Burroughs
B 1700 SYSTEMS

Report Program Generator
REFERENCE MANUAL

Burroughs

B 1700
Systems

Report Program Generator

REFERENCE MANUAL

Burroughs Corporation
Detroit, Michigan 48232

$4.00

Printed in U.S. America 1-73 1057189

AA 349022

Burroughs Corporation believes the program described in this manual to be accurate and reliable, and much care has been taken in its preparation. However, the Corporation cannot accept any responsibility, financial or otherwise, for any consequences arising out of the use of this material. The information contained herein is subject to change. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this document should be forwarded using the Remarks Form at the back of the manual, or may be addressed directly to Systems Documentation, Sales Technical Services, Burroughs Corporation, 6071 Second Avenue, Detroit, Michigan 48232.

Table of Contents

Table of Contents (cont)

Table of Contents (cont)

Section Title Page

Table of Contents (cont)

Table of Contents (cont)

Section Title Page

Table of Contents (cont)

Table of Contents (cont)

List of Illustrations

List of Illustrations (cont)

List of Tables

INTRODUCTION

Burroughs RPG (Report Program Generator) is a machine-independent source language used mainly for writing commercial report application programs. Its syntax includes, as dialects, most constructs allowed by IBM's 360/20 RPG and System/3 RPG II, and UNIVAC's 9000 Series RPG, as well as extensions unique to Burroughs RPG.

Burroughs RPG allows any program written in other manufacturer's RPG or RPG II to be processed by the RPG Compiler and to produce similar output when the resultant program is executed (within the limitations of peripheral devices).

To a computer user Burroughs RPG offers the following major advantages:

a. Expeditious means of program implementation.

b. Accelerated programmer training and simplified retraining requirements.

c. Reduced conversion costs when changing from a computer of one manufacturer to that of another.

d. Significant ease of program modification.

e. Standardized documentation.

f. Significant ease of program conversion to COBOL (refer to appendix H).

This manual provides a complete description of Burroughs RPG as implemented for use on the B 1700 Series System. It is intended as a reference for the RPG Specification Forms and contains a detailed description of each field on the forms together with the allowable entries for each field. Examples are provided to give the programmer direct insight into the practical requirements of problem-solving with Burroughs RPG.

SECTION 1

REPORT PROGRAM GENERATOR (RPG) OPERATION

GENERAL.

A program written in Burroughs RPG, called a source program, is accepted as input by the RPG Compiler. The compiler first verifies that all rules outlined in this manual are satisfied, then converts the source program language into COBOL S-Language, which is then ready for execution on the system. The S-Language generated by the compiler can be executed under control of the Master Control Program (MCP) by using the COBOL Interpreter. The Interpreter causes the system hardware to perform the operations specified by the S-Language Program and, thus, by the programmer who wrote the source program.

The relationships involved between the source program, compiler, S-Language, Interpreter, and hardware is illustrated on the following page (see figure 1-1).

For a more detailed description of the function of the S-language as it relates to the Interpreter and the hardware, refer to the System reference Manual.

RPG_SOURCE_PROGRAM.

A RPG Source Program is divided into seven parts which must appear in the following order:

a. Control card specifications.
b. File description specifications.
c. Extension specifications.
d. Line counter specifications.
e. Input specifications.
f. Calculation specifications.
g. Output-format specifications.

A description of the above specifications is contained in the paragraphs that follow.

CONTROL CARD SPECIFICATIONS.
These specifications provide certain information about the program to the B 1700 RPG Compiler.

FILE DESCRIPTION SPECIFICATIONS.
These specifications provide information about the equipment being used, and associate files with the hardware devices that will be used.

EXTENSION SPECIFICATIONS.
These specifications are used to describe tables and arrays that will be used with the program.

LINE COUNTER SPECIFICATIONS.
These specifications provide information about the number of lines to be printed on each page of the output forms that are used.

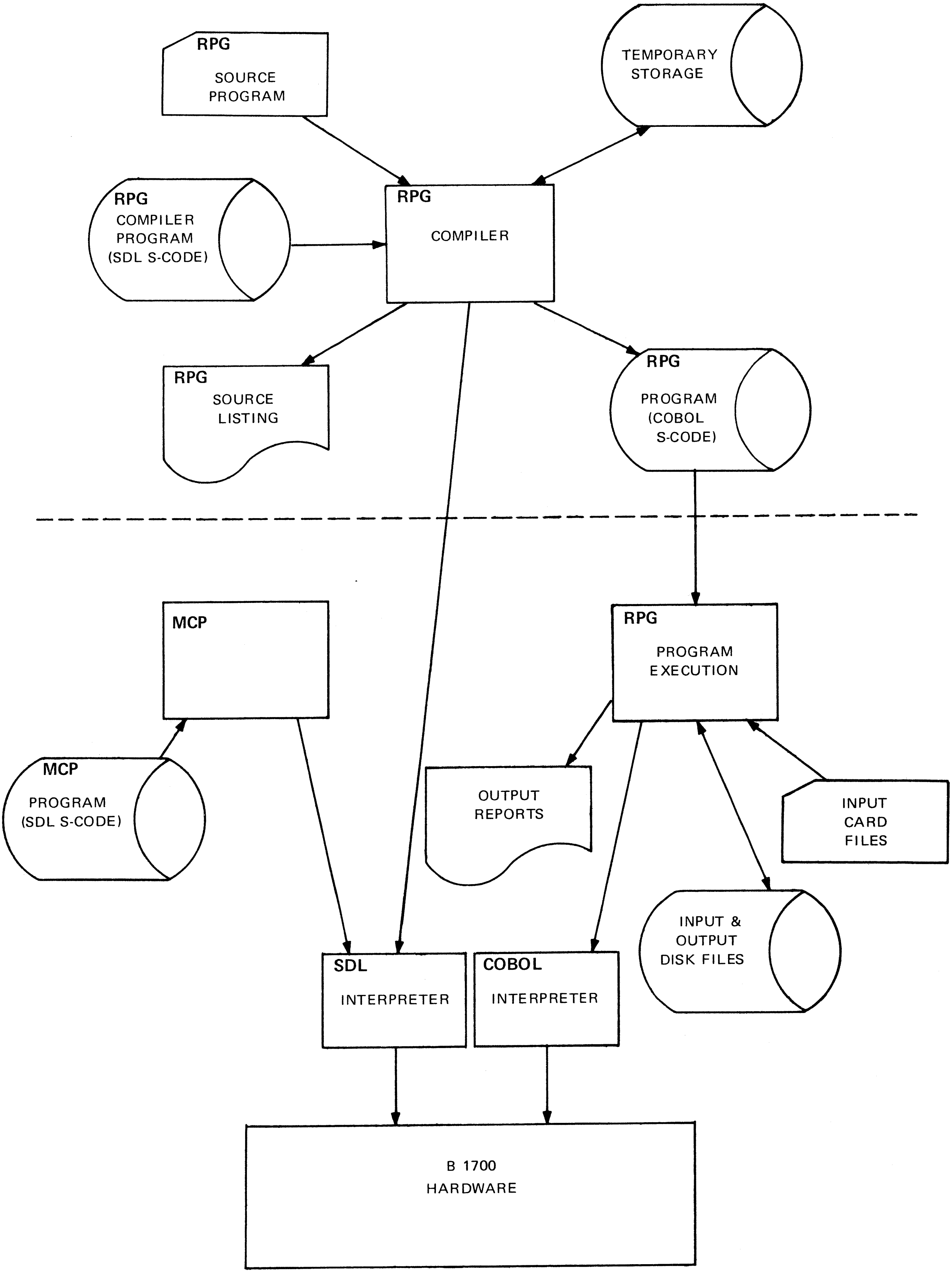


Figure 1-1. Relationship of Hardware and Programming

INPUT SPECIFICATIONS.

These specifications are used to describe the record layouts of all input files used by the program.

CALCULATION SPECIFICATIONS.

These specifications define the steps necessary to accomplish the desired task when operating on data described in the program.

OUTPUT-FORMAT SPECIFICATIONS.

These specifications are used to specify the type and arrangement of data that will be written as output from the program.

The above specifications are coded using the following Report Program Generator forms:

a. Control Card Specifications/File Description Specifications (form 1057908).

b. Extension Specifications/Line Counter Specifications (form 1057924).

c. Input Specifications (form 1057932).

d. Calculation Specifications (form 1057940).

e. Output-Format Specifications (form 1057957).

DOLLAR CARD SPECIFICATIONS.

The Dollar Card ($ Card) Specifications (form 1057916) sheet is also defined for Burroughs RPG. These specifications are used to accommodate machine- and system-dependent features, and may appear anywhere in a RPG Source Program.

RPG SOURCE PROGRAM DECK.

The coded information contained on the specification forms is recorded in punched cards which constitute the source program. The arrangement of these cards in the source deck is illustrated below (see figure 1-2).

Should source corrections become necessary, appropriate changes can be made and the program recompiled. Thus, the source program deck always reflects the S-Language being operationally executed.

RPG FORMAT.

Burroughs RPG follows other manufacturer's RPG and RPG II format with the following exceptions:

a. Control card format requires recoding before running other manufacturer's RPG or RPG II Programs through the Burroughs RPG Compiler.

