programs and subroutines that are in core image format. Programs can be converted from relocatable or absolute format to core image format by the core image converter program (see Card System Core Image Converter). The stacked input arrangement to load a program that has been converted to core image format is shown in Figure 11. Figure 11 also illustrates the input requirements of the core image converter. Note that the subroutines required by the program being loaded by the core image loader are included in the core image object deck. Thus, a program is converted only once. The operation of the core image loader is faster than that of the relocating loader because no relocation is necessary and more words, 51 compared to 45 (see Output Card Formats, Card System Core Image Converter), are included in each compressed card. The core image loader requires approximately 214 words of core storage during the loading of programs. The object program can use locations 0034 through 00D5 as a work area.

Operating Procedure

Place the core image loader followed by the output of the core image converter in the card reader. With the computer in RUN mode:

1. Press IMM STOP and RESET on the console.
2. Press START on the 1442.
3. Press PROGRAM LOAD on the console. When the program loops (location 0073), read in the last card by pressing reader START.

Programming Notes

Since most programs used with the IBM 1130 system are loaded with the core image loader the following should be noted.

If PROGRAM STOP is pressed during the running of any system program that was loaded by the core image loader (and the program has not altered locations 0013, 002D, and 002E), a WAIT will occur at location 002E after all higher priority interrupts have been processed.

To continue processing, set the Mode switch to DISP, press PROGRAM START one time and return the Mode switch to RUN. This will bypass one instruction without executing it. Press PROGRAM START to continue.

PAPER TAPE SYSTEM RELOCATING LOADER

Operating Procedures

1. Place the relocating loader tape in the paper tape reader so that the read starwheels are over one of the frames of delete (7F) code in the leader beyond the program ID. (Each tape should also have a trailer of delete code.)
2. Set the console Mode switch to RUN.
3. Press IMM STOP, RESET, and PROGRAM LOAD on the console.

The relocating loader reads in and then waits at location 0312.

Remove the relocating loader tape and place the mainline tape in the reader. Depress PROGRAM START on the console to load the mainline tape. The program waits at 00E6 for the subroutine library (or EOD1 followed by EOD2) tapes to be read in.

With the IBM Subroutine Library Tapes. The user-written subroutine tapes and IBM subroutine library tapes must now be loaded. The user can facilitate the loading of these tapes through the use of the construct paper tape routine in the utility section of this manual. The construct paper tape routine allows the user to create a single tape containing only those subroutines required for his job.

If the construct paper tape routine has not been used, loading instructions are as follows: If there are two-word, user-written, called subroutines, place each tape, in turn, in the paper tape reader and press PROGRAM START on the console. After the last of these subroutines, the two-word, called IBM subroutine tapes, including EOD1, followed by the one-word, user-written LIBFs and the one-word,

IBM LIBFs including EOD2, should be loaded in a similar manner. Press PROGRAM START to load each tape.

If, at the end of reading in the one-word LIBF IBM subroutines and EOD2, the loader waits at location 00E6, a second pass of the tape is required. However, if the loader waits at location 0222, loading is complete. This WAIT, before going to execution, allows the user to replace the tape in the reader with a data tape. When PROGRAM START is pressed on the console, the loader branches to execute.

Without the IBM Subroutine Library Tapes. Load all two-word user-written called subroutines. Place each tape, in turn, in the paper tape reader and press PROGRAM START on the console. After the last of these subroutines load the IBM EOD1 tape strip (the name EOD1 in the tape leader must be placed beyond the read head). Now load all one-word user-written called subroutines followed by the IBM tape strip marked EOD2 (the name EOD2 in the tape leader must be placed beyond the read head).

Without Subroutines. If no subroutines are used, the EOD1 tape followed by the EOD2 tape must still be used.

The EOD1 and EOD2 tape records must always be used with the relocating loader. The EOD1 records terminate processing of two-word called (CALL) subroutines, thereby making it possible for the relocating loader to recognize and load one-word called (LIBF) subroutines, and to process EOD2 cards. The EOD2 cards terminate processing of one-word called (LIBF) subroutines, thereby making it possible for the relocating loader to initiate terminating procedures, such as the completion of transfer vector processing, and loading of the interrupt branch addresses.

PAPER TAPE SYSTEM CORE IMAGE CONVERTER

The tape loading order is similar to that used with the relocating loader; however, on the first pass the relocating loader should be replaced by the core image loader, followed by the core image converter.

The input tape(s) consists of the mainline program with all the subroutines and EOD1 and EOD2 tapes in the proper order (see Paper Tape System Relocating Loader). If the construct routine is used to combine the input tapes being used with the core image converter, unused routines and type C overlay records (including the type C records in the EOD1 and EOD2 tapes) can be deleted.

Operating Procedure

The core image converter requires two passes of the program tape. The first pass consists of the core image loader tape followed by the core image converter tape. These tapes are then followed by the input tapes described above. The second pass requires the input tapes only (still including the subroutine library tapes, or the EOD1 tape followed by the EOD2 tape). Pass 1 creates the transfer vector, with an option to print a core map. Pass 2 creates the core image binary tape.

Pass 1. Load the core image loader followed by the core image converter. After the core image converter is loaded, the system waits at location 0BA6 if the console printer is used, or at 0C1B if the 1132 Printer is used. Remove the core image converter tape and replace it with the input tape(s). To print a core map, set all Console Entry switches off and press PROGRAM START. If any of the Console Entry switches are on when CONSOLE START is pressed, the map is not printed. (See Core Map, Appendix A.)

The core image converter reads and processes the input tape(s). When the first pass is completed, i. e., all required tapes have been read, the program waits at location 055A with the 1132 Printer, or at 0552 with the console printer.

If Pass 1 is not complete when all input tapes have been read (i. e., a routine other than an ILS is missing), the program waits (reader NOT READY) at location 01E3 with the 1132 printer or at 01D9 with the console printer. Place the tape containing the missing routine, or the input tapes less the mainline (i. e., the subroutine library) in the reader. Press PROGRAM START to continue.

If the missing routine is an ILS, the program loops on itself with 70FF in the SBR. The loop is at 052A with the 1132 and at 0522 with the console printer. To complete Pass 1, the tape containing the missing ILS and the EOD2 tape must be reloaded. Operating instructions are as follows: press IMM STOP, place the required tape in the reader, set the console Mode switch to DISPLAY, press PROGRAM START, set the console Mode switch to RUN, press PROGRAM START.

Pass 2. Before the beginning of Pass 2, ready the punch unit and produce several inches of DELETE code leader by holding down the DELETE and FEED keys simultaneously. Be sure to release the FEED key first. The leader must be punched before starting pass 2 because reading and punching take

place concurrently during the second pass.

Note that it is not necessary to read subroutine tapes more than once in pass 2, even if more than one pass of these tapes was required in pass 1.

Place the input tape(s) (mainline, subroutines, EOD1, and EOD2) in the tape reader. Press PROGRAM START on the console to begin Pass 2.

If the program waits at location 01E3 with the 1132 Printer or at 01D9 with the console printer, the next tape can be loaded. Place the tape in the reader and press PROGRAM START on the console.

If all required tapes have been loaded, the WAIT signifies the end of pass 2. The accumulator contains 9001. The next-to-last word on the output tape is the setting of index register 3, and the last word on the tape (last two frames) is the execute address for the core image programs.

After the last record is punched by the 1055, press DELETE and, while holding it down, press FEED. Allow several inches of delete code to be punched, following the last record. Release FEED and then DELETE.

Record Format

The core image paper tape record is identical, word for word, to the corresponding card system cards, except that ID and sequence numbers are omitted. Each binary record is preceded by a one-frame word count that contains the total number of words in the record. (The word count is not included in the number of words.) Trailing zeros are deleted from tape records.

Delete code (7F) at the beginning and end of the tape and between records is ignored by the core image loader.

Programming Notes

A relocatable program is assigned the starting address of 00D6. An absolute program is assigned its absolute address.

Because an illegal CALL statement in an LIBF program will not be detected by the core image converter, it is advisable for the user, as common practice, to run his program first with the relocating loader.

PAPER TAPE SYSTEM CORE IMAGE LOADER

Operating Procedure

1. Place the core image loader in the tape reader, positioning the tape so that the read starwheels

are over one of the frames of delete code that precede the first record.

2. With the Mode switch set to RUN, press IMM STOP, RESET, and PROGRAM LOAD on the console.
3. Remove the core image loader tape and place the core image object tape in the reader.
4. Press PROGRAM START on the console to load the core image object program.

The core image loader waits at location 00C9 when loading is completed. This WAIT before going to execution allows the user to replace the tape in the reader with a data tape. When PROGRAM START is pressed on the console, the core image loader branches to execute.

UTILITY INPUT/OUTPUT ROUTINE

The IBM 1130 Utility Input/Output Routine accepts data records from either cards or paper tape and provides them as output records on another unit. Specifically, this routine has the following input/output capabilities:

1. Card to console printer
2. Card to 1132 Printer
3. Card to paper tape
4. Paper tape to console printer
5. Paper tape to 1132 Printer
6. Paper tape to card
7. Paper tape to paper tape

Provisions are made for providing output to more than one output unit from one input unit. A summary of outputs available from a single input is shown in Table 5.

The input and output media handled by the utility I/O routine are listed below. (For a detailed description of each code, refer to the publication, IBM 1130 Subroutine Library (Form C26-5929).

Note that because the utility I/O routine uses sub-routines from the subroutine library, the standard pre-operative WAIT for I/O devices not-ready can occur. At the WAIT, the IAR displays 002A.

I/O CHARACTER AND RECORD FORMAT

Card

A character is represented by a card column punched in IBM Card Code (12-bit). A fixed-length record of

Table 5. Utility Input/Output Options

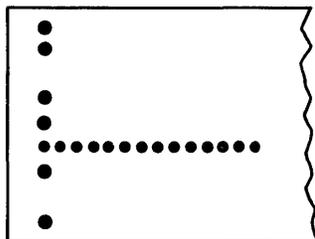
INPUT	OUTPUT			
	CARD	PAPER TAPE	CONSOLE PRINTER	1132 PRINTER
CARD				X
			X	
		X		
		X	X	
		X		X
PAPER TAPE				X
			X	
		X		
	X			
	X			X
	X		X	

output data can occupy several cards if the input tape record contains more characters than there are data positions on one card.

Paper Tape

A character is represented by a frame punched in eight-channel paper tape. Any paper-tape code is acceptable if the output is to paper tape, in which case the output is in the same format as the input. When the output is to cards, 1132 Printer, or console printer, the paper-tape input must be PTTC/8 (perforated tape and transmission code). A typical compressed tape record is illustrated in Appendix H.

PTTC/8 records must be separated by an NL (new line) control character. This character has a hexadecimal value of DD. Its paper tape value is



1132 Printer and Console Printer

A character is represented by one print position. Each record starts at the first print position on a

new line. A record can extend over more than one line.

CONTROL CARD FORMAT

A control card must be used to indicate which input/output operations are required and to convey additional information necessary for the job. Information in each field must be punched, right-justified, with leading zeros. If certain information fields are not applicable, the columns of the fields can be left blank. The control card is divided into three major fields:

1. Input (columns 2-16)
2. Output (columns 18-29)
3. Additional print device (columns 30-31)

The format of the control card is given in Table 6.

USER'S EXIT OPTION – CARD AND PAPER TAPE

The input-output routine provides an exit to allow a user-written subroutine to be executed in combination with the I/O routine. This facility allows, for example, sequence checking of input cards.

To exercise the user's exit option, follow the procedure listed below:

1. Write the subroutine in the 1130 Assembler language.
2. Assemble it absolute, with the origin at hexadecimal address 0DB0. The operand of the END source statement must be 050B. (This is the return address in the main program where execution resumes after the user's program is loaded.)
3. Compress the subroutine, using the 1130 Compressor Program.
4. Remove the first card of the compressed binary deck and replace it with the four special data overlay cards provided with the input/output routine.
5. Place the special end-of-deck card (also furnished with the I/O routine) behind the last card of the binary deck.
6. Place the entire deck immediately behind the control card (see Figure 12).

For paper-tape users, the four special data overlay records are furnished in one tape strip and

Table 6. Control Card Format

Columns	Description	I/O Device Applicable
1	Asterisk (used for control card identification)	
2	Input file unit C-card P-paper tape	C, P
3	Blank	
4	Record Format V-record length is variable. Blank-record length is fixed If column 2 is C, column 4 must be V or blank or a format error will occur. If column 2 is P, column 4 is not checked.	C, P
5-7	Reserved	
8-11	Data positions (input cards). Columns 8-9 specify the first column of data in card (blank is interpreted as 01). Columns 10-11 specify the last column of data in card (blank is interpreted as 80).	C
12-16	Reserved	
17	User's exit. Letter X - the user has added his own subroutine to the program for processing input data records.	C, P
18	Output file unit. C-card P-paper tape T-Console Printer N-1132 printer A console printer or 1132 Printer can be used in addition to card or paper tape output if specified in column 30.	C, P, T, N
20-23	Card output data positions. 20-21 specify the first column of data in card (blank is interpreted as 01). 22-23 specify the last column of data in card (blank is interpreted as 80).	C
24	Sequence number option. Letter X - a sequence number is to be generated in cards. Space must be provided in the cards for punching the sequence number; therefore, data positions (see columns 20-23) should not use all 80 columns of the card when punching sequence numbers. A 4-digit sequence number is punched in columns 77-80.	C
25-29	Reserved	
30	Additional output unit. This column can specify console printer or 1132 Printer output in addition to card or paper tape specified by column 18. T-Console Printer N-1132 printer	T, N
31	Blank	
32-80	Not used.	T, N

the special end-of-deck record is in another tape strip. Steps 4 and 5 above can be implemented by deleting the first record of the compressed binary tape (refer to Compressor Tape Format), and then punching an entirely new tape consisting of the data overlay records, the user's program (less first record), and, finally, the special end-of-deck record. During execution, the program waits for

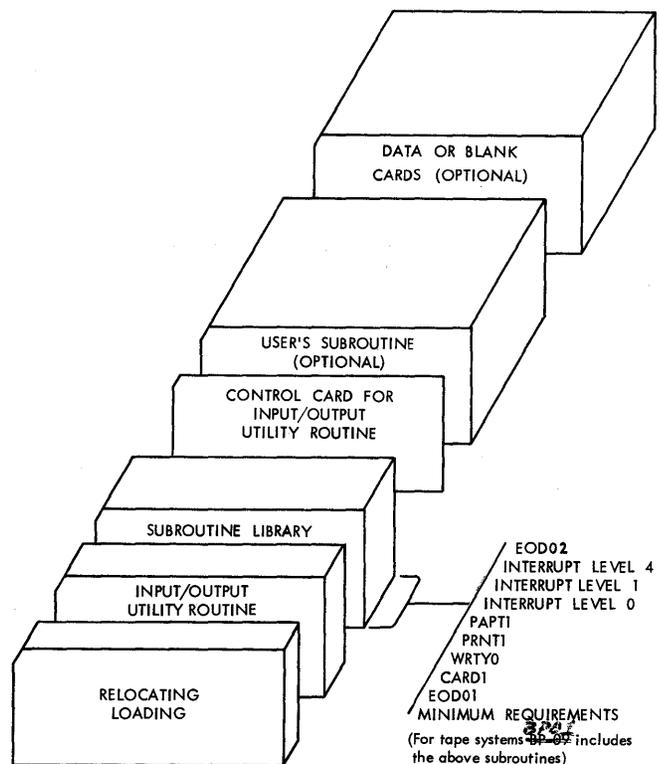


Figure 12. Stacked Input Arrangement for Utility Input/Output Operations

the operator to remove the I/O utility tape from the reader and load the user's program tape.