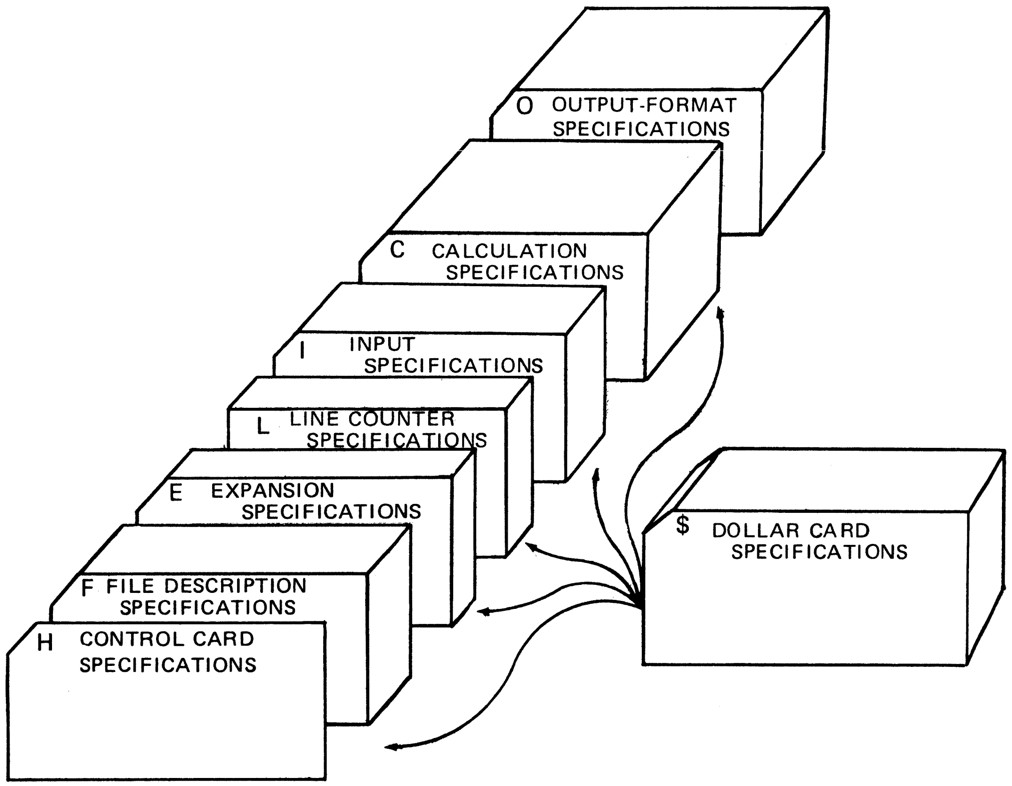

O OUTPUT-FORMAT SPECIFICATIONS
C CALCULATION SPECIFICATIONS
I INPUT SPECIFICATIONS
L LINE COUNTER SPECIFICATIONS
E EXPANSION SPECIFICATIONS
F FILE DESCRIPTION SPECIFICATIONS
H CONTROL CARD SPECIFICATIONS
$ DOLLAR CARD SPECIFICATIONS

Figure 1-2. RPG Source Program Deck

b. The following features and language constructs found in other manufacturer's RPG and RPG II are not supported in Burroughs RPG:

1) Binary data format.
2) Sterling data format.
3) Alternate collating sequence.
4) Record address files.
5) Inquiry programs.
6) Bit operators (BITON, BITOF, TESTB).
7) Printer keyboard output files.
8) External assembler subroutines.
9) Blank after literals.
10) Redefined field lengths.
11) Factor 1 and Factor 2 both literals.

For IBM's RPG, the maximum alpha field length is 256 characters, whereas in Burroughs B 1700 RPG it is 511 characters.

IBM's RPG uses standard right signed numeric fields, whereas Burroughs B 1700 RPG uses standard left sign.

IBM's System 3 standard plus sign is 'F' right sign, the 360/20 standard plus sign is 'C' right sign, whereas the B 1700 standard plus sign is 'C' left sign.

Numeric fields must be edited if the C zone punch is not wanted on output, otherwise a positive unedited numeric field in the form of C subscripted one (1) equals A in the field C1F0F1F3 will appear as A013 on printer output. Editing will assign a F to the leftmost character making the output appear as 1013.

SECTION 2

RPG LANGUAGE ELEMENTS

GENERAL.

Burroughs RPG is a programming language based upon a fixed series of events, called the RPG "program cycle", which take place during program execution. Due to the strict limitations of the generated program, the source language must conform to very rigid rules of syntax. The paragraphs and sections that follow define the rules for writing programs using the RPG language.

CHARACTER SET.

The RPG character set consists of the following 64 characters:

0-9	
A-Z	
	blank or space
&	ampersand
.	period or decimal point
<	less than sign
(	left parenthesis
+	plus sign
-	minus sign
|	logical OR
!	exclamation point
$	dollar sign
[	left bracket
*	asterisk
)	right parenthesis
;	semicolon
¬	logical NOT
/	slash (virgule)
,	comma
%	percent sign
_	underscore
>	greater than sign
?	question mark
:	colon
#	pound sign
@	at sign
'	apostrophe
=	equal sign
"	quotation mark
]	right bracket

CHARACTERS USED FOR NAMES.

The character set used to form names consists of the 36 characters: 0 through 9 and A through Z.

CHARACTERS USED FOR EDITING.

The character set used for special purposes within edit words in the Output-Format Specifications consists of the following nine characters:

	blank or space
0	zero
$	dollar sign
.	decimal point
,	comma
CR	CRedit symbol (two characters)
*	asterisk (check protect)
&	ampersand
-	minus sign

DEFINITION OF NAMES.

A name must be left-justified in the field, must begin with an alphabetic character, and is ended by a space or the end of the field, whichever comes first. All characters in a name except the first may be any combination of alphabetic and numeric characters (special characters are not allowed). Blanks may not appear between the characters in a name.

RPG defines the following four types of names:

a. File names.
b. Vector names (table or array names).
c. Field names (variable names).
d. Labels.

FILENAMES.

A filename is a collective name or word that designates a set of data items. The contents of a file are divided into logical records that are, in turn, made up of any consecutive set of data items. Filenames cannot exceed eight characters and must be unique in the first seven characters.

VECTOR NAMES.

A vector name is used to identify a data item which is actually a table or an array. The table will be loaded with a number of elementary data items which are accessed through use of the vector name that identifies the entire table. Vector names cannot exceed six characters and must be unique.

FIELD NAMES.

A field name is used to identify an individual element of data. Field names cannot exceed six characters and must be unique.

LABELS.

A label is only used to identify a point in the Calculation Specifications to which a GOTO operation will branch, or to identify the beginning of a subroutine. Labels cannot exceed six characters and must be unique.

DEFINITION OF LITERALS.

A literal is an item of data which contains a value identical to the characters being described. There are two classes of literals: numeric and non-numeric.

NUMERIC LITERAL.
A numeric literal is defined as an item composed of characters chosen from the digits 0 through 9, the plus (+) or minus (-) sign, and the decimal point. The rules for the formation of a numeric literal are:

a. There must be at least one digit in a numeric literal.

b. The sign of a numeric literal must appear as the left-most character. If no sign is present, the literal is defined as a positive value.

c. Only one sign character and/or one decimal point may be contained in a numeric literal.

d. The maximum total length of a numeric literal is 10 characters, including sign and decimal point.

The following are examples of numeric literals:

```
  13247
   .005
 +1.808
 -.0968
7894.54
```

NON-NUMERIC LITERAL.
A non-numeric literal may be composed of any allowable character. The beginning and end of a non-numeric literal is denoted by an apostrophe. Any character enclosed within apostrophes is part of the non-numeric literal. Consequently, all spaces enclosed within the apostrophes are considered part of the literal. Two consecutive apostrophes within a non-numeric literal cause a single apostrophe to be inserted into the literal string. Four consecutive apostrophes results in a single ' literal. The rules for the formation of a non-numeric literal are:

a. There must be at least one character in a non-numeric literal.

b. The maximum length of a non-numeric literal used in the Calculation Specifications is eight characters, and in the Output-Format Specifications is 24 characters.

c. Non-numeric literals may not be used for arithmetic operations.

The following are examples of non-numeric literals:

Literal on Source Program Level	Literal Stored by Compiler
'ACTUAL'	ACTUAL
'-1234.56'	-1234.56
'WEEK''S'	WEEK'S
'TODAY''S DATE'	TODAY'S DATE

Literal_on_Source_Program_Level	Literal_Stored_by_Compiler
''''	'
'A''B'	A'B

DEFINITION_OF_RESERVED_WORDS.

Reserved words have a specific function in the RPG syntax and are of two types: special names, and operation codes.

SPECIAL NAMES.
These reserved words are only used in the Input, Calculation, and Output-Format Specifications, and specify such things as page numbering, date fields, card interpreting, and so forth. Special names are discussed fully in the sections where they are used.

OPERATION CODES.
These reseved words are used in the Calculation Specifications, and specify operations to be performed upon data items. A complete description of the operation codes is presented in appendix C.

COMMON_ENTRIES.

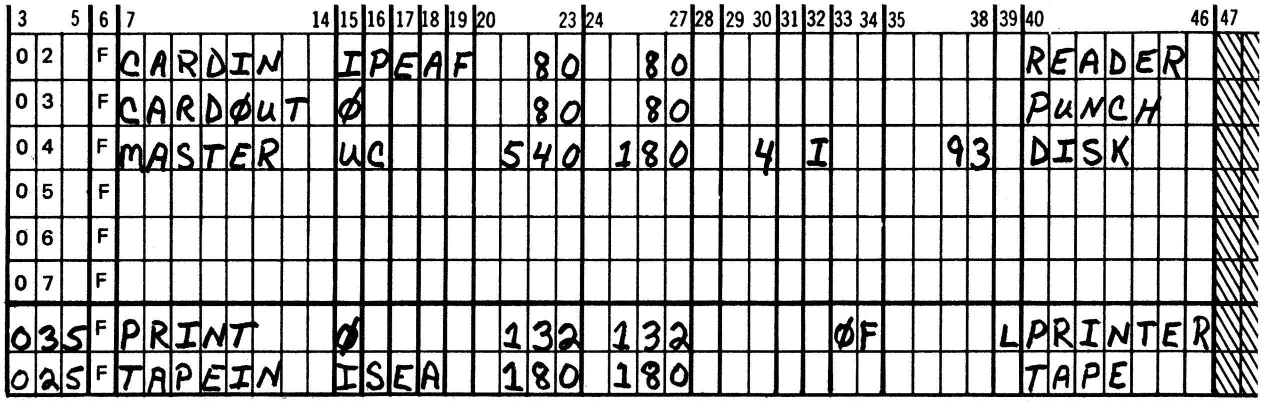
The RPG specification forms have certain common fields which have consistent entries within a RPG Program. These fields and their respective entries are described below, so that they need not be repeated in subsequent sections.