The first word of the user's subroutine must be reserved for a link word and followed by the first instruction to be executed.

During execution, the link word contains the address of the word that specifies the address of the input buffer, which is 80 words in length for a card input. For paper tape it is 512 words (1024 characters) in length for 4K systems and 1288 words (2576 characters) in length for 8K systems. A paper tape stop character appears at the end of the paper tape record. The link word must be incremented by two to obtain the return address to continue processing. However, if the user does not wish to print or punch the record currently being processed, he should increment the link word by one. The user cannot call other library subroutines in his program.

At execution time, the control card is read and analyzed. If a user's exit is specified in column 17, the I/O routine branches to the loader program and loads the user's subroutine. Control is then returned to the I/O routine and execution is continued.

After an input record is read into the buffer area, the program branches to the user's subroutine. The subroutine function is performed, and the program branches back to the I/O routine through the link word.

The user's subroutine cannot exceed 530 words for a 4096-word system, or 2000 words for an 8192-word system, nor can it alter the length of the record in any way; however, the data contained within the record may be changed or rearranged. Index registers 1 and 2 may be used; however, if index register 3 is used, its contents must be saved and restored. The user's exit cannot be used for a paper-tape to paper-tape job.

I/O ROUTINE ERROR CHECKS

The input/output routine checks the control card and the data being processed to determine the validity of the operation. Errors are indicated by a message on the console printer. Error messages are shown in Table 7.

OPERATING INSTRUCTIONS - CARD AND PAPER TAPE

The input/output routine should be arranged for loading as shown in Figure 12. If the input is from paper tape, load the tapes in the same sequence. An abbreviated subroutine library for use with the I/O routine should be arranged as follows:

- EOD01
- CARD1
- WRTY0
- PRNT1
- PAPT1
- ILS00
- ILS01
- ILS04
- EOD02

The relocating loader, I/O routines, and abbreviated subroutine library can then be retained as a unit for more convenient use.

1. Place the deck in the card read punch hopper.
2. Press IMM STOP and RESET on the console.
3. Press START on the card read punch.
4. Press PROGRAM LOAD on the console.

In the paper tape system, the program waits for the operator to change tapes. For a job requiring paper tape input, the program waits when there is no more tape to be read. (The stopping address is either 0029 or 04CD.) For card output a JOB COMPLETE message is typed on the console printer. If the input is paper tape and the output is cards, blank cards must be placed in the card read punch and the unit made ready prior to program load.

DUMP ROUTINES

These routines can dump all or any part of core storage on the various output units connected to the system. Normally, a user's program and/or data is already in core. The desired dump routine is then loaded into core to perform the dump.

Dump between limits is permitted on the card read punch and the console printer.

Each dump routine is contained in a separate deck (or paper tape). Only a dump between limits on the console printer is provided for the paper tape system.

Dump Between Limits

The starting and stopping addresses of the dump are entered in hexadecimal form through the Console Entry switches. Any area of core can be dumped. The DUMP routine is located in (approximately) the first 520 words of core.

The output format of the dump is hexadecimal or decimal, depending upon the setting of the Console Entry switches. In hexadecimal form, each 16-bit word is represented by a four-digit number. In decimal format, each word is represented as a five-digit number, preceded by a plus or minus sign (six characters, total). In either case, the address (location) of the first word in each line is always in hexadecimal form.

Each line of output consists of a four-digit hexadecimal address followed by 8 five-digit decimal or 8 four-digit hexadecimal words, with one space separating each word. This format is used for the card and console printer output devices.

Table 7. I/O Routine Error Messages

Error Message	Description of Error & Associated Column(s) of Control Card	Restart Procedure
ERR 00	No control card (1)	Insert control card and press START.
ERR 01	Input and/or output unit error (2, 18, 30)	Insert corrected control card and press START.
ERR 02	Record format error (4)	Insert corrected control card and press START.
ERR 04	Data position error (8-11, 20-23) and sequence number specification error (24).	Insert corrected control card and press START.

Operating Procedure

Card Loading Procedure

The core-image loader (see Core Image Loader section) must be placed in front of the program deck.

1. Place the cards in the card read punch hopper.
2. Press IMM STOP and RESET on the console.
3. Press START on the card read punch.
4. Press PROGRAM LOAD on the console.

Paper Tape Loading Procedure

1. Place the core image loader tape in the reader so that one of the delete codes beyond the program ID in the leader is under the starwheels.
2. Press IMM STOP, RESET, and PROGRAM LOAD on the console.
3. After this core image loader has been read in, place the dump routine tape in the reader.
4. Press PROGRAM START.

In the paper tape system, the paper-tape core image loader halts before execution of the dump program. The user must press PROGRAM START to begin execution.

Card and Paper Tape Operation

- The program starts and then stops to allow entry of the starting address through the Console Entry switches. This address must be entered in hexadecimal form.
- After the starting address is entered, press PROGRAM START. The program starts and then immediately stops again — this time to allow entry of the stopping address of the dump.
- After the stop address has been entered, press PROGRAM START. The program starts and then immediately stops again; the Console Entry switches must be set to indicate whether hexadecimal or decimal output is required. If all the switches are off, output will be hexadecimal form; if any are on, it will be in decimal form.

The dump proceeds until the stopping address is reached; then the program stops. If another dump is required, press PROGRAM START. The program again makes stops to allow entry of the new start and stop addresses and output format.

If a stopping address less than the starting address is inserted, an error code, FFFF (all bits on), appears in the accumulator and the program again makes stops to accept the correct start and stop addresses.

If execution is manually stopped by the user before reaching the stop address, and it is desired to dump from a new set of limits:

1. Press IMM STOP and RESET on the console.
2. Turn the Mode switch to LOAD.
3. Set the Console Entry switches to 00C9 and press LOAD LAR.
4. Turn the Mode switch to RUN and press PROGRAM START.

1132 Printer Core Dumps

Full Dump. This is a self-loading, four-card routine that dumps the contents of core storage in hexadecimal format on the 1132 Printer (the fourth card is blank). The routine is also available in paper tape.

Dumping begins at hexadecimal address 00A0 and continues to the end of core. Sixteen words per line are printed, preceded by a four-digit hexadecimal address of the first word of each line.

Selective Dump. This is a self loading six-card routine that dumps the contents of core in hexadecimal format on the 1132 printer (the sixth card is blank). This routine is available in card form only.

Dumping begins at hexadecimal address 00F0 and continues to the end of core. Each line contains a 4-digit hexadecimal address, followed by 16 words. Each 16-bit word is represented by a 4-digit hexadecimal number. The address and each word is set off by a space. An additional space is taken after each group of four words on a line.

In order to decrease the dump time, the routine does not print consecutive duplicate lines. Before a line is printed, the 16 words are compared with the 16 words printed on the previous line. If the contents are identical, (for example: all zeros) the routine goes on to the next 16 words in core. When the data does not compare, the printer spaces one line and prints. The address printed is the location of the first word printed on the line.

Card Operating Procedure

1. Ready the 1132 Printer
2. Place the dump routine deck in the 1442 Card Read Punch hopper.

3. Press IMM STOP and RESET on the console.
4. Press START on the 1442.
5. Press PROGRAM LOAD on the console.

Dumping continues until the last 16 words of core are addressed and printed.

The program does not skip to the top of a new page to start, nor is page numbering or page overflow provided.

Paper Tape Operating Procedure

1. Ready the 1132 Printer.
2. Place the Dump from 00A0 tape in the paper tape reader so that one of the delete codes beyond the program ID in the leader is beneath the starwheels.
3. Press IMM STOP, RESET, and PROGRAM LOAD on the console.

The output format is the same as described for the card routine.

KEYBOARD ROUTINE

This routine is designed to assist the user in preparing source documents on cards or paper tape with the keyboard input.

Two facilities are provided by this routine:

1. Keyboard to paper tape
2. Keyboard to cards

Data is entered at the keyboard and is output to paper tape or cards depending on the setting of the Console Entry switches.

Format

The size of the record for card output is limited to 80 characters. The maximum record length for paper tape output is 3,648 characters (1,824 words). Cards are punched in IBM card code (12 bit). The output code for paper tape is PTTC/8.

Operating Procedure

Card Loading Procedure

Place the core image loader in front of the program deck and place the cards in the read punch hopper

(if the output is cards, blank cards must follow the program deck).

1. Press IMM STOP and RESET on the console.
2. Press START on the card read punch.
3. Press PROGRAM LOAD on the console.

Paper Tape Loading Procedure

1. Place the core image loader tape in the reader so that a delete code beyond the program ID in the leader is beneath the starwheels.
2. Press IMM STOP, RESET, and PROGRAM LOAD on the console.
3. After the core image loader has been read in, place the keyboard routine tape in the reader.
4. Press PROGRAM START.

In the paper tape system, the paper tape core image loader halts before execution. The user must press PROGRAM START to begin execution.

Card and Paper Tape Operation

After the keyboard utility program is loaded, it comes to a wait to allow the operator to select the type of output desired. If all Console Entry switches are off, the output will be on cards; if any switch is on, the output will be on paper tape. After setting the Console Entry switches, press PROGRAM START. The keyboard is now unlocked to allow entry of the user's message, and the stored program stays in a loop. As each key is pressed, the program prints the appropriate character on the console printer.

When all data has been entered, press EOF. A paper tape record or card is punched, depending upon the output selected. After this operation, the keyboard is again unlocked, awaiting the next record.

Typing errors can be corrected by pressing BACKSPACE each time an error is made. If an entire record is to be erased, press ERASE FIELD.

Erase Field

When the Erase Field key is pressed the routine prints two slashes on the console printer, restores the carrier to a new line, and prepares to replace the old message in the I/O area with the new message. The operator then enters the new message.

Backspace

When the backspace key is pressed, the last graphic character entered is slashed and the address of the next character to be read is decremented by +1. If the backspace key is pressed twice, consecutively, the character address is decremented by +2, but only the last graphic character is slashed. For example, assume that ABCDE has been entered and the backspace key pressed three times. The next graphic character replaces the C, but only the E is slashed. If the character F had been used for replacement the paper would show ABCDE/FFF, but ABFFF would be stored in the buffer.

CARD REPRODUCING ROUTINE

Since the 1130 Card Assembler punches data into the user's source cards, a utility routine to reproduce cards is useful to customers who do not have off-line card-reproducing facilities.

Format

A header card is placed in front of the deck to be reproduced. A column of the header card that has any character punched will cause the corresponding column to be reproduced from the master deck into the blank deck; blank columns in the header card will not be reproduced from the master deck; however, blank columns appearing between the first and last data column punched in the header card will occupy space in the buffer.

For example, if column 1-20 and 36-49 are to be reproduced, the header card must be punched with any character in columns 1-20 and 36-49; columns 21-35 and 50-80 must be blank.

The routine reads the cards to be reproduced into a buffer area in core until it is filled. In computing the number of cards that can be reproduced from a buffer load, remember that embedded blanks must be counted. Cards are then punched from the buffer and the next batch of cards is read in.

The number of cards the program is capable of reproducing from each buffer load depends on the number of columns to be reproduced (as indicated by the header card).

The size of the buffer is the number of words in core storage, less 596. One word is required for each column punched. Thus, if all 80 columns are to be punched on a 4K system, the maximum number of cards that could be punched from each buffer load is approximately 43 (3500 divided by 80). In an 8K sys-

tem, approximately 94 eighty-column cards can be punched from each buffer load.

Operating Procedure

Place the core-image loader in front of the card-reproducing program deck. Behind these cards, place the header card followed by the deck to be reproduced. Place the entire deck in the 1442 Card Read Punch hopper.

1. Turn the Mode switch on the console to RUN.
2. Press IMM STOP and RESET on the console.
3. Press START on the card read punch.
4. Press PROGRAM LOAD on the console.

During the input phase of the routine, the program reads in enough cards to fill the memory buffer. The loader and program cards are selected into stacker 1, and the cards to be reproduced are selected into stacker 2. When the buffer area is full, the processor waits.

Press the NPRO key, remove from the stacker the two cards which are run out, and place them in front of the deck still in the hopper. These cards represent the first two cards of the next buffer load and will be processed during the next run. When the input cards do not fill the buffer and the reader stops with the hopper empty, the program will loop. Press reader START to complete the input phase.

To start the output phase of the routine, place sufficient blank cards to reproduce the contents of the buffer in the hopper and press reader START and PROGRAM START. The reader punch punches cards from the data stored in the buffer area and selects them into stacker 1. When the punch stops, press the NPRO key and remove all punched cards from stacker 1. (The non-process run-out cards will be blank.)

The portion of the original deck that was set aside can now be loaded into the card read punch hopper and the input phase of the program repeated.

When the input cards do not fill the buffer area, the program makes a last card check. When the card read punch stops with the hopper empty, and the program in a loop, press reader START to process the last two cards.

Now return to the output phase as described above.

CONSOLE ROUTINE

This routine aids the user in the debugging of programs. The programmer can dump portions of core

by loading a single-card console routine which occupies the first 80 words of core. The output device is the console printer. This program is available for the card system only.

Format

This routine dumps core in hexadecimal form, starting with the word specified in the Console Entry switches. Dumping continues until PROGRAM STOP is pressed.

Words are dumped in four-digit hexadecimal form, with a space between each word and the next one. The first word typed is the starting address of the dump. The number of characters per line depends upon the margin settings of the console printer.

Operating Procedure

1. Set the Console Entry switches to the address at which dumping is to start.
2. Place the program card or tape in the appropriate reader and set the Mode Switch to RUN.
3. Press IMM STOP and RESET, reader START (if card system), and console PROGRAM LOAD.

Dumping proceeds until PROGRAM STOP is pressed. Press PROGRAM START to resume the dump.

DISK PACK INITIALIZATION ROUTINE (DPIR)

The DPIR performs the following functions:

1. Writes disk sector addresses on all cylinders.
2. Determines which, if any, sectors are defective and writes the address(es) of the cylinders containing the defective sectors on sector 0000.
3. Establishes a file-protected area for the disk pack.
4. Puts an ID on the disk pack.

The 1130 Disk Routines operate effectively with up to three cylinders containing defective sectors. An attempt to read or write a defective sector that is not identified in sector 0000 results in a read or write error after the operation has been attempted 10 times.

At the completion of DPIR, an eight-word table is written on sector 0000. (If sector 0000 is defective, DPIR will not operate properly.) The first word (word 0) of the table contains the sector address 0000. Words one, two, and three contain the first sector address of any defective cylinders found (maximum of three). When there is no defective cylinder, these words contain 0658. Word 4 contains the ad-

dress of the first non-file-protected sector for that disk pack. Words five, six, and seven contain a five character ID name in packed EBCDIC. Words five and six contain two characters per word, and word seven contains an EBCDIC character in the left half of the word and an EBCDIC blank in the right half of the word.

To determine which sectors are defective, the user can dump core upon completion of execution; the defective sector table starts at location 0739 and the actual count of defective sectors is in location 035E.

In order to get the defective cylinder and file-protect address into the disk subroutines, the user can execute a CALL to SPIR (refer to "Set Pack Initialization Routine" IBM 1130 Subroutine Library, Form C26-5929). An SPIR subroutine should be run for every core load that uses the disk subroutines and every time the disk pack is changed.

Table 8 lists the DPIR halt addresses.

DPIR Card Load Operating Procedures

The procedure for loading and executing the DPIR is as follows:

1. Load the disk pack.
2. Put the relocating loader in the card hopper.

● Table 8. DPIR Halt Addresses

Halt Address (Hexadecimal)	Meaning	Action Required
0346	Any disk read/write error during writing of sector IDs.	FE intervention is required; reload the routine or branch to 02EE manually.
03D2	The disk is not ready.	Make the disk ready and restart the program.
0367	Sector 0000 is defective; the sector addresses have been written on the disk, but the table has not been written on sector 0000.	
03BE	The routine has run successfully and no defective sectors were found.	
03BA	The routine has run successfully and three or less defective sectors were found.	
03C8	The routine has run successfully, but more than three defective sectors were found.	
038A	The routine is in WAIT.	Enter desired file-protect address in the Console Entry switches; press PROGRAM START on the console.
03EA	A seek failure or read failure occurred during a scan of the disk.	

3. Follow the loader with the IBM-supplied DPIR deck.
4. Follow the DPIR deck with the subroutine library.
5. Set the console Mode switch to RUN.
6. Press IMM STOP, and RESET on the console.
7. Press reader START.
8. Press PROGRAM LOAD on the console.

An abbreviated subroutine library for use with the I/O routine should be arranged as follows:

```
EOD1  
TYPE0  
HOLL  
PRTY  
ILS04  
EOD2
```

Therefore, if the program is converted to core image, verify that the latest versions of these routines are used.

The DPIR is read in and the keyboard is selected. (The keyboard Select light comes on.) Wait for the File Ready light to come on and then:

1. Enter the five-character ID to be written on the disk pack. If the ID is less than five characters, left-justify by following the ID with spaces. When the fifth character is entered, the program branches to execute. The disk surface is now cleared and the sector addresses are written. The routine waits at 038A.
2. Set the Console Entry switches to the file protect address to be written on the disk pack.
3. Press PROGRAM START. The defective sector and file protect address data is written on sector 0000. A scan of the disk is now performed to check for seek failures. If a seek or read failure occurs, the routine waits at 03EA. Other DPIR halt addresses are described in Table 8.

DPIR Paper Tape Load Operating Procedure

The procedure for loading and executing the DPIR is as follows:

1. Load the disk pack.
2. Put the relocating loader tape in the reader; position one of the delete codes that appear after the program name in the leader under the read starwheels.
3. Press IMM STOP, RESET, and PROGRAM LOAD on the console.
4. When the loader reads in and waits, load the DPIR tape.
5. Press PROGRAM START on the console.
6. When the DPIR tape reads in, load the BP01 subroutine library tape and press PROGRAM START.

From this point on, the operation is identical to the card load.

CONSTRUCT PAPER TAPE ROUTINE

This routine allows the user to

1. Combine tapes
2. Delete subroutines that are not required from the subroutine library tape
3. Delete type C overlay records from a library tape for input to the core image converter
4. Reproduce tape if the input tape is in a specified format
5. List routine names from input tape on console printer

These options can be used in combination and are controlled by the settings of Console Entry switches 0, 1, 13, 14, and 15.

An abbreviated subroutine library for use with the I/O routine should be arranged as follows:

```
EOD1  
WRTY0  
PAPEB  
EBPRT  
EBPA  
PRTY  
ILS04  
EOD2
```

Therefore, if the program is converted to core image, verify that the latest versions of these routines are used.

Operating Instructions

The construct routine is loaded with the relocating loader:

1. Ready the paper tape punch.
2. Place the relocating loader tape in the reader so that one of the DEL codes in the leader beyond the program ID is under the read starwheels.
3. Put the Console Mode switch in RUN position.
4. Press IMM STOP, RESET and PROGRAM LOAD.
5. The relocating loader program will be read in and WAIT.
6. Load the construct tape followed by the BP01 subroutine library tape. Press PROGRAM START to load each tape.
7. Press PROGRAM START. The program will execute and WAIT at location 0029 with FFFF in the accumulator. A leader of DEL code for the new tape will be punched on the paper tape punch.

Reproduce Option

To reproduce a program:

1. Load the input tape in the reader.
2. Set Console Entry switches as follows:
 - CES 0 ON - To reproduce the program and wait at location 41 with FFFF in the accumulator, after EOP record has been punched
 - CES 13 ON - To print name of program on console printer
 - CES 15 ON - To delete type C (overlay) records
3. Press PROGRAM START

The input tape is assumed to contain records in the format:

Any number of DEL characters
 One character word count (n)
 2 n characters, equaling words 1 through n of a compressed format card

Only the following types of programs should appear on the input tape:

Program (types 1 and 2) - printed as bbb*****b
 Subroutine (types 3 and 4) - printed as bbbxxxxxb for each name (xxxxx) on the header record.
 ISS (types 5 and 6) - printed as bbbxxxxxb for the name (xxxxx) on the header record
 ILS (type 7) - printed as bbbILS0xb for the level (x) specified on the header record

Any ISS will set the appropriate ILS-needed indicators in the CONSTRUCT program (see Subroutine Library Option).

Name Tape Option

The keyboard utility routine must be used to create a name tape. The name tape is used as input to the construct routine if the user wants to add or delete subroutines. The user can elect to enter the names of the subroutines required for a job or the names of the subroutines not required. The setting of Console Entry switch 14 in the Subroutine Library Option will determine whether named or unnamed programs are deleted from the tape. To create a name tape:

1. Load the keyboard utility routine.
2. Create a leader of DEL code on the paper tape punch.
3. Set the Console Entry switches to a non-zero value.
4. Key in each name, followed by an EOF character. All names contain five alphameric characters (xxxxx, xxxxb, xxxbb), i.e., expand all names to five characters by adding blanks /b/bb/. Only characters A-Z and 0-9 will be printed correctly.
5. After all names have been entered, depress the EOF key a second time so that the tape is as follows:

```

          D      D      D D
Name D NAME D .... NAME D D

```

6. Create a trailer of DEL code for the name tape.

Do not key in the numbers of the ILS required for the programs to be punched later. The CONSTRUCT program takes care of this (see Subroutine Library Option).

To read the name tape:

CES 0 OFF, CES 1 ON - To read name tape and wait at location 41 with FFFF in the accumulator, after EOF (DD) record encountered with no preceding data.

Set the Console Entry switches as indicated, load the name tape in the reader, and press PROGRAM START.

Subroutine Library Option

The input tape is assumed to have the same format as the reproduce option.

In addition to the programs accepted by the reproduce option, the EOD1 and EOD2 records can appear on the input tape.

EOD1 (type D) - printed as bbbbEOD1b
 EOD2 (type E) - printed as bbbbEOD2b

To process a library tape, load a subroutine library tape in the reader and use the following Console Entry switch settings:

CES 0 OFF, CES 1 OFF - To process subroutine library tape and wait at location 41

- CES 13 ON - with FFFF in the accumulator after EOD1 or EOD2 record has been punched. To print names of programs
- CES 14 ON - To delete named programs
- OFF - To delete unnamed programs
- CES 15 ON - To delete type C (overlay) records

Place the tape in the reader and press PROGRAM START.

The processing for each program is as follows:

- Program -- deleted from the output tape
- Subprogram -- deleted or reproduced according to CES 14 and the last name tape (if any) read
- ISS -- deleted or reproduced according to CES 14 and the last name tape (if any) read
- ILS -- reproduced only if specified by an ISS already reproduced under either the reproduce option or subroutine library option
- EOD1 -- reproduced
- EOD2 -- reproduced and followed by a trailer of DEL characters * (See below)

If the console printer option is selected each name printed will be on a separate line. If the program appears on the output tape, the name will be preceded by 3 or 4 blanks; otherwise the second blank will be replaced by a D - bbbxxxxx or bDbxxxxx.

The subroutine library option does not clear ILS-needed indicators or name tape names when the subprogram or subroutine is encountered.

If any one of the entry points in a subprogram match a name from the name tape, all are assumed to match.

Additional Ways to Use the CONSTRUCT Program

- To list a strange tape, set CES 0, 1, 14 OFF and CES 13 ON
- To reproduce a tape with no mainline programs, set CES 0 and 1 OFF and CES 14 ON

OPERATING NOTES

The standard not-ready halts at location 41 may occur on preoperative checks. Ready the indicated

** NOTE: If the delete C option is selected, the EOD1 & EOD2 tapes are specially handled. The type D & type E records are rechecked summed to prevent core image convertor checksum waits*

device and press PROGRAM START on the console.

- 2000 - Console printer not ready
- 3000 - Paper tape reader or punch not ready

While processing paper tape the program may halt at location 0301 if the reader or punch becomes not ready. To continue ready the indicated device and press PROGRAM START on the console. The accumulator specifies the device:

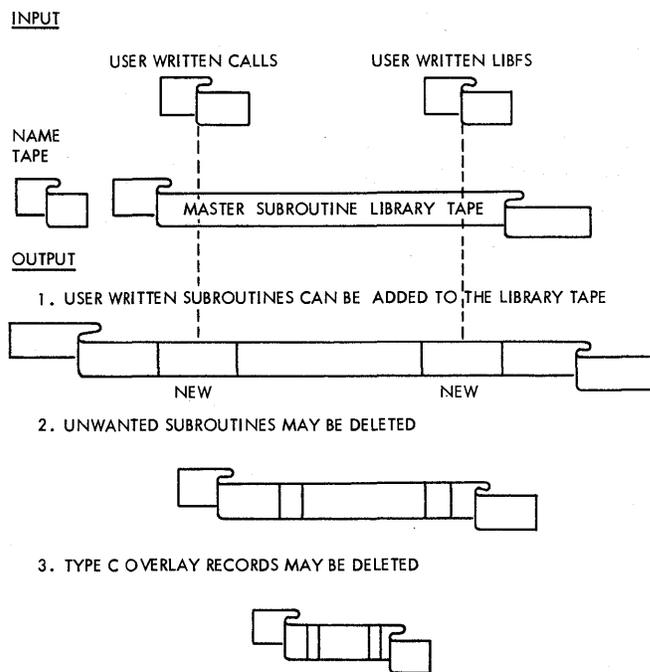
- 0004 - Paper tape punch not ready
- 0005 - Paper tape reader not ready

To restart the program:

1. Press IMM STOP on the console.
2. Set the console Mode switch to LOAD.
3. Set 0306 in the Console Entry switches.
4. Press PROGRAM START.
5. Set the console Mode switch to RUN.
6. Press PROGRAM START.

A new leader of delete codes will be punched and the ILS-needed indicators will be cleared.

The following illustration shows some of the construct routine uses:



APPENDIX A. CORE MAP

A sample Core Map (Table A1) for a 4K Machine is shown below (all numbers in hexadecimal). The left-most column displays the Relocation Factors. The second column displays the effective core size of the program. Starting in the third column are the corresponding entry point names that have been referenced. The relocation factor for an absolute mainline is zero because the locations of the instructions are dependent only on the source code. For a relocatable mainline program or subroutine, the Relocation Factor is the first location available for loading of the program. Following the list of programs is a cross reference table. This table associates each two-word-called entry point with an

actual core address. The next table listed is the Transfer Vector Summary.

NOTE: If there are no two-word-called entry points, title information is printed.

The transfer Vector Summary displays all entry points in the TV along with their associated core addresses.

IND 0000 FAC 0000 printed as the last entry points in the TV means that the Error Indicator and Floating Accumulator are zeroed out. ILS in the list of program entry points means that an Interrupt Level Subroutine is located there.

The last three lines printed are self explanatory.

00D6	0020	MAINLINE									
00F6	0020	VARI									
0116	0010	TTEST									
0126	0008	STOP									
012E	0322	VIOF	VCOMP	VWRT	VFIO						
0450	0032	WRTYZ									
0482	0010	GETAD									
0492	0036	EBCTB									
04C8	000A	FLOAT									
04D2	0026	IFIX									
04F8	002E	ELD	ESTO								
0526	001E	NORM									
TWO WORD CALLED ENTRY POINTS											
TV SUMMARY											
NORM	0526	EBCTB	04C5	GETAD	0482	IFIX	04D2	FLOAT	04C8	VCOMP	0131
VIOF	013E	VWRT	015B	TTEST	0116	ESTO	04FC	STOP	0126	VARI	00F6
ELD	0512	WRTYZ	0454	VFIO	0221	IND	0000	FAC	0000		
LOWEST ADDRESS OF A TV ENTRY IS 0FCD INDEX REGISTER 3 IS 0F80 HIGHEST ADDRESS OF A TV ENTRY IS 0FFF											

APPENDIX B. CORE REQUIREMENTS AND PROGRAM TIMES

CORE REQUIREMENTS

The core requirements listed below for the assembler and the FORTRAN compiler are for the programs themselves. Actually, because of the symbol table and the statement string, all of core is required for assembly or compilation.

ASSEMBLER AND COMPRESSOR

The card assembler occupies core locations 008C to 099A. The card compressor occupies locations 0032 to 0791.

The paper tape assembler occupies core locations 0000 to 09B6. The paper tape compressor occupies locations 0000 to 070C.

FORTRAN

The FORTRAN card compiler occupies core locations 0000 to 0976. The FORTRAN paper tape compiler occupies locations 0000 to 0940.