PAGE (COLUMNS 1-2).
The PAGE entry in the upper left-hand corner of each specification form is used to number the coding forms sequentially for each source program. Normally, only numeric characters in the range 01-99 would be used, however, any EBCDIC characters (including blanks) are valid entries.

LINE (COLUMNS 3-5).
The LINE entry is used to number the individual lines on each specification form sequentially. Normally, only numeric characters in the range 000-999 would be used, however, any EBCDIC characters (including blanks) are valid entries.

Columns 3-4 are preprinted so that in most cases the entry is already made. For example, the File Description Specifications form contains line numbers for lines 02 through 07. If more than six lines are needed, additional entries may be made below line 07 and numbered 08, 09, etc.

The units position (column 5) may be used to insert a line between two previously written lines (see figure 2-1).



3 5	6	7 14	15	16	17	18	19	20 23	24 27	28	29 30	31	32	33 34	35 38	39	40 46	47
02	F	CARDIN	I	P	E	A	F	80	80								READER	
03	F	CARDØUT	Ø					80	80								PUNCH	
04	F	MASTER	U	C				540	180		4		I		93		DISK	
05	F																	
06	F																	
07	F																	
035	F	PRINT	Ø					132	132					ØF		L	PRINTER	
025	F	TAPEIN	I	S	E	A		180	180								TAPE	

Figure 2-1. Insertion of Coding Lines

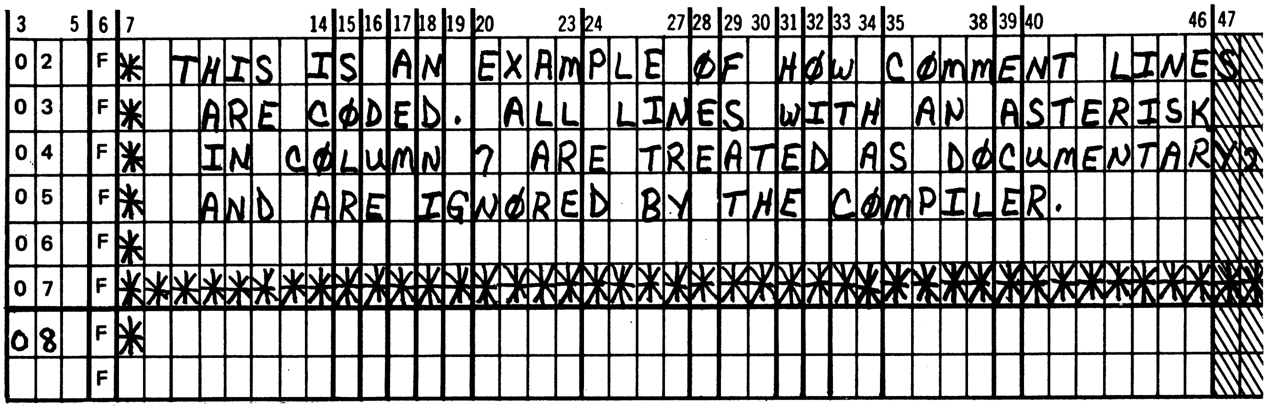
The compiler sequence-checks all input cards on columns 1-5 so that all lines should be numbered in ascending numerical order. If a sequence number (page and line number) is encountered that is equal to or less than the preceding sequence number, a sequence error warning message (S) will be emitted by the compiler.

FORM TYPE (COLUMN 6).
Column 6 contains a pre-printed entry on all the specification forms. This entry identifies the type of specification for each line of coding. Valid entries for this field are:

Entry	Definition
Blank	Comment Card or Dollar Card (depending upon entry in column 7).
H	Header Card (Control Card Specification).
F	File Description Specification.
E	Extension Specification.
L	Line Counter Specification.
I	Input Specification.
C	Calculation Specification.
O	Output-Format Specification.

COMMENTS/DOLLAR CARD (COLUMN 7).
Since it is often necessary to write explanatory statements within the source program, the Comment Card allows the entire line to the right of

of column 7 (which contains an asterisk, "*") to be produced on the source program listing for documentation clarity. Comments are not instructions to the RPG Program or compiler, but only serve as a means of including program documentation. Any valid EBCDIC characters may be used in a comment line. An example of comment line coding is illustrated below (see figure 2-2).



```
3   5 6 7      14 15 16 17 18 19 20  23 24  27 28 29 30 31 32 33 34 35  38 39 40  46 47
0 2   F* THIS IS AN EXAMPLE ØF HØW CØMMENT LINES
0 3   F*  ARE CØDED. ALL LINES WITH AN ASTERISK
0 4   F*  IN CØLUMN 7 ARE TREATED AS DØCUMENTARY
0 5   F*  AND ARE IGNØRED BY THE CØMPILER.
0 6   F*
0 7   F*****************************************
0 8   F*
      F
```

Figure 2-2. Comment Line Coding

The $ specification in column 7 designates the line to be a Dollar Card Specification which allows the RPG Compiler to accommodate machine- and system-dependent features. Dollar Cards may appear at any point in the RPG Source Program Deck. Section 10 describes the use of these specifications.

PROGRAM IDENTIFICATION (COLUMNS 75-80).
The PROGRAM IDENTIFICATION entry in the upper right-hand corner of each specification form is ignored by the compiler, but will appear on the source program listing. Thus, it can be used for documentation to identify different portions of the program, if desired.

SECTION 3

CONTROL CARD SPECIFICATIONS

GENERAL.

RPG can use a special card, called a Control Card, to provide certain information about the program to the RPG Compiler. The Control Card can transmit to the compiler such information as type of source input, whether debugging is to take place, sign positions, and so forth. If columns 7 through 74 are blank, the Control Card is not required. One line is provided on the Control Card Specifications and File Description Specifications form for coding of the Control Card information (see figure 3-1).

CONTROL CARD SPECIFICATIONS.

The fields for the Control Card Specification form are defined in the paragraphs that follow.

PAGE (COLUMNS 1-2).
Refer to section 2 for a complete description.

LINE (COLUMNS 3-5).
Refer to section 2 for a complete description.

FORM TYPE (COLUMN 6).
This field must be coded with the letter H.

DEBUG (COLUMN 15).
Column 15 specifies whether or not the DEBUG operation is to be significant during program compilation. To perform the DEBUG operation:

a. Column 15 of the Control Card must be coded with a 1.

b. The operation code for DEBUG must appear in the Calculation Specifications.

If this field is left blank, all DEBUG operations encountered in the Calculation Specifications are ignored during compilation.

SIGN POSITION (COLUMNS 17-18).
This field specifies the location of the sign in numeric data items and may have the following entries:

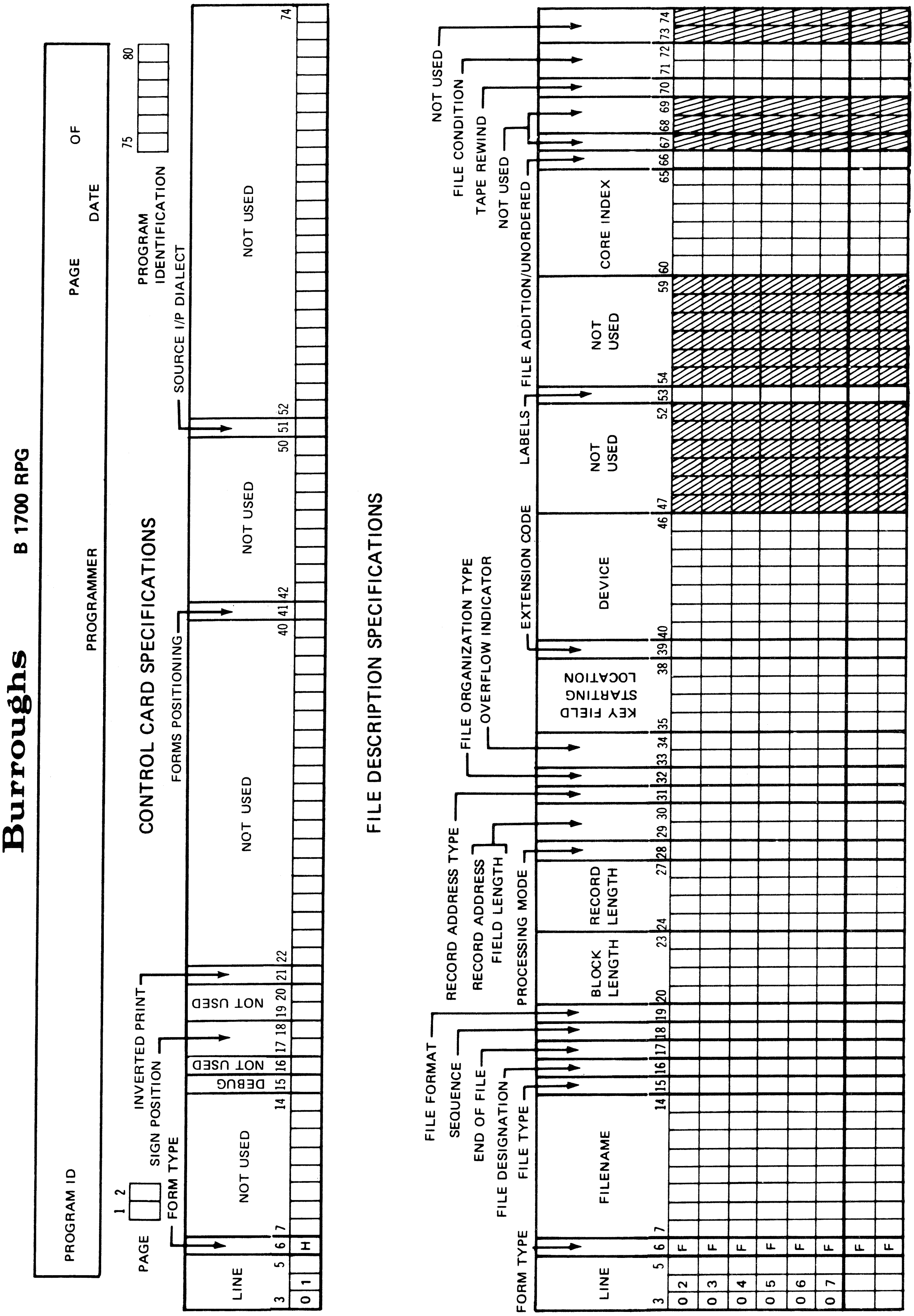


Figure 3-1. Control Card and File Description Specification Form

Entry	Definition
Blank or L	Sign in left-most (high-order) character position.
R	Sign in right-most (low-order) character position.