UTILITIES

The relocating loader initially occupies

Cards -- 0000 to 04F2
Paper tape -- 0000 to 05EA

Other programs can be loaded as low as 027C. Relocatable mainline programs are loaded starting at location 027C. The core image loader can start loading at

Cards -- 00D6
Paper tape -- 00D6

SUBROUTINES

Subroutine core requirements are shown in the publication IBM 1130 Subroutine Library (Form C26-5929).

PROGRAM TIMES

ASSEMBLER AND COMPRESSOR

Card System

The assembly/compression rate for a 300 cpm reader is 66-77 statements per minute. The time required to load the assembler deck is approximately 12 sec., the compressor deck, 9 sec. The assembly/compression rate for a 400 cpm reader is 92-102 statements per minute. The time required to load the assembler deck is approximately 9 sec., the compressor deck, 6 sec.

Variations in the assembly/compression rate can be attributed to variations in the size of the object decks produced, varying number of comments, etc.

Paper Tape System

The time required to load the assembler tape is approximately 1.4 min., the compressor tape, 1.0 min. For a tape record length of 19 characters/statement, the assembly/compression rate is 16 statements/min. For a tape record length of 26 characters/statement, the assembly/compression rate is 13 statements/min.

Variations in the assembly/compression rate can be attributed to variations in the size of the object tapes produced, varying number of comments, etc.

FORTRAN

Card System

The average compilation times on the IBM 1130 FORTRAN Card System can be calculated as follows: (Assumed: 400 CPM 1442 Read Punch)

Complete 1132 listing:

$$1.5 \times SC + 90 + CC = \text{Running time in seconds}$$

No listing:

$$.45 \times SC + 87 + .2CC = \text{Running time in seconds}$$

where

SC = Number of non-comment source cards

CC = Number of comment source cards

NOTE: For source programs larger than 200 statements, the running time should be increased by 10%.

Paper Tape System

The average compilation times for the FORTRAN paper tape system can be calculated as follows:

Complete 1132 listing:

$$3 \times SC + 80 + CC = \text{Running time in seconds}$$

No listing:

$$2.25 \times SC + 870 + .5CC = \text{Running time in seconds}$$

where

SC = Number of non-comment source cards

CC = Number of comment source cards

NOTE: For source programs larger than 200 statements, the running time should be increased by 10%.

UTILITIES

Card System

Core image loader: loads at card read speed (1442 Model 6, 300 cpm; 1442 Model 7, 400 cpm).

Relocating loader: loads at card read speed. Core image converter:

Reads at an average speed of 380 cpm.

Punches at punch speed (1442 Model 6, 80 cpm; 1442 Model 7, 160 cpm).

Types (when required) 14.8 char. per sec.

Prints (1132) 82 lines per min.

Paper Tape System

Core image loader: loads at read speed (60 char. per sec.).

Relocating loader: loads at read speed.

Core image converter:

Pass one, read speed for input.

Pass two, input at read speed, output at punch speed; however, half of the output tape is generated while reading (punch speed, 15 char. per sec.)

Core Maps

At least 6 lines are printed for each core map. In addition, one line (of about 30 characters) is printed for each subprogram required. The summary contains N lines where

N is approximately equal to $M/6$

M is the total number of entry points used

SUBROUTINES

Subroutine times are shown in the publication IBM 1130 Subroutine Library (Form C26-5929).

APPENDIX C. SAMPLE PROGRAMS

ASSEMBLER CARD SYSTEM SAMPLE PROGRAM

The 1130 card assembler sample program is an absolute mainline program that calls no subroutines. It can be assembled and then loaded with the relocating loader. If desired, it can be assembled, converted to core image format, and loaded by the core image loader. The card system source program deck listing is shown in listing C2.

ASSEMBLY

Reproduce the source deck if desired.

Pass 1

1. With the console Mode switch set to RUN, press IMM STOP and RESET on the console.
2. Place the core image loader, the assembler, and the assembler sample program (source) in the card reader.
3. Press reader START and PROGRAM LOAD on the console.
4. Press reader START to process the last card. The source deck will be located in stacker #2.

Pass 2

1. Remove the source program deck from stacker #2 and place it in the reader hopper.
2. Press reader START. Data will be punched in the first 20 columns of the source deck. The output of Pass 2 is a list deck.
3. Press reader START to process the last card.

COMPRESSION

1. Press IMM STOP and RESET.
2. Place the core image loader, the compressor, and the list deck followed by blank cards in the card read punch hopper.
3. Press reader START and PROGRAM LOAD.
4. The compressed object deck in stacker #2 is the program object deck.

PROGRAM EXECUTION USING THE RELOCATING LOADER

1. Press IMM STOP and RESET on the console.
2. Place the relocating loader, the program object deck, EOD1, and EOD2 in the reader.
3. Place a blank card behind the stacked input in the reader.
4. Press reader START and PROGRAM LOAD.
5. When the blank card following EOD2 is read, the indicator lights of the accumulator and accumulator extension will blink on and off.

PROGRAM EXECUTION USING THE CORE IMAGE LOADER

1. Core image convert the program object deck (see Core Image Converter in Utility section of this manual).
2. Place the core image loader and the core image program deck in the card reader.
3. Place a blank card behind the stacked input in the card reader.
4. With the console Mode switch set to RUN, press reader START, IMM STOP, RESET, and PROGRAM LOAD.
5. When the blank card following the core image program deck is read, the indicator lights of the accumulator and accumulator extension will blink on and off.

ASSEMBLER PAPER TAPE SYSTEM SAMPLE PROGRAM

The 1130 paper tape assembler sample program is an absolute mainline program that calls no subroutines. It can be assembled and then loaded with the relocating loader. The paper tape source program tape listing is shown in listing C1.

OPERATING PROCEDURE

1. With the console Mode switch set to RUN, press IMM STOP and RESET on the console.
2. Place the core image loader tape in the paper

tape reader so that a delete code beyond the program ID is under the read starwheels.

3. Press PROGRAM LOAD on the console.
4. When the core image loader has been read in, the system will WAIT. Load the assembler tape in the reader and press PROGRAM START on the console.
5. The assembler tape will read in and the system will WAIT.
6. Load the assembler sample program source tape in the reader and press PROGRAM START.
7. The assembler sample program tape will read in and the system will WAIT. Ready the paper tape punch and reload the source tape in the reader.
8. Press PROGRAM START. During Pass 2 a list tape will be punched.
9. Press IMM STOP and RESET.
10. Place the core image loader tape in the paper tape reader.
11. Press PROGRAM LOAD.
12. When the core image loader has been read in, the system will WAIT.
13. Load the compressor tape in the paper tape reader.
14. Press PROGRAM START. When the compressor has been read in the system will WAIT.
15. Load the list tape in the paper tape reader.
16. Press PROGRAM START. During this pass, a compressed object tape will be punched.

SAMPLE PROGRAM EXECUTION

1. Press IMM STOP and RESET.
2. Place the relocating loader in the paper tape reader.
3. Press PROGRAM LOAD.
4. Place the paper tape sample object tape in the paper tape reader.
5. Press PROGRAM START.
6. Place the EOD1 tape in the paper tape reader.
7. Press PROGRAM START.
8. Place the EOD2 tape in the paper tape reader.
9. Press PROGRAM START.
10. When EOD2 has read in, the program will WAIT. Press PROGRAM START to execute the program. The indicator lights for the accumulator and accumulator extension will blink on and off.

FORTRAN CARD SYSTEM SAMPLE PROGRAM

This FORTRAN sample program is a simultaneous equation routine containing 94 cards. It will compile in approximately 3.5 minutes on a 4K, 3.6 μ s machine.

A listing of the FORTRAN card system sample program is shown in listing C3.

DECK FORMAT

The program consists of the following:

6	Compiler Control Cards
1	Declarative Statement
14	Format Statements
17	Read/Write Statements
27	Control Statements
18	Arithmetic Statements
11	Comments Cards

There are also 8 data cards after the END card.

The average length of a statement, including imbedded blanks, is less than 25 characters.

The operating procedures describe how to use the system as released by IBM. If the user has an 1130 with a configuration other than that required, he must change the unit assignment to conform to his configuration. As written, the program requires an 1132 Printer and a 1442 Card Read Punch.

OPERATING PROCEDURES

Compilation

1. With the console mode switch set to RUN, press IMM STOP and RESET on the console.
2. Place the following decks in the card read punch:

Core Image Loader
FORTRAN Compiler Input Phase (edited)
Sample Program (source) less 8 data cards
Remainder of Edited FORTRAN Compiler
(phases 2-26)
Blank Cards

3. Press reader START.

4. Press PROGRAM LOAD.

The program will execute and the message shown on the last page of the source program listing (listing C3) will be printed.

FORTRAN PAPER TAPE SYSTEM SAMPLE PROGRAM

This FORTRAN sample program is a simultaneous equation routine containing 94 records. It will compile in approximately 20 minutes on a 4K, 3.6 μ s machine.

A listing of the FORTRAN card system sample program is included in this appendix. The tape listing is identical except for the control records and the missing card IDs.

The *IOCS control records for paper tape are TYPEWRITER and PAPER TAPE.

PAPER TAPE FORMAT

The program consists of the following:

6	Compiler Control Records
1	Declarative Statement
14	Format Statements
17	Read/Write Statements
27	Control Statements
18	Arithmetic Statements
11	Comments Records

There are also 8 data records after the END record.

The average length of a statement, including imbedded blanks, is less than 25 characters. As written, the program requires a console printer, a 1054 paper tape reader, and a 1134 paper tape punch.

OPERATING PROCEDURES

Compilation

1. With the console Mode switch set to RUN, press IMM STOP and RESET on the console.
2. Ready the paper tape punch.
3. Place the core image loader tape in the paper tape reader.
4. Press PROGRAM LOAD on the console.
5. Place the FORTRAN Phase 1 tape in the reader.
6. Press PROGRAM START on the console.
7. Place the sample program (FORTRAN source) tape in the reader.
8. Press PROGRAM START. Note that there is a series of data records on the end of the source program tape. Do not load these records at this time, they will be required during program execution.
9. Place the remainder of the FORTRAN compiler in the reader.
10. Press PROGRAM START.

The object tape is punched by the paper tape punch when the compilation is complete.

Program Execution

1. Press IMM STOP and RESET.
2. Place the relocating loader tape in the paper tape reader.
3. Press PROGRAM LOAD.
4. Place the FORTRAN object tape in the reader.
5. Press PROGRAM START.
6. Place the subroutine library tapes (standard precision) in the reader.
7. Press PROGRAM START.
8. Place the source data records (on the end of the source tape) in the reader.
9. Press PROGRAM START.

The program will execute and the message shown on the last page of the source program listing (listing C3) will be printed.

```

*** HDNG 1130 PT SAMPL. ASSEMBLY SAMAP000
ABS SAM00000
* 1130 PAPER TAPE SYSTEM SAMPLE SAM00010
ALLF DC /CCCC SAM00020
0280 0 CCCC BSS E 0 SAM00030
0282 0 0000 DC /CCCC SAM00040
0284 00 00000000 CNT DEC 0 SAM00050
0286 0 0200 NT DC /200 SAM00060
0287 START EQU * START SAM00070
STD LDX I1 NT SAM00080
0289 00 75000400 MDX L1 /400 SAM00090
028B 0 C8F4 LDD ALLF SAM00100
028C 0 71FF MDX 1 -1 BLINK LIGHTS SAM00110
028D 0 70FD MDX *-3 SAM00120
028E 00 65800286 LDX I1 NT SAM00130
0290 00 75000250 MDX L1 /250 SAM00140
0292 0 C8F1 LDD CNT SAM00150
0293 0 71FF MDX 1 -1 SAM00160
0294 0 70FD MDX *-3 SAM00170
0295 0 C0F0 LD NT SAM00180
0296 0 901D S ONE SAM00190
0297 0 D0EE STO NT SAM00200
0298 00 4C18029E BSC L EX,6- SAM00210
029A 0 C8E5 LDD ALLF BLINK, BLINK, BLINK, SAM00220
029B 0 18C1 RTE 1 SAM00230
029C 0 D8E3 STD ALLF SAM00240
029D 0 70E9 MDX STD SAM00250
029E 00 65000CCC EX LDX L1 /0CCC SAM00260
02A0 0 69DF STX 1 ALLF SAM00270
02A1 00 65000300 LDX L1 /300 SHIFT, SAM00280
02A3 0 69E2 STX 1 NT SAM00290
02A4 0 C80D LDC LDD CHECK SAM00300
02A5 00 65800280 LDX I1 ALLF SAM00310
02A7 0 71FF MDX 1 -1 SAM00320
02A8 0 70FE MDX *-2 SAM00330
02A9 0 18C1 RTE 1 AND SAM00340
02AA 0 D807 STD CHECK BLINK SAM00350
02AB 0 C0D4 LD ALLF SOME SAM00360
02AC 0 9007 S ONE MORE. SAM00370
02AD 0 D0D2 STO ALLF SAM00380
02AE 0 4820 BSC 2 SAM00390
02AF 0 70F4 MDX LDC SAM00400
02B0 0 7004 MDX NXT SAM00410
02B2 0000 BES E 0 SAM00420
02B2 0 0600 CHECK DC /0600 SAM00430
02B3 0 0700 DC /0700 SAM00440
02B4 0 0001 ONE DC 1 SAM00450
02B5 NXT EQU * SAM00460
02B5 0 3000 WAIT SAM00470
02B6 00 74030280 MDX L ALLF,3 SAM00480
02B8 00 4C000287 BSC L STD SAM00490
02BA 0287 END START SAM00500

```

```

*** HDNG 1130 CARD SAMPL. ASSEMBLY SAMA 000
ABS SAMA 001
* SAMPLE PROGRAM - READ CARD AND BLINK LIGHTS SAMA 002
CARD BSS 80 INPUT SAMA 003
CRP EQU /1000 CARD READ/PUNCH SAMA 004
SDEV EQU /0700 SENSE DEVICE SAMA 005
CTL EQU /0400 CONTROL SAMA 006
READF EQU /0200 READ SAMA 007
WRITE EQU /0100 WRITE SAMA 008
FSTRT EQU /0002 FEED START SAMA 009
RSTRT EQU /0004 SAMA 010
L1 EQU 1 SAMA 011
L2 EQU 2 SAMA 012
030C 0000 BES E 0 SAMA 013
030C 0 0000 SCRIP DC 0 SAMA 014
030D 0 1700 DC CRP&SDEV SAMA 015
030E 0 0000 STRD DC 0 SAMA 016
030F 0 1404 DC CRP&CTL&RSTRT SAMA 017
0310 0 0000 SCRPD DC 0 SAMA 018
0311 0 1702 DC CRP&SDEV&L2 SAMA 019
0312 0 0000 SCRPT DC 0 SAMA 020
0313 0 1701 DC CRP&SDEV&L1 SAMA 021
0314 0 02BC RCLM DC CARD SAMA 022
0315 0 1200 DC CRP&READF SAMA 023
0316 0 0000 RDCM DC 0 SAMA 024
0317 00 65000381 START LDX L1 ISS0 SET ISS ADDRESSES SAMA 025
0319 00 6D000008 STX L1 8 SAMA 026
031B 00 65000370 LDX L1 ISS4 SAMA 027
031D 00 6D00000C STX L1 12 SAMA 028
031F 0 6100 LDX 1 0 SAMA 029