The sign position specified by this field may be overridden by the RSIGN Dollar Card Specification (refer to section 10) if different sign positions are required within the program.

INVERTED PRINT (COLUMN 21).
This field is used to specify the type punctuation to be used for numeric literals in the Calculation Specifications, the order of the system date field, and the edit codes used on output. The valid codes are:

Entry	Definition
Blank	Domestic format.
I	International format.
J	International format (leading zeros not suppressed for zero balances).
D	Sterling format.

FORMS POSITIONING (COLUMN 41).
Since it sometimes becomes necessary to align special forms in the line printer before beginning to print a report, the FORMS POSITIONING option allows all output lines conditioned by the 1P indicator to be printed more than once when the program is executed. After each time the lines are printed, the program is temporarily suspended to allow the operator to reposition the forms. Printing of the lines may be requested by the operator as many times as necessary to align the forms properly.

If this field is left blank, 1P lines are printed only once.

SOURCE INPUT DIALECT (COLUMN 51).
Because of the differences that arise between RPG and RPG II specification entries in some places, this option specifies the type of input that the compiler should expect for processing. If coded with a 1, this field specifies RPG source program input of other manufacturers. If this field is left blank, it specifies RPG II source program input.

NOTE

The sections that follow are written for RPG II specifications and Burroughs RPG extensions. For differences arising in IBM's RPG, refer to appendix G.

PROGRAM IDENTIFICATION (COLUMNS 75-80).
Refer to section 2 for a complete description.

SECTION 4

FILE DESCRIPTION SPECIFICATIONS

GENERAL.

Every file to be used by a RPG Program must be described to the compiler through the File Description Specifications. Information such as file-names, hardware devices, record and block lengths, and how the file is to be used, are provided by these specifications.

The lower portion of the Control Card Specifications and File Description Specifications(form 1057908) is used for coding the File Description information. Each file to be described requires one line on the form.

FILE DESCRIPTION SPECIFICATIONS.

Coding examples of File Description Specifications and related specifications for Dollar Card, Input Calculation, and Output-Format are illustrated below (see figure 4-1).

FILE CREATION

FILE DESCRIPTION SPECIFICATIONS

LINE (3-5)	FORM TYPE (6)	FILENAME (7-14)	FILE TYPE (15)	FILE DESIGNATION (16)	END OF FILE (17)	SEQUENCE (18)	FILE FORMAT (19)	BLOCK LENGTH (20-23)	RECORD LENGTH (24-27)	PROCESSING MODE (28)	RECORD ADDRESS FIELD LENGTH (29-30)	RECORD ADDRESS TYPE (31)	FILE ORGANIZATION TYPE (32)	OVERFLOW INDICATOR (33-34)	KEY FIELD STARTING LOCATION (35-38)	EXTENSION CODE (39)	DEVICE (40-46)	NOT USED (47-52)	LABELS (53)	NOT USED (54-59)	CORE INDEX (60-65)	FILE ADDITION/UNORDERED (66)	NOT USED (67-69)	TAPE REWIND (70)	FILE CONDITION (71-72)	NOT USED (73-74)
02	F	INPUT	I	P	E		F	80	80				2				READER	▨		▨			▨			▨
03	F	DIRECT	W	C			F	430	10				2				DISK	▨		▨			▨			▨
04	F	INDEXED	W	C			F	170	17		4		I		11		DISK	▨		▨	400		▨			▨

PAGE (1 2) □□

DOLLAR CARD SPECIFICATIONS

PROGRAM IDENTIFICATION (75-80) □□□□□□

LINE (3-5)	FORM TYPE (6)	$ OPTION (7)	NOT (8)	KEY WORD (9-14)	VALUE (15-24)	COMMENTS (25-74)
01		$		FAMILY	MULTIDISK	MULTIFILE ID
02		$		FILEID	DIRECT	FILE ID
03		$		AREAS	5	NUMBER ØF AREAS ALLØCATED
04		$		RPERA	111	NUMBER ØF RECØRDS PER AREA
05		$				
06		$		FAMILY	INDEXFILE	MULTIFILE ID
07		$		FILEID	INDEXED	FILE ID
08		$		AREAS	5	NUMBER ØF AREAS ALLØCATED
09		$		RPERA	50	NUMBER ØF RECØRDS PER AREA

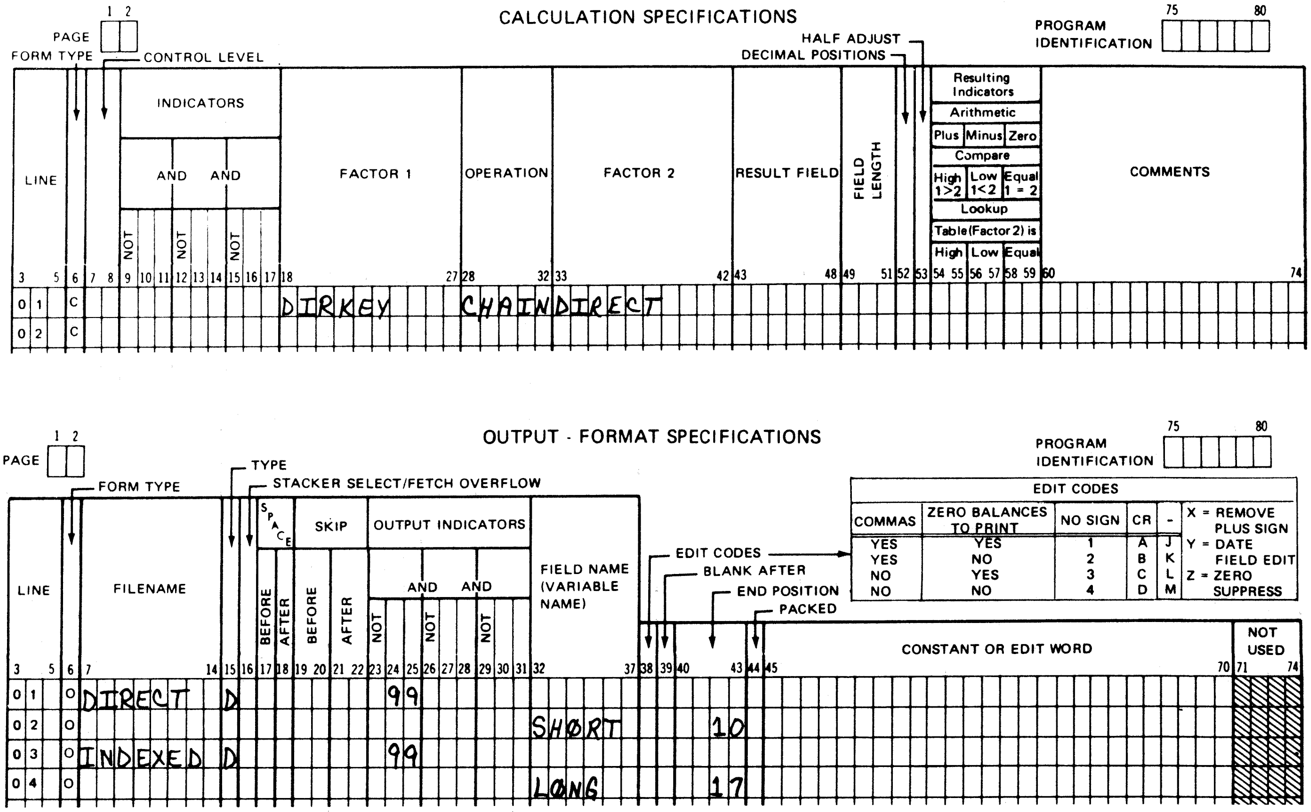
PAGE (1 2) □□

INPUT SPECIFICATIONS

PROGRAM IDENTIFICATION (75-80) □□□□□□

LINE (3-5)	FORM TYPE (6)	FILENAME (7-14)	SEQUENCE (15-16)	NUMBER (17)	OPTION (18)	RECORD IDENTIFYING INDICATOR (19-20)	RECORD IDENTIFICATION CODES 1: POSITION (21-24)	NOT (N) (25)	C/Z/D (26)	CHARACTER (27)	2: POSITION (28-31)	NOT (N) (32)	C/Z/D (33)	CHARACTER (34)	3: POSITION (35-38)	NOT (N) (39)	C/Z/D (40)	CHARACTER (41)	STACKER SELECT (42)	PACKED (43)	FIELD LOCATION FROM (44-47)	FIELD LOCATION TO (48-51)	DECIMAL POSITIONS (52)	FIELD NAME (VARIABLE NAME) (53-58)	CONTROL LEVEL (59-60)	MATCHING FIELDS OR CHAINING FIELDS (61-62)	FIELD RECORD RELATION (63-64)	FIELD INDICATORS PLUS (65-66)	MINUS (67-68)	ZERO OR BLANK (69-70)	NOT USED (71-74)
01	I	INPUT	AA			99																									▨
02	I																														▨
03	I																				1	50		DIRKEY							▨
04	I																				11	14		INXKEY							▨
05	I																				1	10		SHØRT							▨
																					1	17		LØNG							▨

Figure 4-1. Indexed and Direct Files
(Sheet 1 of 4)



CALCULATION SPECIFICATIONS

PAGE □□ PROGRAM IDENTIFICATION □□□□□□

LINE	FORM TYPE	CONTROL LEVEL	INDICATORS	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS	COMMENTS
01	C			DIRKEY	CHAIN	DIRECT						
02	C											

OUTPUT - FORMAT SPECIFICATIONS

PAGE □□ PROGRAM IDENTIFICATION □□□□□□

EDIT CODES					
COMMAS	ZERO BALANCES TO PRINT	NO SIGN	CR	-	X = REMOVE PLUS SIGN Y = DATE FIELD EDIT Z = ZERO SUPPRESS
YES	YES	1	A	J	
YES	NO	2	B	K	
NO	YES	3	C	L	
NO	NO	4	D	M	

LINE	FORM TYPE	FILENAME	TYPE	STACKER SELECT/FETCH OVERFLOW	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS	FIELD NAME (VARIABLE NAME)	EDIT CODES	BLANK AFTER	END POSITION	PACKED	CONSTANT OR EDIT WORD	NOT USED
01	O	DIRECT	D						99							
02	O									SHØRT			10			
03	O	INDEXED	D						99							
04	O									LØNG			17			

Figure 4-1. Indexed and Direct Files
(Sheet 2 of 4)

FILE UPDATE

FILE DESCRIPTION SPECIFICATIONS

LINE 3-5	FORM TYPE 6	FILENAME 7-14	FILE TYPE 15	FILE DESIGNATION 16	END OF FILE 17	SEQUENCE 18	FILE FORMAT 19	BLOCK LENGTH 20-23	RECORD LENGTH 24-27	PROCESSING MODE 28	RECORD ADDRESS FIELD LENGTH 29-30	RECORD ADDRESS TYPE 31	FILE ORGANIZATION TYPE 32	OVERFLOW INDICATOR 33-34	KEY FIELD STARTING LOCATION 35-38	EXTENSION CODE 39	DEVICE 40-46	NOT USED 47-52	LABELS 53	NOT USED 54-59	CORE INDEX 60-65	FILE ADDITION/UNORDERED 66	NOT USED 67	TAPE REWIND 70	FILE CONDITION 71-72	NOT USED 73-74
02	F	INPUT	I	P	E		F	80	80				2				READER									
03	F	DIRECT	Ø	C			F	430	10				2				DISK									
04	F	INDEXED	Ø				F	170	17		4		I		11		DISK				400	U				

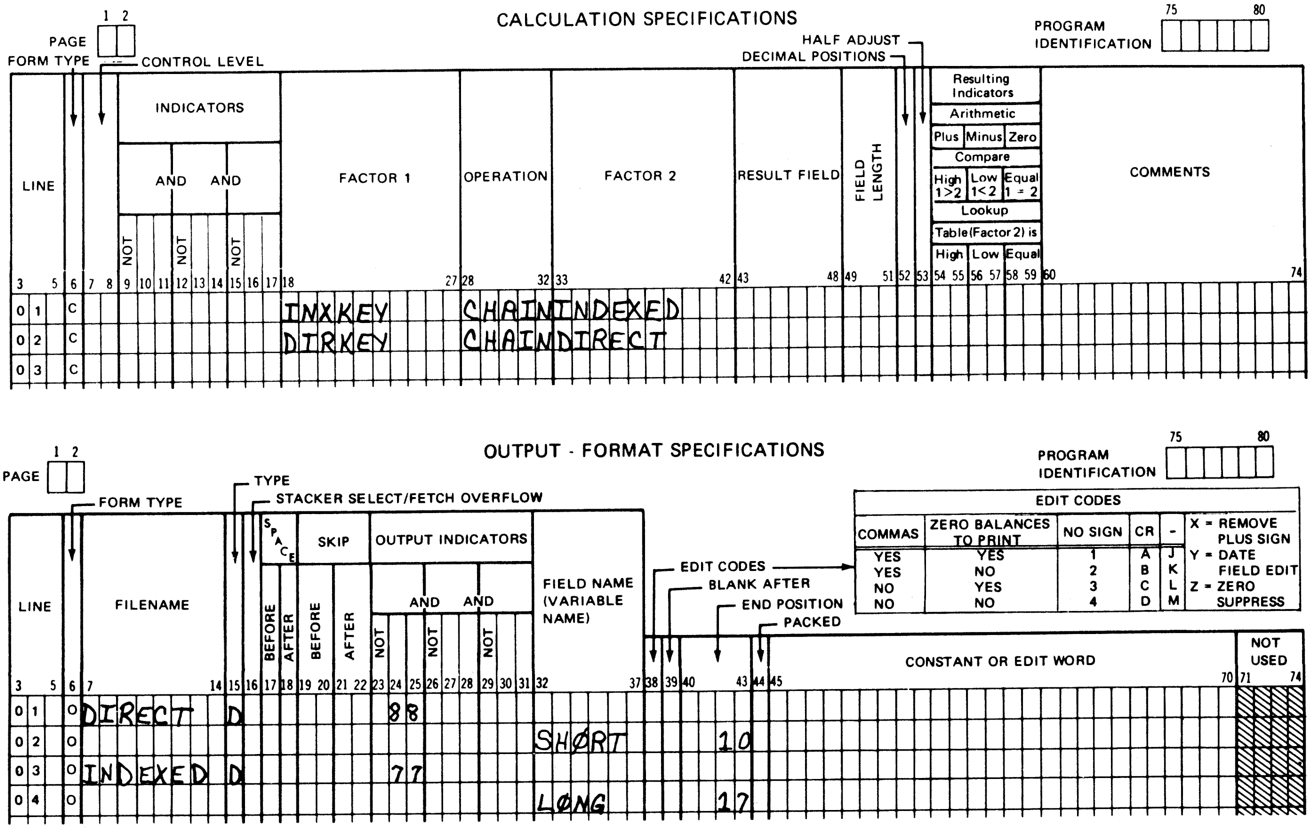
PAGE □□ DOLLAR CARD SPECIFICATIONS PROGRAM IDENTIFICATION □□□□□□

LINE 3-5	FORM TYPE 6	$ OPTION 7	NOT 8	KEY WORD 9-14	VALUE 15-24	COMMENTS 25-74
01		$		FAMILY	MULTIDISK	MULTIFILE ID
02		$		FILEID	DIRECT	FILE ID
03		$		AREAS	5	NUMBER ØF AREAS ALLØCATED
04		$		RPERA	111	NUMBER ØF RECØRDS PER AREA
05		$				
06		$		FAMILY	INDEXFILE	MULTIFILE ID
07		$		FILEID	INDEXED	FILE ID
08		$		AREAS	5	NUMBER ØF AREAS ALLØCATED
09		$		RPERA	50	NUMBER ØF RECØRDS PER AREA

PAGE □□ INPUT SPECIFICATIONS PROGRAM IDENTIFICATION □□□□□□

LINE 3-5	FORM TYPE 6	FILENAME 7-14	SEQUENCE 15-16	NUMBER 17	OPTION 18	RECORD IDENTIFYING INDICATOR 19-20	RECORD IDENTIFICATION CODES 21-41	STACKER SELECT 42	PACKED 43	FIELD LOCATION FROM 44-47	FIELD LOCATION TO 48-51	DECIMAL POSITIONS 52	FIELD NAME (VARIABLE NAME) 53-58	CONTROL LEVEL 59-60	MATCHING FIELDS OR CHAINING FIELDS 61-62	FIELD RECORD RELATION 63-64	FIELD INDICATORS PLUS 65-66	MINUS 67-68	ZERO OR BLANK 69-70	NOT USED 71-74
01	I	INPUT	AA			99														
02	I									1	5	0	DIRKEY							
03	I									11	14		INXKEY							
04	I									1	10		SHØRT							
05	I									1	17		LØNG							
06	I	DIRECT	BB			88														
07	I									1	5	0	DIRKEY							
08	I	INDEXED	CC			77														
09	I									11	14		INXKEY							

Figure 4-1. Indexed and Direct Files
(Sheet 3 of 4)



CALCULATION SPECIFICATIONS

PAGE □□ (1 2) FORM TYPE — CONTROL LEVEL — HALF ADJUST — DECIMAL POSITIONS — PROGRAM IDENTIFICATION □□□□□□ (75 80)

LINE (3-5)	FORM TYPE (6)	CONTROL LEVEL (7-8)	INDICATORS (9-17)	FACTOR 1 (18-27)	OPERATION (28-32)	FACTOR 2 (33-42)	RESULT FIELD (43-48)	FIELD LENGTH (49-51)	DECIMAL POSITIONS (52)	HALF ADJUST (53)	RESULTING INDICATORS (54-59)	COMMENTS (60-74)
01	C			INXKEY	CHAIN	INDEXED						
02	C			DIRKEY	CHAIN	DIRECT						
03	C											

Resulting Indicators: Arithmetic — Plus, Minus, Zero; Compare — High 1>2, Low 1<2, Equal 1 = 2; Lookup — Table (Factor 2) is High, Low, Equal

OUTPUT - FORMAT SPECIFICATIONS

PAGE □□ (1 2) FORM TYPE — TYPE — STACKER SELECT/FETCH OVERFLOW — PROGRAM IDENTIFICATION □□□□□□ (75 80)

EDIT CODES					
COMMAS	ZERO BALANCES TO PRINT	NO SIGN	CR	-	X = REMOVE PLUS SIGN
YES	YES	1	A	J	Y = DATE FIELD EDIT
YES	NO	2	B	K	Z = ZERO SUPPRESS
NO	YES	3	C	L	
NO	NO	4	D	M	

LINE (3-5)	FORM TYPE (6)	FILENAME (7-14)	TYPE (15)	STACKER SELECT (16)	SPACE BEFORE (17)	SPACE AFTER (18)	SKIP BEFORE (19-20)	SKIP AFTER (21-22)	OUTPUT INDICATORS (23-31)	FIELD NAME (VARIABLE NAME) (32-37)	EDIT CODES (38)	BLANK AFTER (39)	END POSITION (40-43)	PACKED (44)	CONSTANT OR EDIT WORD (45-70)	NOT USED (71-74)
01	O	DIRECT	D						88							
02	O									SHØRT			10			
03	O	INDEXED	D						77							
04	O									LØNG			17			

Figure 4-1. Indexed and Direct Files
(Sheet 4 of 4)

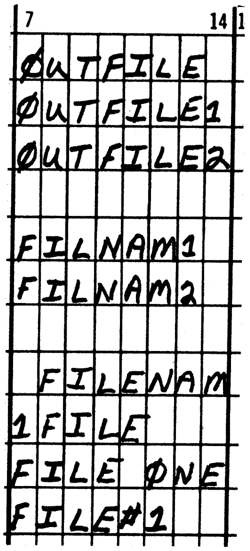
Figure 4-1 can be used in conjunction with the following field definitions for the File Description Specifications.