```

0320 00 67000F80		LDX L3 /OF80	INITIALIZE X3 FOR XFR	SAMA 030
0322 0 69F3		STX 1 RDCM	READ COMPLETE OFF	SAMA 031
0323 0 08E8	READ	XIO SCRIP	AWAIT READY	SAMA 032
0324 00 4C040323		BSC L *-3,E		SAMA 033
0326 00 650002BC		LDX L1 CARD		SAMA 034
0328 0 69EB		STX 1 RCLM	SET READ ADDR	SAMA 035
0329 0 08E4		XIO STRD		SAMA 036
032A 0 1010		SLA 16		SAMA 037
032B 0 DOEA		STO RDCM		SAMA 038
032C 0 COE9		LD RDCM		SAMA 039
032D 0 4818		BSC 6-	WAIT FOR	SAMA 040
032E 0 70FD		MDX *-3	READ COMPLETE	SAMA 041
032F 0 6135		LDX 1 53		SAMA 042
0330 00 C50002CE		LD L1 CARD&18	FIND RIGHTMOST NONBLANK	SAMA 043
0332 00 4C20033D		BSC L FDL+2		SAMA 044
0334 0 71FF		MDX 1 -1		SAMA 045
0335 0 70FA		MDX *-6		SAMA 046
0336 0 7006		MDX FDL		SAMA 047
0338 0000		BSS E 0		SAMA 048
0338 0 CCCC	ALLF	DC /CCCC		SAMA 049
0339 0 CCCC		DC /CCCC		SAMA 050
033A 00 00000000	CNT	DC 0		SAMA 051
033C 0 0200	NT	DC /200		SAMA 052
033D	FDL	EQU *		SAMA 053
033D 00 6580033C	STD	LDX I1 NT		SAMA 054
033F 00 75000400		MDX L1 /400		SAMA 055
0341 0 C8F6		LDD ALLF		SAMA 056
0342 0 71FF		MDX 1 -1	BLINK LIGHTS	SAMA 057
0343 0 70FD		MDX *-3		SAMA 058
0344 00 6580033C		LDX I1 NT		SAMA 059
0346 00 75000250		MDX L1 /250		SAMA 060
0348 0 C8F1		LDD CNT		SAMA 061
0349 0 71FF		MDX 1 -1		SAMA 062
034A 0 70FD		MDX *-3		SAMA 063
034B 0 COF0		LD NT		SAMA 064
034C 0 901D		S ONE		SAMA 065
034D 0 DOEE		STO NT		SAMA 066
034E 00 4C180354		BSC L EX,6-		SAMA 067
0350 0 C8E7		LDD ALLF		SAMA 068
0351 0 18C1		RTE 1		SAMA 069
0352 0 D8E5		STD ALLF		SAMA 070
0353 0 70E9		MDX STD		SAMA 071
0354 00 65000CCC	EX	LDX L1 /0CCC		SAMA 072
0356 0 69E1		STX 1 ALLF		SAMA 073
0357 00 65000300		LDX L1 /300	SHIFT,	SAMA 074
0359 0 69E2		STX 1 NT		SAMA 075
035A 0 C80D	LDC	LDD CHECK		SAMA 076
035B 00 65800338		LDX I1 ALLF		SAMA 077
035D 0 71FF		MDX 1 -1		SAMA 078
035E 0 70FE		MDX *-2		SAMA 079
035F 0 18C1		RTE 1	AND	SAMA 080
0360 0 D807		STD CHECK	BLINK	SAMA 081
0361 0 C0D6		LD ALLF	SOME	SAMA 082
0362 0 9007		S ONE	MORE.	SAMA 083
0363 0 D0D4		STO ALLF		SAMA 084
0364 0 4820		BSC Z		SAMA 085
0365 0 70F4		MDX LDC		SAMA 086
0366 0 7004		MDX NXT		SAMA 087
0368 0000		BES E 0		SAMA 088
0368 0 0600	CHECK	DC /0600		SAMA 089
0369 0 0700		DC /0700		SAMA 090
036A 0 0001	ONE	DC 1		SAMA 091
036B	NXT	EQU *		SAMA 092
036B 0 3000		WAIT		SAMA 093
036C 00 74030338		MDX L ALLF,3		SAMA 094
036E 00 4C00033D		BSC L STD		SAMA 095
0370 0 0000	ISS4	DC 0		SAMA 096
0371 0 D01D		STO SAVA4		SAMA 097
0372 0 0899		XIO SCRIP		SAMA 098
0373 0 1004		SLA 4	TEST FOR READ OP COMPLETE	SAMA 099
0374 00 4C280379		BSC L READX,6Z	TEST FOR CRP	SAMA 100
0376 0 C018	EXIT	LD SAVA4	RESTORE.	SAMA 101
0377 00 4CC00370		BOSC I ISS4	EXIT.	SAMA 102
0379 00 0C000310	READX	XIO L SCRIPD		SAMA 103
037B 0 689A		STX RDCM	SET RDCM ON	SAMA 104
037C 00 650002BC		LDX L1 CARD	RESET READ	SAMA 105
037E 00 6D000314		STX L1 RCLM	ADDR	SAMA 106
0380 0 70F5		MDX EXIT		SAMA 107
0381 0 0000	ISS0	DC 0		SAMA 108
0382 0 D00B		STO SAVA0		SAMA 109
0383 00 0C000312		XIO L SCRIP	TURN OFF INTERRUPT INDIC.	SAMA 110
0385 00 0C000314		XIO L RCLM	READ COLUMN	SAMA 111
0387 00 74010314		MDX L RCLM,1	INCR READ ADDRESS	SAMA 112
0389 00 0C00030E		XIO L STRD		SAMA 113
038B 0 C002		LD SAVA0		SAMA 114
038C 00 4CC00381		BOSC I ISS0		SAMA 115
038E 0 0000	SAVA0	DC 0		SAMA 116
038F 0 0000	SAVA4	DC 0		SAMA 117
0390 0317		END START		SAMA 118

```
**      IBM 1130 FORTRAN SAMPLE PROGRAM OF 86 STATEMENTS          SAMF 001
*LIST ALL                                                         SAMF 002
*ONE WORD INTEGERS                                              SAMF 003
*IOCS(1132 PRINTER)                                           SAMF 004
*IOCS(CARD)                                                    SAMF 005
*NAME SAMPL                                                    SAMF 006
      IBM 1130 FORTRAN SAMPLE PROGRAM OF 86 STATEMENTS          PAGE 02

C      IBM 1130 FORTRAN SAMPLE PROGRAM OF 86 STATEMENTS          SAMF 007
C      SIMULTANEOUS EQUATION ROUTINE                             SAMF 008
      DIMENSION A(10,10),X(10),B(10)                             SAMF 009
301  FORMAT (1H1,20X15HINCOMPATIBILITY)                          SAMF 010
302  FORMAT (1H 20X41HMORE EQUATIONS THAN UNKNOWNNS-NO SOLUTIONS) SAMF 011
303  FORMAT (1H 20X46HMORE UNKNOWNNS THAN EQUARIONS-SEVERAL SOLUTIONS) SAMF 012
304  FORMAT (1H 20X15HSOLUTION MATRIX)                           SAMF 013
305  FORMAT(1H 20X8HMATRIX A)                                    SAMF 014
306  FORMAT(1H 20X8HMATRIX B)                                    SAMF 015
307  FORMAT (1H 20X10H A-INVERSE)                               SAMF 016
308  FORMAT(1H 20X24HDIAGONAL ELEMENT IS ZERO)                  SAMF 017
      M=2                                                         SAMF 018
      L=3                                                         SAMF 019
      READ (M,10)                                                 SAMF 020
10  FORMAT(72H          SPACE FOR TITLE                           SAMF 021
      1)                                                           SAMF 022
      WRITE (L,10)                                                SAMF 023
12  FORMAT (6I10)                                                SAMF 024
      READ (M,12)M1,M2,L1,L2,N1,N2                               SAMF 025
C      M1 = NO. OF ROWS OF A                                     SAMF 026
C      M2 = NO. OF COLS OF A                                     SAMF 027
C      L1 = NO. OF ROWS OF X                                     SAMF 028
C      L2 = NO. OF COLS OF X                                     SAMF 029
C      N1 = NO. OF ROWS OF B                                     SAMF 030
C      N2 = NO. OF COLS OF B                                     SAMF 031
13  FORMAT (7F10.4)                                               SAMF 032
17  FORMAT (10F10.4)                                               SAMF 033
      IF (N2-1)63,64,63                                           SAMF 034
64  IF (L2-1)63,65,63                                           SAMF 035
65  IF (L1-M2)63,66,63                                           SAMF 036
66  IF (M1-N1)63,11,63                                           SAMF 037
63  WRITE (L,301)                                                 SAMF 038
      GO TO 2                                                       SAMF 039
11  N=M1                                                           SAMF 040
      N=M2                                                           SAMF 041
      IF (M1-M2) 91,14,93                                         SAMF 042
91  WRITE (L,302)                                                 SAMF 043
      GO TO 2                                                       SAMF 044
93  WRITE (L,303)                                                 SAMF 045
      GO TO 2                                                       SAMF 046
14  WRITE (L,305)                                                 SAMF 047
      DO 70 I=1,N                                                  SAMF 048
          READ (M,13)(A(I,J),J=1,N)                               SAMF 049
          WRITE (L,17)(A(I,J),J=1,N)                              SAMF 050
```

```

70 CONTINUE
89 FORMAT (F10.4)
WRITE (L,306)
READ (M,89)(B(I),I=1,N)
WRITE (L,89)(B(I),I=1,N)
C INVERSION OF A
20 DO 120 K=1,N
D=A(K,K)
IF(D)40,200,40
40 A(K,K)=1.0
50 DO 60 J=1,N
60 A(K,J)=A(K,J)/D
IF(K=N)80,130,130
80 IK=K+1
IBM 1130 FORTRAN SAMPLE PROGRAM OF 86 STATEMENTS
DO 120 I=1,K,N
D=A(I,K)
A(I,K)=0.0
DO 120 J=1,N
120 A(I,J)=A(I,J)-(D*A(K,J))
C BACK SOLUTION
130 IK=N-1
DO 180 K=1,IK
140 I1=K+1
DO 180 I=I1,N
D=A(K,I)
A(K,I)=0.0
170 DO 180 J=1,N
180 A(K,J)=A(K,J)-(D*A(I,J))
GO TO 202
200 WRITE (L,308)
GO TO 2
202 WRITE (L,307)
DO 201 I=1,N
WRITE (L,17)(A(I,J),J=1,N)
201 CONTINUE
DO 21 I=1,N
X(I)=0.0
DO 21 K=1,N
21 X(I)=X(I)+A(I,K)*B(K)
WRITE (L,304)
WRITE (L,89)(X(I),I=1,N)
2 STOP 5555
END
IBM 1130 FORTRAN SAMPLE PROGRAM
3 3 3 1 3 1
42150 -12120 11050
-21200 35050 -16320
11220 -13130 39860
32160
12470
23456

```

SAMF 051
SAMF 052
SAMF 053
SAMF 054
SAMF 055
SAMF 056
SAMF 057
SAMF 058
SAMF 059
SAMF 060
SAMF 061
SAMF 062
SAMF 063
SAMF 064
PAGE 03
SAMF 065
SAMF 066
SAMF 067
SAMF 068
SAMF 069
SAMF 070
SAMF 071
SAMF 072
SAMF 073
SAMF 074
SAMF 075
SAMF 076
SAMF 077
SAMF 078
SAMF 079
SAMF 080
SAMF 081
SAMF 082
SAMF 083
SAMF 084
SAMF 085
SAMF 086
SAMF 087
SAMF 088
SAMF 089
SAMF 090
SAMF 091
SAMF 092
SAMF 093
SAMF 094
SAMF 095
SAMF 096
SAMF 097
SAMF 098
SAMF 099
SAMF 100
SAMF 101

IBM 1130 FORTRAN SAMPLE PROGRAM OF 86 STATEMENTS

PAGE 04

VARIABLE ALLOCATIONS
A =00C6 X =00DA B =00EE D =00FO M =00F2 L =00F3 M1 =00F4 M2 =00F5 L1 =00F6 L2 =00F7
N1 =00F8 N2 =00F9 N =00FA I =00FB J =00FC K =00FD IK =00FE I1 =00FF

UNREFERENCED STATEMENTS
20 50 140 170

STATEMENT ALLOCATIONS
301 =010D 302 =011A 303 =0134 304 =0150 305 =015D 306 =0166 307 =016F 308 =0179 10 =018A 12 =0180
13 =0183 17 =0186 89 =0189 64 =01FO 65 =01F6 66 =01FC 63 =0202 11 =0208 91 =0218 93 =021E
14 =0224 70 =025D 20 =0296 40 =02AA 50 =02B5 60 =02B9 80 =02D4 120 =02F3 130 =0323 140 =032D
170 =0348 180 =034C 200 =037E 202 =0384 201 =03A5 21 =03BE 2 =0408

FEATURES SUPPORTED
ONE WORD INTEGERS
IOCS

CALLED SUBPROGRAMS
FADDX FMPYX FDIV FLD FLDX FSTO FSTOX FSBRX WRED WWRT WCOMP WFIO WIOFX WIOI SUBSC
STOP CARDZ PRNTZ

REAL CONSTANTS
.100000E 01=0104 .000000E 00=0106

INTEGER CONSTANTS
2=0108 3=0109 1=010A 5555=010B 21845=010C

CORE REQUIREMENTS FOR SAMPL
COMMON 0 VARIABLES 260 PROGRAM 774

Table C3. 1130 Arithmetic and Functional Subroutine Times

STANDARD		EXTENDED	
FADD/FADDX } FSUB/FSUBX } FMPY/FMPYX FDIV/FDIVX FLD/FLDX } FSTO/FSTOX } FLOAT IFIX NORM FSBR/FSBRX FDVR/FDVRX SNR FABS/FAVL IABS FGETP FARC XMDS FIXI/FIXIX XSQR 550 av. (860 max.) XMD XDD FSIN/FSINE } FCOS/FCOSN } FATAN/FATN FSQRT/FSQR FALOG/FLN FEXP/XP FAXI/FAXIX FAXB/FAXBX FTANH/FTNH FBTD (bin. to dec.) FDTB (dec. to bin.)	460 560 560 766 180 180 330 140 260 650 1090 80 50 100 330 60 260 465 550 av. (860 max.) 520 1760 3.0 3.4 5.2 4.5 5.1 2.0 3.8 8.0 4.3 40.0 20.0	EADD/EADDX } ESUB/ESUBX } EMPY/EMPYX EDIV/EDIVX ELD/ELDX } ESTO/ESTOX } ESBR/ESBRX EDVR/EDVRX EABS/EAVL EGETP ESIN/ESINE } ECOS/ECOSN } EATAN/EATN ESQRT/ESQR EALOG/ELN EEXP/EXP EAXI/EAXIX EAXB/EAXBX ETANH/ETNH	440 490 790 2060 160 170 330 140 260 740 2520 80 60 100 320 60 -- 465 550 av. (860 max.) 520 1760 5.4 5.9 8.9 10.4 8.0 4.4 4.7 13.3 8.1 40.0 20.0

APPENDIX D. 1130 CARD DECK AND PAPER TAPE IDENTIFICATION

Table D1 contains the card deck and paper tape ID for all programs and routines except those contained in the 1130 Subroutine Library. For subroutine library ID see Figure 4 in text.