PAGE (COLUMNS 1-2).
Refer to section 2 for a complete description.

LINE (COLUMNS 3-5).
Refer to section 2 for a complete description.

FORM TYPE (COLUMN 6).
This field must contain a F.

FILENAME (COLUMNS 7-14).
This field is used to assign a unique name to every file used by the program. FILENAMEs must be assigned in accordance with the rules outlined in sectfon 2 of this manual (see figure 4-2).



Second and third FILENAMEs are not unique in the first seven characters, and thus are invalid.

Valid FILENAMEs. Both are unique in the first seven characters.

Invalid FILENAMEs. FILENAMEs must be left-justified, must start with an alphabetic character, and must contain no embedded spaces or special characters.

Figure 4-2. Assigning FILENAMEs

FILE TYPE (COLUMN 15).
This field is used to identify the manner in which the program uses the file. Valid entries are:

Entry	Definition
I	Input file.
O	Output file.
U	Update file.
C	Combined file.
D	Display file.

INPUT FILES. Input files contain records that the program uses as a source of data. If a file is described as input, it indicates that records will be read from that file. Input files must be further described on the Input Specifications form with the exception of table files, which must be described on the Extension Specifications form.

OUTPUT FILES. Output files contain records that are written, printed, or punched as output from the program. All output files must be further described on the Output-Format Specifications form, except for output table files.

UPDATE FILES. Update files are disk files from which a program reads a record, changes fields in the record, and writes it back into the same location from which it was read. All update files must be further described on both the Input Specification and Output-Format Specifications forms.

COMBINED FILES. Combined files are card files which can be used for both input and output. These files consist of cards that are read by the program and are subsequently punched and/or printed as output. The punching may be into the same cards that have been read or into blank cards that have been inserted into the combined file. Combined files must be further described on both the Input Specifications and Output-Format Specifications forms.

DISPLAY FILES. Display files are used to print a field or record directly on the console printer (SPO). The DSPLY operation code must be used in the Calculation Specifications in order to perform the print operation. Display files are described only on the File Description Specifications form.

FILE DESIGNATION (COLUMN 16).
This field is used to further describe the use of input, update, and combined files. It must be left blank for all output files (including display files), except for chained output files. Acceptable entries for this field are:

Entry	Definition
Blank	Output file.
P	Primary sequential file.
S	Secondary sequential file.
C	Chained (random) file.
T	Input table file.
D	Demand file.

PRIMARY FILES. A primary file is the principal file from which the program reads input records, and may be designated as input, update, or combined. Every RPG Program must have one primary file; all primary files described after the first one named are considered to be secondary files.

If there is no P entry in the File Description Specifications, a warning message is emitted by the compiler and the first S (secondary) file defined is assumed to be the primary file. When no primary or secondary files are present, a syntax error is emitted.

NOTE

A P entry must be the first file declared, or a warning message is emitted.

SECONDARY FILES. All files other than the primary file involved in record selection (note that this excludes table, chained, and demand files) during multifile processing are considered secondary files. These files are processed in the same order in which they are written in the File Description Specifications.

CHAINED FILES. A chained file may be an input, output, or update file assigned to disk that uses the CHAIN operation code to read or write records randomly.

NOTE

Refer to appendix G for a description of automatic chaining in IBM's RPG.

TABLE FILES. A table file contains the vector entries which can be read into the program during either compilation or execution. However, only execution-time vectors are described on the File Description Specification form. Both compile- and execution-time vectors must be described on the Extension Specification form.

Table files are only a means for supplying entries for tables used by the program and are not involved in record selection during processing at execution time. All records in table files read during program execution are read before any other data records.

Both compile- and execution-time vectors may be changed at execution time; however, vector entries read during compilation can be permanently altered only by recompiling the program.

DEMAND FILES. A demand file may be an update, input, or combined file from which records are read through use of the READ operation code in the Calculation Specifications.

END OF FILE (COLUMN 17).
This column specifies which files are to be checked for End-of-File during multifile processing in order to turn the last record indicator (LR) ON. It applies only to input, update, and combined files declared as primary or secondary, and is used to indicate whether or not the program may end before all of the records from the file are processed. Valid entries for this field are:

Entry	Definition
Blank	If this entry is blank for all files, all records must be read from all files before the program may end.
	The program may end whether or not all records from the file have been read.
E	All records in the file must be read and processed before the program may end.

If all records from all input files must be read and processed before the LR indicator can be turned ON, then this column must be blank (or contain E) for all files.

SEQUENCE (COLUMN 18).
This field is used by update, combined, and all input files except table files, to indicate whether or not the program is to check the sequence of the input records. The Input Specifications form (columns 61-62) must be used to specify the match fields within the input record records. This entry must be left blank for output, display, and table files. Acceptable entries for this field are:

Entry	Definition
Blank or A	Records with matching fields are to be sequence-checked in ascending order.
D	Records with matching fields are to be sequence-checked in descending order.

Sequence checking is required when matching fields have been specified for the records in a file.

If a record from a matching input file is found to be out of sequence, the program halts. At this time the system operator may:

a. Ignore the record out of sequence and read the next record from the same file.

b. Ignore the record out of sequence, turn on the LR indicator, and perform all final detail and total calculation procedures.

c. Discontinue (DS) the program.

All sequence checking is performed according to the EBCDIC collating sequence.

NOTE

If any matching file specifies descending (D) sequence, all files must specify descending sequence.

FILE FORMAT (COLUMN 19).

This field specifies whether the file contains fixed- or variable-length records. Variable-length records may not be specified for files other than those assigned to tape. The acceptable entries are:

Entry	Definition
Blank or F	Fixed record length.
V	Variable record length (tape files only).

BLOCK AND RECORD LENGTH (COLUMNS 20-27).

These fields are used to specify the block and record sizes for the file. The minimum and maximum record and block lengths allowed depend upon the device to which the file is assigned. This information is specified in table 4-1.

Table 4-1

Record and Block Sizes

	Minimum		Maximum	
Device	Record Length	Block Length	Record Length	Block Length
Printer	132*	N/A	132*	N/A
Disk	1	1	4095	9999
Tape	7	7	4095	9999
Card	80,96*	N/A	80,96*	N/A
*Device dependent				

Entries in the BLOCK LENGTH (columns 20-23) and RECORD LENGTH (columns 24-27) fields must be right-justified, and leading zeros may be omitted. If these entries are left blank, default sizes are assumed; 132 characters for printer files and 80 characters for all other files.

FIXED-LENGTH RECORDS. The length of one block and one record must be entered in the BLOCK LENGTH and RECORD LENGTH fields respectively. The BLOCK LENGTH must be an integral multiple of the RECORD LENGTH. For unblocked files, the BLOCK LENGTH will be equal to the RECORD LENGTH.

VARIABLE-LENGTH RECORDS (TAPE FILES ONLY). To be defined later.

PROCESSING MODE (COLUMN 28).
This field is ignored by the compiler and should be left blank.

RECORD ADDRESS FIELD LENGTH (COLUMNS 29-30).
This field applies only to chained files and specifies the length of the key field within the records of the file. All key fields in the file must be the same length and must be less than 100 characters in length. The numeric entry must be right-justified within the field. Leading zeros may be omitted.

RECORD ADDRESS TYPE (COLUMN 31).
This field specifies the manner by which the records in the file are identified. The acceptable entries are listed below.

Entry	Definition
Blank, K, A	Alphanumeric.
P	Packed.
N	All numeric data in alphanumeric format.

FILE ORGANIZATION TYPE. Records may be organized in a file in one of three ways: sequential, direct, and indexed.

Sequential Files.
Records in sequential files are organized in the same order in which they were placed in the file. For instance, the fifth record placed in the file occupies the fifth record position.

All files other than those assigned to disk are sequential. Disk files may be sequential, direct, or indexed. All demand files (including those assigned to disk) must be sequential.

Direct Files.
Direct files are files assigned to disk in which records are assigned specific record positions, regardless of the order in which they are put into the file. Relative record numbers (actual keys) specify the relative position of each record within the file (see figure 4-1).

Indexed Files.
In an indexed file, a portion of memory is set aside to contain the locations of data records within the file. This portion, called the index, contains the information from the key field of some records, and is used to identify the records of an indexed file (see figure 4-1). Data keys are required when using an indexed file (refer to

RECORD ADDRESS FIELD LENGTH - columns 29-30 and KEY FIELD STARTING LOCATION - columns 35-38). An indexed file may be loaded in ascending key sequence or unordered (refer to FILE ADDITION/UNORDERED - column 66). If the file is loaded in an unordered sequence, the file is sorted after the load to place the records in ascending sequence.

NOTE
Only chained input and chained update files may be indexed.

ADDITIONAL INPUT/OUTPUT AREAS. The use of additional I/O areas for a file increases the efficiency of the program when it is executed; however, it also increases the size of the program. A good balance between increased efficiency and increased size must be reached in order to achieve the greatest throughput from the system when programs are multiprocessed.

Additional I/O areas cannot be used with table or demand files, nor with indexed disk files.

OVERFLOW INDICATOR (COLUMNS 33-34).
This field applies only to files assigned to the printer, and is used to specify the overflow indicator used to condition records being printed in the file. Each printer file must have a unique overflow indicator assigned to it, if overflow printing (i.e., the printing of special lines when the overflow line is reached) is desired for that file. Acceptable entries for this field are:

Entry	Definition
OA-OG, OV	Specified indicator is used to condition records in the file.
Blank	Default to OF or OV according to entry in columns 40-46 (DEVICE).

KEY FIELD STARTING LOCATION (COLUMNS 35-38).
This field applies only to chained-indexed files and specifies the character position within each record where the key field begins. All key fields in the file must occupy the same position in each record. The entry in the RECORD ADDRESS FIELD LENGTH (columns 29-30) specifies the length of the key field. The KEY FIELD STARTING LOCATION entry must be numeric; between 1 and 4095 inclusive, and must be right-justified. Leading zeros may be omitted.

EXTENSION CODE (COLUMN 39).
This field applies to printer output files and table or array files that are to be read during program execution. It indicates whether the file is further described on the Extension Specifications (table or array files) or Line Counter Specifications (printer files) form. Valid entries for this field are:

Entry	Definition
E	Extension Specifications further describe the file.
L	Line Counter Specifications further describe the file.
Blank	No Line Counter or Extension Specifications are used for this file.

DEVICE (COLUMNS 40-46).
This field is used to identify the input/output device to which the file will be assigned. All entries must be left-justified. Valid entries for this field are:

Entry	Definition
READER	80-column card reader.
MFCU1	96-column MFCU - primary hopper.
MFCU2	96-column MFCU - secondary hopper.
PUNCH	80-column card punch.
PRINTER or PRINTR2	Line printer.
TAPE	Magnetic tape.
DISK	Disk file.
SPO or CONSOLE	Console printer.

> NOTE
>
> If the OVERFLOW INDICATOR field is left blank, a PRINTER entry in the DEVICE field will result in a default setting of OF for the overflow indicator; a PRINTR2 entry will result in a default setting of OV.

For a description of the use of this field for IBM's RPG, refer to appendix G of this manual.

NOT USED (COLUMNS 47-52).
This field is ignored by the compiler and should be left blank.

LABELS (COLUMN 53).
This field is considered to be purely documentational by the compiler. All options are ignored, and the files are all assumed to contain Burroughs standard labels.

NOT USED (COLUMNS 54-59).
This field is ignored by the compiler and should be left blank.

CORE INDEX (COLUMNS 60-65).
This field is used only by chained-indexed disk files to specify the number of bytes of memory to be set aside for an index. The entry must be right-justified, and leading zeros may be omitted. The number entered should be large enough to contain at least 10 keys. If the field is left blank or is not large enough to contain 10 keys, a default size of 10 times the RECORD ADDRESS FIELD LENGTH will be assigned, and a warning will be issued (see figure 4-1).

The core index is a table in memory which contains entries from the key fields of the data file on disk. The use of the core index significantly reduces the time necessary to process an indexed file because it allows a more direct access to the specific record required. The program must only search a portion of the file instead of all entries in the file preceding the required record.

FILE ADDITION/UNORDERED (COLUMN 66).
This field applies only to sequential disk and indexed disk files and indicates:

a. New records are to be added to an existing file; or
b. Unordered records are to be loaded into an output file.

Valid entries for this field are listed below.

Entry	Definition
A	Records are to be added to the file.
U	Records are to be loaded for an indexed file in unordered sequence.
Blank	Output file (must be ordered).

Records added to a sequential or indexed file are added at the end of the file. After all records have been added to an indexed file, the file is sorted so that the records are in ascending order.

NOTE
An A entry in this field is required if there is an ADD entry in columns 16-18 of the Output-Format Specifications record description for the file.

NOT USED (COLUMN 67).
This field is ignored by the compiler and should be left blank.

NOT USED (COLUMNS 68-69).
This field is ignored by the compiler and should be left blank.

TAPE REWIND (COLUMN 70).
This field specifies the action to be taken during closing of the file, and includes the provision for rewind and/or lock for tape reels, where desired. Valid entries are:

Entry	Definition
P	Close with purge.
U	Close with unload (lock).
N	Close with no rewind.
Blank	Close with release.
R	Removes file-name from Disk Directory at EOJ.

To show the effects of the various options, each type of file will be discussed separately in the paragraphs that follow.

CARD INPUT. All options are ignored. The input areas are released and the unit is returned to the MCP.

CARD OUTPUT. All options are ignored. The output areas are released, the trailer label is punched, and the unit is returned to the MCP.

TAPE INPUT. The effects of the various options for tape input are described in the text that follows.

U (Close With Unload).
Releases the input areas, rewinds the tape, and the MCP marks the unit "not ready".

N (Close With No Rewind).
Same as "Blank" (close with release), except the tape is not rewound.

Blank (Close With Release).
Releases the input areas, rewinds and returns the unit to the MCP.

TAPE OUTPUT. The effects of the various options for tape output are described in the text that follows.

P (Close With Purge).
Releases the output areas, writes the trailer label, rewinds the tape, and overwrites the label, thus making the tape a scratch tape.

U (Close With Unload).
Releases the output areas, writes the trailer label, rewinds the tape, and the MCP marks the unit "not ready".

N (Close With No Rewind).
Same as "Blank" (close with release), except the tape is not rewound.

Blank (Close With Release).
Releases the output areas, writes the trailer label, rewinds the tape, and returns the unit to the MCP.

PRINTER OUTPUT. All options are ignored. A page is ejected, a trailer label is written, and the printer is returned to the MCP.

DISK FILES. The action taken on files assigned to disk will be discussed in terms of "old files" and "new files". An old file is one that already exists on disk and appears in the MCP Disk Directory. A new file is one created by the program, and does not appear in the Directory. A new file may only be referenced by the program which creates it.

P (Close With Purge).
An old file is removed from the disk and deleted from the Directory, or a new file is removed from disk.

U (Close With Unload).
For an old file, the file remains in the Directory and is made available. A new file is entered in the Directory (thereby making it an old file) and made available.

N (Close With No Rewind).
Use of this option is not permitted with disk files.

Blank (Close With Release).
Same as for U (close with unload).

FILE CONDITION (COLUMNS 71-72).
This field applies to input (excluding table input files), update, output, and combined files, and indicates whether or not the file is conditioned by an external indicator. This is used to indicate (at execution time) whether or not the file is to be used by the program. If a file is conditioned by an external indicator, the file is used only when that indicator is ON. When the indicator is OFF, the file is treated as though End-of-File had been reached, and no records may be read or written into the file. Valid entries for this field are:

Entry	Definition
U1-U8	The specified external indicator is used to condition the file.
Blank	The file is not conditioned by an external indicator.

The external indicators are set at the start of program execution by using the MCP control statements to be defined.

NOT USED (COLUMNS 73-74).
This field is ignored by the compiler and should be left blank.

PROGRAM IDENTIFICATION (COLUMNS 75-80).
Refer to section 2 for a complete description.

SECTION 5

EXTENSION SPECIFICATIONS

GENERAL.

Extension Specifications are used to complete the description of tables and arrays which are partially defined on the File Description Specifications form. Refer to appendix I for a complete dicussion of vectors, along with specific entries required by each type.

The coding form for Extension Specifications (see figure 5-1) is illustrated below.

FIELD DEFINITIONS.

The fields for the Extension Specifications Form are defined in the text that follows.

PAGE (COLUMNS 1-2).
Refer to section 2 for a complete description.

LINE (COLUMNS 3-5).
Refer to section 2 for a complete description.

FORM TYPE (COLUMN 6).
An E must appear in this field.

NOT USED (COLUMNS 7-8).
This field is ignored by the compiler and should be left blank.

CHAINING FIELD NUMBER (COLUMNS 9-10).
This field is ignored by the compiler and should be left blank.