The basic organization of card columns 73-80 is as follows:

- 73-77 Identification
- 78 Modification level number
- 79-80 Sequence number

The identification field (columns 73-77) for the FORTRAN compilers is sub-divided as follows:

- 73-74 Identification (FC)
- 75-76 Phase number
- 77 I/O device indicator
 - V - 1132 Printer
 - T - Console printer
 - A - All

Paper tapes are identified by the program name and BP number punched in the leader.

Some of the utilities are made up of (or preceded by) special bootstrap cards that do not contain ID and sequence information. Table D2 identifies these cards.

● Table D1. Program and Routine Identification

Program and Routine Name	Card ID (col. 73-77)	Paper Tape ID (punched in leader)
Sample Assembly Program	SAMA0	BP01
Sample FORTRAN Program	SAMF0	BP01
1130 Assembler	ASM00	BP02
1130 Compressor	COM00	BP03
1130 FORTRAN Compiler for 1132 Printer	FC01V FC02A-FC19A FC20V-FC21V FC22A FC23V-FC25V FC26A	BP04 BP05 BP05 BP05 BP05 BP05
1130 FORTRAN Compiler for Console Printer	FC01T FC02A-FC19A FC20T-FC21T FC22A FC23T-FC25T FC26A	BP02 BP03 BP03 BP03 BP03 BP03
1130 I/O Utilities	UT100	BP06
1130 Relocating Loader	RLD00	BP01
1130 Core Image Con- verters	CICT CICV	BP03 ① BP04 ② BP02
1130 Core Image Loader		
1130 Dump Between Limits on 1442	UT400	
1130 Dump Between Limits on Console Printer	UT500	BP14
1130 Keyboard Routine	UT8	
1130 Card Reproducing Routine	UT900	
1130 Disk Pack Initializa- tion Routine	DPIR	BP10 BP05 BP07
Dump and Console Utilities		
Construct Paper Tape		
EOD1 (end of deck, two- word call)	EOD01	BP08
EOD2 (end of deck, one- word LIBF.)	EOD02	BP09

① Core map on console printer.

② Core map on 1132 Printer.

Table D2. Dump Routines and Loaders — Card ID

		1132 CORE DUMP FROM 00F0													
CARD	COLUMN	1		2		3		4			5		6		
		1	2	1	2	1	2	1	2	3	1	2			
12					X		X	X				X	X		ALL COLUMNS BLANK
11		X			X		X	X				X	X		
0		X													
1		X	X				X					X	X		
2					X								X		
3					X		X								
4															
5								X				X			
6															
7		X	X												
8		X													
9		X													

		1132 CORE DUMP FROM 00A0								
CARD	COLUMN	1		2		3		4		
		1	2	1	2	1	2			
12							X	X		ALL COLUMNS BLANK
11		X	X				X	X		
0		X								
1		X					X			
2			X							
3					X					
4		X								
5								X		
6		X								
7		X						X		
8		X						X		
9		X						X		

		1 CARD DUMP*		
CARD	COLUMN	1		
		1	2	3
12		X		X
11		X		X
0				
1			X	X
2		X		
3			X	
4				
5				
6			X	
7		X		
8				X
9		X	X	X

		CORE IMAGE LOADER											
CARD	COLUMN	1		2		3		4		5		6	
		1	2	1	2	1	2	1	2	1	2	1	2
12								X					
11		X				X		X					
0		X								X		X	
1		X	X				X	X		X		X	
2						X		X					
3					X			X					
4						X				X		X	
5					X								
6									X				
7		X	X										
8													
9		X											

		RELOCATING LOADER													
CARD	COLUMN	1		2			3			4		5		6	
		1	2	1	2	3	1	2	3	1	2	1	2	1	2
12															X
11		X				X					X			X	X
0		X							X		X	X			X
1		X	X						X	X	X				X
2									X		X				X
3					X				X		X		X	X	X
4									X		X			X	X
5										X				X	X
6					X				X			X			X
7		X	X												X
8															X
9		X													X

*Console routine.

APPENDIX E. RELOCATABLE CARD AND PAPER TAPE FORMATS

The following cards are generated by the compressor and FORTRAN compiler and are processed by the relocating loader and core image converter.

The format is in terms of words on binary card (see Appendix F). The card ID and sequence numbers (columns 73-80) are in IBM card code.

In the paper tape system, each binary record is preceded by a one word frame count that gives the total number of words in this record (not counting the word count). Trailing zeros are deleted from all non-data records. At the beginning and end of the tape and between records, the delete codes are ignored.

Mainline Header Card

A mainline header card specifies the size of the common area and the size of the work area. It is the first card of the mainline program. The format is as follows:

<u>Word</u>	<u>Contents</u>
1	Reserved
2	Checksum*
3	Type code (first 8 bits): 0000 0001 - absolute 0000 0010 - relocatable Precision code (last 8 bits): 0000 0001 - standard 0000 0010 - extended 0000 0000 - undefined
4	Reserved
5	Length of COMMON storage area (FORTRAN mainline program only)
6	0000 0000 0000 0011
7	Work area required (FORTRAN only)
8-54	Reserved

*The checksum is the two's complement of the logical sum of the record count (position of the record within the deck or tape) and the data word(s). The logical sum is obtained by summing the data word(s) and the record count arithmetically with the addition of a one each time a carry occurs out of the high order position of the accumulator.

Data Cards

Data cards contain the instructions and data that constitute the assembled program. The format is as follows:

<u>Word</u>	<u>Contents</u>
1	Location (The relative load address of the first data word of the card or record. Succeeding words go into higher numbered core locations. The relocation factor must be added to this address to obtain the actual load address. For an absolute program the relocation factor is zero.)
2	Checksum
3	Type code (first 8 bits): 0000 1010 Data word count (last 8 bits)
4-9	Relocation indicators (2 bits per data word): 00 - nonrelocatable or absolute 01 - relocatable 10 - LIBF (one word call) 11 - CALL (two word call)
10	Data word 1
11-54	Data words 2 through 45

EOP Card

An EOP (end of program) card is the last card of each program and subroutine. The format is as follows:

<u>Word</u>	<u>Contents</u>
1	Starting location of next routine (this number is always even and is assigned by the assembler)
2	Checksum
3	Type code (first 8 bits): 0000 1111 Last 8 bits: 0000 0000
4	XEQ address, if mainline program
5-54	Reserved

Subroutine Header Card

The compressor or FORTRAN compiler produces a subroutine header card for all compressed subroutines. A maximum of 10 entry points can be defined for each subroutine. The format of the subroutine header card is as follows:

<u>Word</u>	<u>Contents</u>
1	Reserved
2	Checksum
3	Type code (first 8 bits); 0000 0011 - to be called by a one-word call only (LIBF) 0000 0100 - to be called by a two-word call only (CALL) Precision code (last 8 bits): 0000 0000 - undefined 0000 0001 - standard 0000 0010 - extended
4-5	Reserved
6	Number of entry points times three
7-9	Reserved
10-11	Name of entry point 1
12	Relative address of entry point 1
13-39	Names and relative addresses of entry points 2 through 10
40-54	Reserved

ISS Header Card

The compressor produces an ISS (interrupt service subroutine) header card for each user-written interrupt service subroutine. This card identifies the entry point defined by an ISS statement in the user's program. Only one entry point can be defined for each subroutine. The format of the ISS header card is as follows:

<u>Word</u>	<u>Contents</u>
1	Reserved
2	Checksum
3	Type code (first 8 bits): 0000 0101 - to be called by a one-word call only (LIBF) 0000 0110 - to be called by a two-word call only (CALL) Precision code (last 8 bits): 0000 0000 - undefined 0000 0001 - standard 0000 0010 - extended

<u>Word</u>	<u>Contents</u>
4-5	Reserved
6	Six plus number of interrupt levels required
7-9	Reserved
10-11	Subroutine name
12	Relative entry address
13	Address of ISTV (interrupt service transfer vector) is equal to 51 ₁₀ plus the ISS number. *
14	ISS number (displacement in ISTV) =1-20 ₁₀
15	Number of interrupt levels required
16	ID number for the primary interrupt level required = (0-5)
17-29	ID numbers for remaining interrupt levels required = (0-5)
30	Edit word (1 for 1130, 2 for 1800)
31-54	Reserved

*The ISTV table is initialized during the execution of the re-locating loader and core image converter programs. This table starts at location 0034. Each TV entry in this table contains the starting addresses for the corresponding ISS routine (maximum of 19 TV entries).

ILS Header Card

An ILS (interrupt level subroutine) header card identifies the ILS routine. The format of the ILS header card is as follows:

<u>Word</u>	<u>Contents</u>
1	Reserved
2	Checksum
3	Type code (first 8 bits): 0000 0111 Reserved (last 8 bits)
4-5	Reserved
6	0000 0000 0000 0100
7-9	Reserved
10-12	Reserved
13	Interrupt level number
14-54	Reserved

Loader Overlay Cards

Loader overlay cards contain instructions that are part of the loader. Several loader cards (usually three before and three after the header card) are generated for each subroutine. ~~The format is as follows.~~ *EOD1 & EOD2 cards are also loader overlay cards. The loader overlay card format is as follows:*

<u>Word</u>	<u>Contents</u>
1	Location
2	Checksum
3	Type code (first 8 bits): 0000 1100, 0000 1101, or 0000 1110

<u>Word</u>	<u>Contents</u>
	Word count (last 8 bits)
4-9	Relocation indicators
10-54	Loader data

APPENDIX G. ERROR WAITS AND PROGRAM WAITS

Description	Card System			Paper Tape System		
	Relocating Loader	Core Image Loader	Core Image Converter	Relocating Loader	Core Image Loader	Core Image Converter
<u>No mainline in deck.</u> No mainline program in deck.	0412		0E7Bw/1132 0E11w/C.P.	02FB		0C98w/1132 0C1Cw/C.P.
<u>Automatic Load (ALD) Error.</u> The ALD card has a format error. (The first 4 columns of an ALD card must be blank. Punches in these columns will not be detected as errors.)	02F9		0AC2w/1132 0AC8w/C.P.			
<u>READ or FEED Error.</u> Incorrect card registration or a difference between the first and second reading of a column. Press NPRO. Place the two non-processed cards in the hopper ahead of the remainder of the deck. Press reader START and then program START on the console.	025B	0093	07E3			
<u>Precision Error.</u> If the required subroutine is of an incorrect precision, the program stops, prior to loading that subroutine, at these locations.	01FA		0359	01FA		0417w/1132 040Fw/C.P.
If the incorrect precision concerns an ISS (interrupt service subroutine), the program stops, prior to loading the subroutine, at these locations.	0225		0416	0225		04D4w/1132 04CCw/C.P.
<u>PUNCH Error.</u> Press NPRO. Discard the non-processed cards and continue by pressing reader START and program START on the console.			Loops: 0744-0747			
<u>Incorrect Program Reference.</u> If a subroutine is referred to by an LIBF (CALL) and its header card states it must be referred to by a CALL (LIBF), the program waits at these locations:	0221		037D	0221		043Bw/1132 0433w/C.P.
If either of these conditions is noted on an ISS, the program waits at:	0227		0420	0227		04DEw/1132 04D6w/C.P.
Console Entry switches about to be read in:			0DD5w/1132 0D72w/C.P.			0C1Bw/1132 0BA6w/C.P.
End of first pass--core image converter:			04A6			055Aw/1132 0552w/C.P.
<u>ILS Required, but Not Found.</u> If the ILS (interrupt level subroutine) is in the program deck, another pass is required. If the ILS is found, insert it in the reader, followed by the EOD2 records. Press reader START and program START on the console.	01C1		046C	01C1		052Aw/1132 0522w/C.P.
<u>Transfer Vector Filled or About To Be Overlaid.</u> When the program waits at these locations:	008E		00F0	008E		00F0w/1132 00E6w/C.P. (SBR-70FF)
1. TV (transfer vector) is filled; no more entries are required, or 2. TV is about to be overlaid with data, or 3. More than one mainline is in the deck.						
<u>Multiple Use of an ISTV.</u> If more than one ISS attempts to use the same ISTV (interrupt service transfer vector), the program branches to itself at these locations.	01F5		03E6	01F5		04A4w/1132 049Cw/C.P.

Description	Card System			Paper Tape System		
	Relocating Loader	Core Image Loader	Core Image Converter	Relocating Loader	Core Image Loader	Core Image Converter
<p><u>Checksum.</u> This error occurs if a card is not read correctly, or if an erroneous card is in the deck. If the stop occurs when a user's data information card is read, it indicates that the loading process is not finished and that this data information card does not belong in the program deck. For the paper tape system, see appendix H.</p> <p>Checksum error on first record.</p>	024D	0025	050F	024D	0033	05C3w/1132 05B8w/C.P. (SBR-70FF)
<p><u>CALL Error.</u> An LIBF subprogram contains a CALL.</p>	01A6		051D	01A6		05D1w/1132 05C9w/C.P. (SBR-70FF)
<p><u>Program Loops.</u> Card read punch out of cards; paper tape reader not ready.</p> <p>Paper tape punch not ready.</p>	Loops: 00A1-00A3	Loops: 0073-0075	Loops: 0112-0114	00E6	006C	01E3w/1132 01D9w/C.P. 01B8w/1132 01AEw/C.P.
<p><u>PROGRAM STOP.</u> Pressing PROGRAM STOP causes a halt. Press PROGRAM START to continue. An exception to the rule is the paper tape core image converter wait. To continue:</p> <ol style="list-style-type: none"> 1. Set console Mode switch to DISPLAY and press PROGRAM START. 2. Set the console Mode switch to RUN and press PROGRAM START. 	0012	0011	0012	0012	0012	000F

APPENDIX H. PAPER TAPE READER ERROR (CHECKSUM) RESTART

The following restart procedure must be used if a paper tape reader error occurs while programs are being read under the control of the core image loader, the relocating loader, or the core image converter.

Restart Procedure

Using the appropriate row in Table H 1, decrement the contents of the location specified in column A by 1. Display the contents of the location specified in column B. The two right-hand hexadecimal characters displayed compose the word count of the record just read. Back the tape up $x+2$ frames ($x = 2$ times the word count), branch to the location specified in column C, and restart.

Table H1. Restart Procedure Displays

Name	A	B	C
Core image loader	0034	0028	006E
Relocating loader	0196	0241	00A1
Core image converter with 1132 Printer	02C2	05B0	01BF
Core image converter with console printer	02B8	05A8	01B5

If, after a restart, the checksum error causes the reader to stop at the same point on the tape, then a defective tape is indicated (i.e., worn or damaged tape, or a punch error). If the stop is not at the same point on the tape, the restart point must be recalculated by initiating another restart procedure.

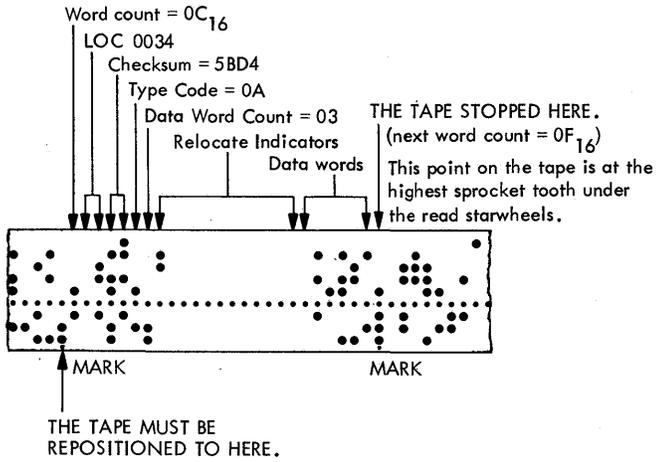
Detailed Operator Instructions

When a paper tape read error occurs:

1. Press IMM STOP on the console

2. Set the Console Entry switches to the appropriate address specified in column A of Table H 1
3. Turn the console Mode switch to LOAD
4. Press LOAD IAR
5. Turn the console Mode switch to DISPLAY
6. Press PROGRAM START
7. Subtract 0001 hexadecimal from the contents of the Storage Buffer register and write down the result
8. Turn the console Mode switch to LOAD
9. Press LOAD IAR
10. Set the Console Entry switches to the result written down in step 7
11. Press PROGRAM START
12. Set the Console Entry switches to the appropriate address specified in column B of Table H 1
13. Press LOAD IAR
14. Turn the console Mode switch to DISPLAY
15. Press PROGRAM START
16. Multiply the decimal equivalent of the two right-hand hexadecimal characters (word count) in the Storage Buffer register by 2. Add 2 to this figure. The result is the total number of frames the tape must be back-spaced for a restart.
17. Place a mark on the tape adjacent to the highest sprocket tooth under the read star-wheels as a point of reference. Count back the number of frames calculated in step 16 (in lieu of counting back, the tape may be measured - 10 frames per inch) and mark the tape, move the tape until the mark is at the point of reference. If the frame to the right of the reference point does not contain the word count displayed in step 16, a reader problem is indicated
18. Turn the console Mode switch to LOAD
19. Set the Console Entry switches to the appropriate address specified in column C of Table H 1
20. Press LOAD IAR
21. Turn the console Mode switch to RUN
22. Press PROGRAM START

Example:
 Typical data record (type A)



Step 15 of the procedure displays the word count in the SBR. In this case it is 670C. Convert the two right characters to decimal: 0C₁₆ = 12₁₀, multiply this number by 2 and add 2, as indicated in step 16.

$$12 \times 2 + 2 = 26$$

Mark the tape at the high point under the read head. Count back 26 frames and mark the restart position. The hexadecimal characters in the tape frame to the right of the restart position should be the same as the two right-hand hexadecimal characters displayed in the SBR.

APPENDIX I CONSOLE AND INTERRUPT RESTART PROCEDURES

This appendix describes the sequence of events which occur on the 1130 card/paper tape system when the following manual interventions occur:

PROGRAM STOP	} Cause Interrupts
INT RUN (console Mode switch setting)	
INT REQ (keyboard)	
IMM STOP	
RESET	
1442 Card Read Punch STOP	
1132 Printer STOP	

The identification and purpose of each intervention are covered, and the operator actions required to continue are described in detail for PROGRAM STOP, INT RUN, and INT REQ.

As noted above, three of the operator actions initiate interrupts. Each of these interrupts causes an automatic branch from the normal program sequence. Six interrupt priority levels are available; they are assigned as follows (0 = highest priority, 5 = lowest priority):

<u>Level</u>	<u>Device</u>
0	1442 Card Read Punch (column read, punch)
1	1132 Printer
2	Disk Storage
3	1627 Plotter
4	1442 (operation complete); Keyboard/Console Printer; 1134 Paper Tape Reader, 1055 Paper Tape Punch
5	Console

When an interrupt request is detected, the program is directed to service the request by interrupting the program sequence. All interrupt requests of equal or lower status are prevented from interrupting while a higher priority interrupt is being serviced.

Sample ILS05 programs are given in text.

Tables I1 and I2 list interrupt stop and restart procedures for the card and paper tape systems.

MACHINE OPERATIONS

OPERATOR INTERVENTIONS WHICH CAUSE INTERRUPTS

PROGRAM STOP

Pressing PROGRAM STOP causes an interrupt on level 5.

Purpose. Allows the processor to "cycle down" by completing all I/O operations in progress before halting. Cycle steal operations and interrupt level processing (in progress or pending) for levels 0-4 are completed before a WAIT instruction actually halts the machine.

Identification. Bit 0 of the console DSW (Device Status Word) is ON.

Recovery.

1. Press PROGRAM START to clear the interrupt.
2. Execute a BOSC instruction to clear the level.

INT RUN

Running with the mode switch in INT RUN causes an interrupt on level 5 after each mainline instruction is executed. A mainline instruction is one that is executed when no interrupt level is on.

Purpose. Allows a mainline program to be traced.

Identification. Bit 1 of the console DSW is ON.

Recovery. Execute a BOSC instruction to clear the level.

INT REQ

Pressing INT REQ on the Input Keyboard causes an interrupt on level 4.

Purpose. To inform the program that the operator wishes to enter information through the Input Keyboard or Console Entry switches.

Identification. Bit 2 of the input keyboard/console printer DSW is ON.

Recovery.

1. Execute an XIO (sense DSW with reset) instruction for the input keyboard/console printer to reset the interrupt.
2. Execute a BOSC instruction to clear the level.

OPERATOR INTERVENTIONS WHICH DO NOT CAUSE INTERRUPTS

IMM STOP

Halts the processor by preventing the execution of any further instructions.

1. Cycle steals will be honored and interrupts will be set but not acknowledged; 1442 Card Read Punch and 1132 Printer I/O operations may not be recoverable.
2. Recovery - Press PROGRAM START.

RESET

Terminates any cycle steal or I/O operation in progress, resets all clocks, I/O devices, machine registers, and interrupts.

1. The index registers are not reset since they are part of core storage.
2. Reset is operational only when the machine is halted by a PROGRAM STOP, an IMM STOP, or a WAIT instruction.

1442 Card Read Punch STOP

Pressing this key while a 1442 operation is not in progress or when the 1442 operation-complete

interrupt is given turns off the 1442 Ready light and removes the card read punch from a ready status. Bit 15 of the 1442 DSW is set ON. If a reader punch operation is in progress, the Stop key should be held down until the request is honored.

1132 Printer STOP

Pressing this key turns off the 1132 Ready light and removes the printer from a ready status. Bit 5 of the 1132 DSW is set ON. These actions occur when an 1132 I/O operation is not in progress or when the next XIO (Stop Printer) instruction is given.

PROGRAMMING SYSTEM OPERATION

PROGRAM STOP INTERRUPTS

Recovery from a PROGRAM STOP interrupt requires both operator and program action. Therefore, whether or not recovery is possible depends on the program in operation. See Tables I1 and I2 for the response of the IBM-provided programs and sub-routines.

User-written programs, loaded by either the relocating loader or core image loader, fall under the heading "user" if:

1. The user program does not call an ISS requiring a level 5 ILS, and
2. The user program does not alter the interrupt branch address for level 5 (core location 000D), and
3. The user program does not alter the interrupt trap (core locations 002E-0031).

If these conditions are met, pressing PROGRAM STOP will cause a branch to the interrupt trap.

A sample ILS05 follows. This program halts if PROGRAM STOP is pressed and restarts when PROGRAM START is pressed. This program ignores INT RUN interrupts.

SAVA	ILS	05		
	DC	/0001	must not be zero (at load time); modified by the loader; used to save the accum- ulator	
START	DC	0	interrupt entry point	
	STO	SAVA		
	XIO	SENSE-1	sense console DSW	
	BSC	Z +	skip if not PROGRAM STOP	
	WAIT			
	LD	SAVA		
	BOSC	I	clear interrupt level and exit	
SENSE	DC	/3F00	must be in odd location	
	END			

INT RUN INTERRUPTS

Recovery from an INT RUN interrupt requires program action. Therefore, whether or not recovery is possible depends on the program in operation. See Tables I1 and I2 for the response of the IBM-provided programs and subroutines.

A user-written program is treated in the same manner and with the same restrictions as described in the PROGRAM STOP Interrupts section.

A sample ILS05 follows. This program halts if PROGRAM STOP is pressed, and restarts when PROGRAM START is pressed. This program processes INT RUN interrupts.

	ILS	05		
	DC	/0001	must not be zero (at load time); modified by loader	
START	DC	0	interrupt entry point	
	STD	SAVAQ	save accumulator and extension	
	XIO	SENSE-1	sense console DSW	

	BSC	L	WAIT, Z+	branch if PRO- GRAM STOP
TRACE	LD		START	store address of next instruction to be executed in extension
	SRA		16	
	LD		SAVA	
WAIT	WAIT			WAIT for opera- tor action
	LDD		SAVAQ	
	BOSC	I	START	clear interrupt level and exit
SAVAQ	BSS	E	3	
SENSE	DC		/3F00	

INT REQ INTERRUPTS

Recovery from an INT REQ interrupt requires program action. Therefore whether or not recovery is possible depends on the program in operation. See Tables I1 and I2 for the response of the IBM-provided programs and subroutines.

User-written programs, loaded by either the relocating loader or core image loader, fall under the heading "user" if:

1. User program does not alter the interrupt branch address for level 4 (core location 000C), and
2. User program does not alter the interrupt trap (core locations 002E-0031), and
3. User program
 - a. Does not call any ISS requiring ILS04, or
 - b. Does not call an ISS 2 to process Input Keyboard/Console Printer interrupts, or
 - c. Does call an ISS 2 but does not alter the ISS Keyboard Operator exit (core location 002C)

If these conditions are met, an INT REQ will cause a branch to the interrupt trap via location 000C (if 3a is true), from ILS04 (if 3b is true), or via location 002C (if 3c is true).

INTERRUPT TRAP

The interrupt trap is constructed to force operator action if an unexpected interrupt occurs; an unexpected interrupt occurs if:

1. There is no ILS for the interrupt level, or
2. There is no ISS loaded for the interrupt even though there is an ILS for the level, or
3. The ISS Keyboard Operator exit (core location 002C) is not altered by the user.

If the interrupt is on level 5, or if INT REQ assumption 3 c above is true, see recovery procedure 1 on

Tables I1 and I2. Otherwise, recovery involves pressing RESET, which may disrupt the process. See recovery procedure 2 on Tables I1 and I2.

The interrupt trap will work only if interrupt processing is done by a closed subroutine, i. e., one which exits by means of a BSC I or BOSC I, via the return link. Otherwise the effect of an unexpected interrupt is undefined.

Table I1. Card System Interrupt Stop and Restart Procedures

PROGRAM OR ROUTINE NAME	PROGRAM STOP OR INTERRUPT RUN MODE STOP	KEYBOARD INTERRUPT REQUEST
Relocating Loader	Halt at 0012. Press PROGRAM START.	Undefined, restart program.
Core Image Loader	Halt at 0011. Press PROGRAM START.	Undefined, restart program.
Core Image Converters	Halt at 0012. Press PROGRAM START.	Undefined, restart program.
Assembler	Halt at 002E. Recovery 1.	Undefined, restart program.
Compressor	Halt at 002E. Recovery 1.	Undefined, restart program.
FORTRAN Compiler (If halt at 0011, see Core Image Loader)	Halt at 002E. Recovery 1.	Undefined, restart program.
FORTRAN Execution	Halt at 002E. Recovery 1.	Halt at 002E. Recovery 1 if ISS 2 loaded. Recovery 2 if ISS 2 not loaded.
Subroutine Library	Halt at 002E. Recovery 1.	Halt at 002E. Recovery 1 if ISS 2 loaded. Recovery 2 if ISS 2 not loaded.
User	Halt at 002E. Recovery 1.	Halt at 002E. Recovery 1 if ISS 2 loaded. Recovery 2 if ISS 2 not loaded.
Utility	Halt at 002E. Recovery 1.	Ignored, no stop occurs.
Dump	Undefined, restart program.	Undefined, restart program.
Recovery 1 1. Set the console Mode switch to DISPLAY 2. Press PROGRAM START 3. Set the console Mode switch to RUN 4. Press PROGRAM START		Recovery 2 1. Press PROGRAM RESET 2. Set the console Mode switch to LOAD 3. Set the Console Entry switches to 0030 (hexadecimal) 4. Press LOAD IAR 5. Set the console Mode switch to RUN 6. Press PROGRAM START

Table 12. Paper Tape System Interrupt Stop and Restart Procedures

PROGRAM OR ROUTINE NAME	PROGRAM STOP OR INTERRUPT RUN MODE STOP	KEYBOARD INTERRUPT REQUEST
Relocating Loader	Halt at 0012. Press PROGRAM START.	Undefined, restart program.
Core Image Loader	Halt at 0012. Press PROGRAM START.	Undefined, restart program.
Core Image Converters	Halt at 000F. Recovery 1.	Undefined, restart program.
Assembler	Undefined, restart program.	Undefined, restart program.
Compressor	Undefined, restart program.	Undefined, restart program.
FORTRAN Compiler (If halt at 0011, see Core Image Loader)	Halt at 002E. Recovery 1.	Undefined, restart program.
FORTRAN Execution	Halt at 002E. Recovery 1.	Halt at 002E. Recovery 1 if ISS 2 loaded. Recovery 2 if ISS 2 not loaded.
Subroutine Library	Halt at 002E. Recovery 1.	Halt at 002E. Recovery 1 if ISS 2 loaded. Recovery 2 if ISS 2 not loaded.
User	Halt at 002E. Recovery 1.	Halt at 002E. Recovery 1 if ISS 2 loaded. Recovery 2 if ISS 2 not loaded.
Utility	Halt at 002E. Recovery 1.	Ignored, no stop occurs.
Dump	Undefined, restart program.	Undefined, restart program.
<p>Recovery 1</p> <ol style="list-style-type: none"> 1. Set the console Mode switch to DISPLAY 2. Press PROGRAM START 3. Set the console Mode switch to RUN 4. Press PROGRAM START 		<p>Recovery 2</p> <ol style="list-style-type: none"> 1. Press PROGRAM RESET 2. Set the console Mode switch to LOAD 3. Set the Console Entry switches to 0030 (hexadecimal) 4. Press LOAD IAR 5. Set the console Mode switch to RUN 6. Press PROGRAM START

INDEX

- Adding subroutines 19
- ALD card (relocating loader and core image converter) 28
- Arith/Func subroutine core requirements (Table B1) 43
- Arith/Func subroutine times (Table C3) 48
- Assembler (card system) 1
 - Card read punch errors 5
 - Consecutive assembly 4
 - Error detection 4
 - Error detection codes 6
 - Error waits 8
 - Format 3
 - Options 4
 - Pass 1 2
 - Pass 2 2
 - Symbol table, max. size 1
 - Symbol table, oversize 5
 - Uninterrupted assembly 4
- Assembler (paper tape system) 6
 - Checksum error 7
 - Error waits 8
 - Format 7
 - Pass 1 7
 - Pass 2 7
- *ARITHMETIC TRACE 10

- Backspace 19
- Batching 5
- Binary card layout (Appendix F) 54
- Buffer status after keyboard input (FORTRAN) 15

- CALL TSTOP 12
- CALL TSTRT 12
- Card-character definition 31.2
- Card ID (Appendix D) 49
- Card read punch errors, assembler and compressor 5
- Card read punch errors and operator procedures 17
- Card reproducing routine 37
- Card subroutine errors (CARD0 and CARD1) 17
- Card System
 - Assembler 1
 - Assembler options 4
 - Card ID 49
 - Compressor 4
 - Compressor options 4
 - Core image converter 28
 - Core image loader 31
 - FORTRAN compiler loading 9
 - FORTRAN object deck loading 14
 - Interrupt stop and restart procedures 62
 - Relocating loader 26
 - Subroutine card ID 21
- Causes multiple passes (relocating loader) 27
- Checksum (definition) 51
- Checksum restart, paper tape system (Appendix H) 57
- Common Arith/Func calls (Table 4) 22
- Common Arith/Func LIBF's (Table 4) 24
- Common FORTRAN calls (Table 4) 21
- Compilation error messages 12.1
- Compilation messages 12

- Compiler loading, card system 9
- Compressed deck format 4
- Compressed tape format 8, 58
- Compressed tape data record 58
- Compression of oversize decks 4.1
- Compression of oversize program 5
- Compressor (card system) 1
 - Card read punch errors 5
 - Options 4
 - Oversize decks 4, 5
 - Procedures 4
- Compressor (paper tape system) 7
 - Consecutive assembly 4
- Console printer-character definition 32
- Console printer subroutine errors (TYPE0 and WRTY0) 19
- Console restart procedures (Appendix I) 59
- Console Routine 37
- Construct paper tape routine 39
- Control card format, utility I/O 32
- Control record options (FORTRAN) 10
- Conversion routines (Table 4) 24
- Conversion subroutine core requirements (Table B3) 44
- Conversion times (Table C1) 46
- Core image converter, card system 28
 - Core image converter, card system, Pass 1 29
 - Core image converter, card system, Pass 2 30
 - Core image converter, paper tape system 31.1
 - Core image converter, paper tape system, Pass 1 31.1
 - Core image converter, paper tape system, Pass 2 31.1
 - Core image converter, programming notes 30, 31, 31.2
- Core image loader, card system 31
 - Core image loader, paper tape system 31.2
- Core image paper tape record format 31.2
- Core map (Appendix A) 42
- Core map option (core image converter) 30, 31.1
- Core requirements (Appendix B) 43

- Data card 51
- Data card format (core image loader) 30
- Disk pack initialization routine (DPIR) 38
- DPIR halt addresses (Table 8) 38
- Dump between limits 34
- Dump routines 34
- Dump routine card ID (Table D2) 50

- EOD Loader Cards 26, 28
- EOP card 51
- Erase field 19
- Error and program wait table (Table G1) 55
- Errors
 - Assembler card 5
 - Assembler error detection 4
 - Assembler, paper tape 8
 - CARD0, CARD1 subroutine 17
 - Checksum error, paper tape assembler 7
 - Checksum error, paper tape restart 57
 - Compiler card load (FORTRAN) 9
 - Compressor, card 5
 - Compressor, paper tape 8
 - Console printer (TYPE0, WRTY0) 19

- DPIR halt addresses 38
- Error and program wait table 55
- Error detection codes, card system assembler 6
- FORTTRAN compilation error codes 12.1
- FORTTRAN compilation error messages 12.1
- FORTTRAN I/O 15
- FORTTRAN Paper Tape Load 16
- Input/Output routine 34
- ISS pre-operative 17
- 1442 operator procedures 17
- Execute card format (core image loader) 30
- Extended Arith/Func calls (Table 4) 21
- Extended Arith/Func LIBF's (Table 4) 23
- *EXTENDED PRECISION 10
- Formats
 - ALD card (relocating loader and core image converter) 26
 - Assembler card 3
 - Assembler, paper tape 7
 - Compressed deck (compressor) 4
 - Compressed tape (compressor) 8
 - Compressed tape data record 58
 - Control card, utility I/O 32
 - Core image paper tape record 31.1
 - Data card (core image loader) 30
 - Execute card (core image loader) 30
 - FORTTRAN source record 16.1
 - List deck (assembler) 3
 - Object program data record (FORTTRAN) 15
 - Relocatable (compressor and FORTTRAN compiler) 51
 - Symbolic assembly card (assembler) 3
- FORTTRAN 9
 - Buffer status after keyboard input 15
 - Card load errors, compiler 9
 - Compilation error codes 12.1
 - Compilation error messages 12.1
 - Compilation messages 12
 - Compiler loading, card system 9
 - Compiler loading, paper tape system 16
 - Control record options 10
 - I/O errors 15
 - I/O logical unit definitions 9
 - Keyboard input of data records 15
 - Object deck loading 14
 - Object program load, paper tape 16
 - Option control cards 10
 - Printouts 12
 - Source record format 16
 - 1800 system control record options 12
- FORTTRAN common LIBF's (Table 4) 23
- FORTTRAN extended trace routines (Table 4) 22
- FORTTRAN I/O and conversion routines (Table 4) 23
- FORTTRAN I/O, extended (Table 4) 22
- FORTTRAN I/O, standard (Table 4) 22
- FORTTRAN sign transfer calls (Table 4) 21
- FORTTRAN standard trace routines (Table 4) 22
- Full dump on 1132 printer 35
- I/O character and record format 31.2
 - *IOCS (1130 system) 10
 - *IOCS (1800 system) 12
- I/O logical unit definitions (FORTTRAN) (Table 2) 9
- ILS header card 52
- ILS05 sample programs 60
- IMM STOP 60
- Interrupt level subroutines (Table 4) 25
- Interrupt priority levels (1130) 59
- INT REQ interrupts 61
- INT REQ (keyboard) 59
- Interrupt restart procedures (Appendix I) 59
- INT RUN (console Mode switch setting) 59
- INT RUN interrupts 61
- Interrupt service subroutines (Table 4) 24
- Interrupt stop and restart procedures, card system (Table I1) 62
- Interrupt stop and restart procedures, paper tape system (Table I2) 62
- Interrupt trap 61
- ISS header card 52
- ISS/ILS core requirements (Table B2) 44
- ISS pre-operative errors (subroutine library) 17
- ISS times (Table C2) 47
- ISTV table 52
- Keyboard input of data records (FORTTRAN) 15
- Keyboard routine 36
- Keyboard subroutine functions (TYPEO) 19
- *LIST ALL 10
- *LIST control cards (operating notes) 11
- List deck card format (Figure 5) 3
- *LIST SOURCE PROGRAM 10
- *LIST SUBPROGRAM NAMES 10
- *LIST SYMBOL TABLE 10
- Loader card ID (Table D2) 50
- Loader cards (EOD) 26, 28
- Loader overlay cards 52
- Loading routines 26
- Machine requirements ii
- Mainline header card 51
- *MULTIPLE DEVICE (1800 system) 12
- Name tape (construct paper tape routine) 40
- *NAME XXXXXX 10
- Object deck loading (FORTTRAN) 14
- Object program data record format (FORTTRAN) 15
- *ONE-WORD INTEGERS 10
- Operator procedures, 1442 errors 17
- Optional tracing (FORTTRAN) 12
- Options
 - Assembler, card system 4
 - Compressor, card system 4
 - Core map 30, 31.1
 - FORTTRAN control record 10
 - Optional tracing, FORTTRAN 12
 - User's exit - utility I/O 32
 - Utility routine (Table 5) 32
 - 1800 system control record 12
- PACH 4.1
 - Paper tape -- character definition 32
 - Paper tape ID (Appendix D) 49
 - Paper Tape Load Errors (FORTTRAN) 16
 - Paper tape reader error (checksum) restart procedure (Appendix H) 57

Paper tape subroutine (PAPT1, PAPTN)	19
Paper tape system	
Assembler	6
Checksum error, assembler	7
Compressor	7
Core image converter	31.1
Core image loader	31.2
FORTRAN compiler loading	16
FORTRAN object program loading	16.1
Interrupt stop and restart procedures	62
Load errors (FORTRAN)	16
Paper tape ID	49
Relocating loader	31
Subroutine paper tape ID	21
Printer core dumps (1132)	35
PROGRAM STOP	59
PROGRAM STOP interrupts	60
Program times (Appendix B)	43
Relocatable card and paper tape formats (Appendix E)	51
Relocating loader, card system	26
Relocating loader, paper tape system	31
Removing subroutines	19
RESET	60
Sample Programs (Appendix C)	45
*SAVE LOADER	11
Selective dump on 1132 printer	35
Source record format (FORTRAN)	16.1
Standard Arith/Func calls (Table 4)	22
Standard Arith/Func LIBF's (Table 4)	23
Subroutine card ID (Table 4)	21
Subroutine header card	52
Subroutine Library	17
Adding subroutines	19
ISS pre-operative errors	17
Removing subroutines	19
Subroutine card ID	21
Subroutine library listing	20
Subroutine paper tape ID	21
Subroutine library listing	20
Subroutine paper tape ID (Table 4)	21
Symbol table, maximum size	1
Symbol table, oversize	5
Symbolic Assembly Program Card (Figure 6)	3
*TRANSFER TRACE	11
Uninterrupted assembly	4
User's exit option, utility I/O	32
Utility calls (Table 4)	21
Utility I/O control card format	32
Utility Input/Output routine error checks	34
Utility Input/Output routine, operating instructions	34
Utility routine options (Table 5)	32
Utility Routines	26
Card reproducing	37
Console	37
Construct paper tape	39
Disk pack initialization	38
Dump	34
Keyboard	36
Loading routines	26
Utility input/output	31.2
1132 Printer -- character definitions	32
1132 Printer core dumps	35
1132 Printer STOP (key)	60
1442 Card Read Punch STOP (key)	60
1442 errors and operator procedures	17
1800 system control record options	12

File Number 1130-20

Re: Form No. C26-3629-0

This Newsletter No. N26-0556

Date September 1, 1966

Previous Newsletter Nos. None

IBM 1130 Card/Paper Tape Programming System Operator's Guide (C26-3629-0)

The attached pages bring the above publication up to date. Changes are indicated by a vertical line at the left of affected text, a bullet (●) at the left of the title of a changed illustration, and a bullet beside the page number of a page that should be reviewed in its entirety. Pages that contain changes are coded in the upper outside corner.

Replace the following pages:

iii and iv	31 and 32
3 and 4	37 and 38
9 and 10	39 and 40
11 and 12	43 and 44
15 and 16	45 and 46
21 and 22	47 and 48
23 and 24	49 and 50
25 and 26	63 and 64
27 and 28	65 and 66
29 and 30	

Add the following pages:

4.1 and blank	31.1 and 31.2
12.1 and blank	48.1 and 48.2
16.1 and blank	48.3 and blank

Make the following changes on the pages indicated:

Page 6, Col. 1, Change the first sentence under PAPER TAPE ASSEMBLER PROGRAM to read as follows:

The paper tape assembler is supplied to the user as self-loading binary tape, approximately 40 feet in length.

Page 8, Col. 2, Under ERROR CONDITIONS, delete the following sentence:

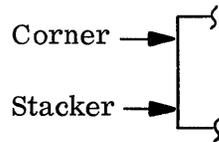
See Appendix H for checksum restart procedure.

Page 17, Col. 1, Change to read as follows:

Line 4	Descriptions of the subroutines, their timing and core requirements, and the methods for
Line 22	The IAR displays the address 002A.

Page 18, Col. 1,

Under Transport, change the illustration to read:



Error Indicator: TRANS

Col. 2,

Under Feed Cycle, Feed Check (read station) and Read Registration, change corner stacker location as illustrated above.

Page 19, Col. 1,

Under Punch Check change corner stacker location as illustrated above.

Page 20, Col. 2,

Change BP-09 to BP01. Change BP-10 to BP02.
Change BP-11 to BP03.

Page 33, Col. 2,

Change callout on Figure 12 to read as follows:
(For Tape Systems, BP01 includes the above subroutines)

Page 41, Col. 1,

After line 28, DEL characters, add the following:
NOTE: If the delete type C option is selected, the EOD1 and EOD2 tapes are specially handled. The type D and type E records are rechecksummed to prevent core image converter checksum waits.

Page 42, Col. 2,

Change line 8 to read as follows:
IND 0000 FAC 0000 printed as the last entry

Page 51, Col. 2,

Change next-to-last line under EOP Card to read as follows:
4 XEQ address, if mainline program

Page 52, Col. 2,

Change the first line under ILS Header Card to read as follows:
An ILS (interrupt level subroutine) header card iden-

Change the last two lines under Loader Overlay Cards to read as follows:
generated for each subroutine. EOD1 and EOD2 cards are also loader overlay cards. The loader overlay card format is as follows:

NOTE: Subroutine timing and core requirements, removed from the Operator's Guide by this Technical Newsletter, were incorporated in IBM 1130 Subroutine Library (Form C26-5929-2) by Technical Newsletter N26-0557.

File this Newsletter at the back of the manual. It will provide a reference to changes, a method of determining that all amendments have been received, and a check for determining if the manual contains the proper pages.