NOTE

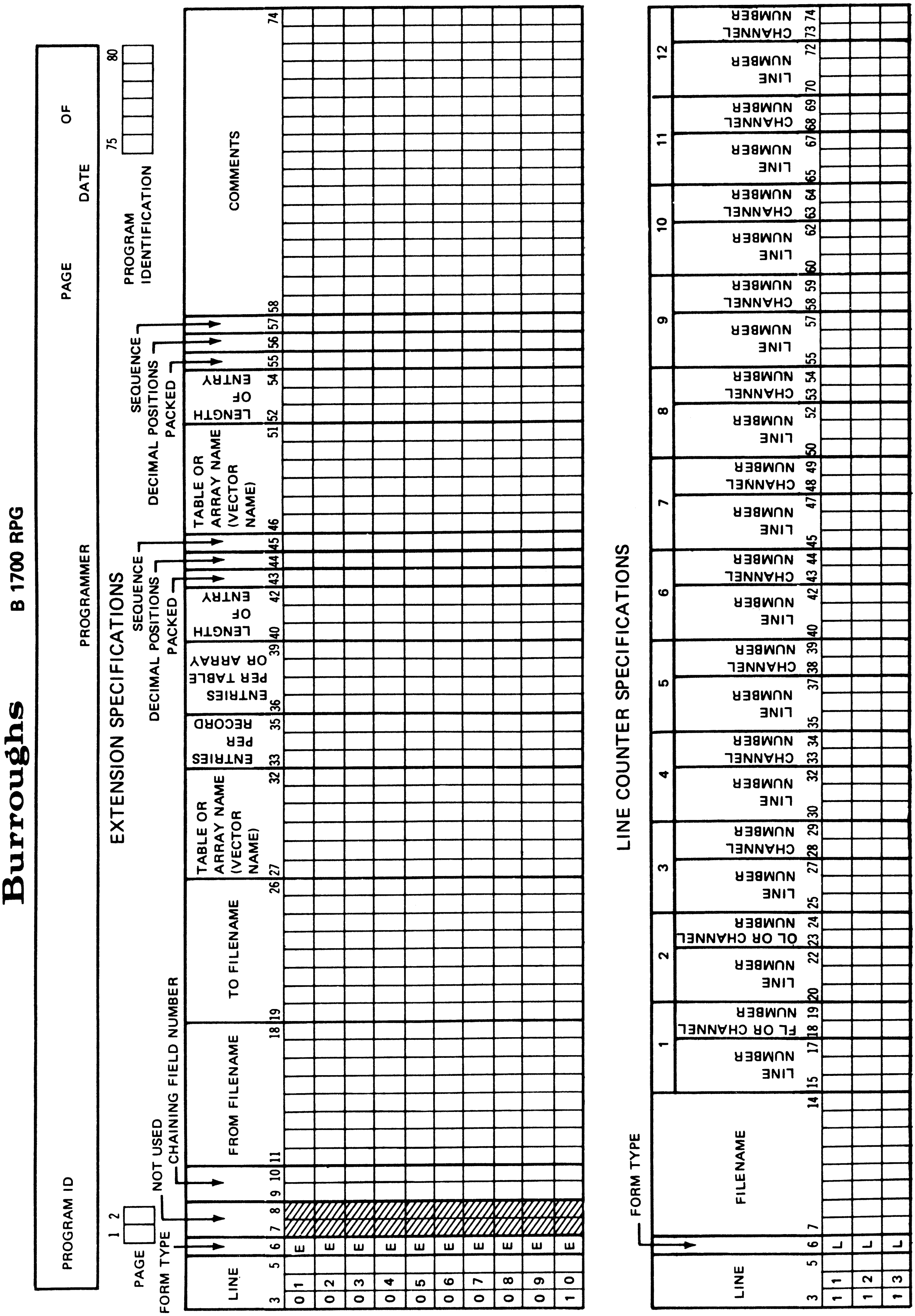
For a description of the use of this field by IBM's RPG, refer to appendix G.

FROM FILENAME (COLUMNS 11-18).
This field is used to name a vector file and must contain the FILENAME of every execution-time vector to be used by the program; if the vector is loaded at compile time or via Input or Calculation Specifications during execution, this field must be left blank.

Filenames must always be entered in this field left-justified. When a vector is loaded at compile time, it becomes a permanent part of the program so that a vector file deck is not needed when the program is executed. Only those vectors that do not change often should be compiled with the program. When vectors are being compiled with the program, the vector file deck must follow the source program deck.

NOTE

For a description of the use of this field by IBM's RPG, refer to appendix G.



Burroughs B 1700 RPG

PROGRAM ID | PROGRAMMER | PAGE | DATE | OF

PAGE 1 2 | PROGRAM IDENTIFICATION 75 80

EXTENSION SPECIFICATIONS

LINE 3 5	FORM TYPE 6	NOT USED 7 8	CHAINING FIELD NUMBER 9 10	FROM FILENAME 11 18	TO FILENAME 19 26	TABLE OR ARRAY NAME (VECTOR NAME) 27 32	ENTRIES PER RECORD 33 35	ENTRIES PER TABLE OR ARRAY 36 39	LENGTH OF ENTRY 40 42	PACKED 43	DECIMAL POSITIONS 44	SEQUENCE 45	TABLE OR ARRAY NAME (VECTOR NAME) 46 51	LENGTH OF ENTRY 52 54	PACKED 55	DECIMAL POSITIONS 56	SEQUENCE 57	COMMENTS 58 74
01	E	▨																
02	E	▨																
03	E	▨																
04	E	▨																
05	E	▨																
06	E	▨																
07	E	▨																
08	E	▨																
09	E	▨																
10	E	▨																

LINE COUNTER SPECIFICATIONS

LINE 3 5	FORM TYPE 6	FILENAME 7 14	1 LINE NUMBER 15 17	1 FL OR CHANNEL NUMBER 18 19	2 LINE NUMBER 20 22	2 OL OR CHANNEL NUMBER 23 24	3 LINE NUMBER 25 27	3 CHANNEL NUMBER 28 29	4 LINE NUMBER 30 32	4 CHANNEL NUMBER 33 34	5 LINE NUMBER 35 37	5 CHANNEL NUMBER 38 39	6 LINE NUMBER 40 42	6 CHANNEL NUMBER 43 44	7 LINE NUMBER 45 47	7 CHANNEL NUMBER 48 49	8 LINE NUMBER 50 52	8 CHANNEL NUMBER 53 54	9 LINE NUMBER 55 57	9 CHANNEL NUMBER 58 59	10 LINE NUMBER 60 62	10 CHANNEL NUMBER 63 64	11 LINE NUMBER 65 67	11 CHANNEL NUMBER 68 69	12 LINE NUMBER 70 72	12 CHANNEL NUMBER 73 74
11	L																									
12	L																									
13	L																									

Figure 5-1. Extension Specifications Form

TO FILENAME (COLUMNS 19-26).
If it is desirable to write or punch out a vector file, the file-name of the proper output file must be entered in this field. The designated output file must have been previously described on the File Description Specifications form. A vector can be written to only one output device, and is automatically written or punched at End-of-Job after all other records have been processed.

File-names must always be entered in this field left-justified.

NOTE
For a description of the use of this field by IBM's RPG, refer to appendix G.

VECTOR NAME (COLUMNS 27-32).
This field is used to name tables and arrays to be used by the program. Each vector name must be unique, and must follow the rules for the formation of vector names as described in section 2. Table names must begin with the letters TAB; any name appearing in this field which does not begin with TAB is considered an array name.

Vector files are processed in the same order in which they appear on the Extension Specifications form. Thus, if more than one vector file is specified for the program, the files must be loaded in the same order in which they appear on the form.

If two related vectors are in alternating form within one vector file, the first vector must be named in columns 27-32, and the second vector must be named in columns 46-51.

ENTRIES PER RECORD (COLUMNS 33-35).
This field is used to specify the exact number of entries contained in each vector input record. Every record except the last one, must contain the same number of entries as specified in this field; the last record may contain fewer entries than specified. Entries in this field must be right-justified; leading zeros may be omitted. Corresponding items from related (alternating) vectors must be on the same record, and in alternating format. Each pair of items is considered one entry.

When loading a vector, the following rules must be observed:

a. To load a vector at execution time, entries must be made in the FROM FILENAME and ENTRIES PER RECORD fields.

b. To load a vector at compile time, the FROM FILENAME field must be blank, and an entry must be made in the ENTRIES PER RECORD field.

c. To load a vector via the Input or Calculation Specifications, both the FROM FILENAME and ENTRIES PER RECORD fields must be blank.

ENTRIES PER TABLE OR ARRAY (COLUMNS 36-39).
This field is used to specify the maximum number of items that can be contained in the vector named in the first VECTOR NAME field (columns 27-32). For vectors to be loaded in alternating format, this number also applies to the one named in the second VECTOR NAME field (columns 46-51). Entries in this field must be right-justified; leading zeros may be omitted.

LENGTH OF ENTRY (COLUMNS 40-42).
This field is used to specify the length (in bytes) of each element in the vector named in the first VECTOR NAME field (columns 27-32). For numeric vectors in packed decimal format, enter the number of digits as if the field were unpacked. Entries in this field must be right-justified; leading zeros may be omitted.

Numeric items in the vector input records must have leading zeros added if their length is less than that specified; alphanumeric entries must have either leading or trailing blanks.

The maximum length of a numeric vector element is 31 characters; the maximum for an alphanumeric vector element is 255 characters. However, an element must be completely contained on one record; therefore, input record sizes will also limit the maximum element sizes.

PACKED (COLUMN 43).
This field specifies the format of the vector elements. Acceptable entries are:

Entry	Definition
Blank	Vector elements are in either unpacked decimal or alphanumeric format.
P	Vector elements are in packed decimal format.

DECIMAL POSITIONS (COLUMN 44).
This field is used to specify the number of decimal positions contained in each element of the vector named in the first VECTOR NAME field (columns 27-32).

This field must not be blank for a numeric vector; if the elements have no decimal positions, a zero must be entered.

The acceptable entries for this field are:

Entry	Definition
Blank	Alphanumeric vector.

Entry	Definition
0-9	Number of positions to the right of implied decimal point for numeric vector elements.

SEQUENCE (COLUMN 45).
This field is used to specify the sequence in which elements will be loaded for the vector named in the first VECTOR NAME field (columns 27-32). A sequence error during a compile time vector load will not be detected; however, it can cause an error or "not found" condition during program execution. Valid entries for this field are:

Entry	Definition
Blank	Unordered elements.
A	Elements arranged in ascending order.
D	Elements arranged in descending order.

VECTOR NAME, LENGTH OF ENTRY, PACKED,
DECIMAL POSITIONS, SEQUENCE (COLUMNS 46-57).
These fields are used only when describing a second vector which is loaded in alternating format with the vector named in the first VECTOR NAME field (columns 27-32). All of these fields require the same type of entries as the corresponding fields in columns 27-45, but only apply to the second vector. For a single vector description, these fields must be left blank.

COMMENTS (COLUMNS 58-74).
This field is available for inclusion of comments and documentary remarks, and may contain any valid EBCDIC characters.

PROGRAM IDENTIFICATION (COLUMNS 75-80).
Refer to section 2 for a complete description.

SECTION 6

LINE COUNTER SPECIFICATIONS

GENERAL.

Line Counter Specifications are optional and are only used for printer files to indicate form length, the position of the last line to be printed on each page, and the format of the carriage control tape. If Line Counter Specifications are absent for a printer file, a form length of 66 lines (11 inches long @ 6 lines per inch) and an overflow line of 60 (one inch from bottom of form @ 6 lines per inch) are assumed.

The coding form for Line Counter Specifications is illustrated below (see figure 6-1).

FIELD DEFINITIONS.

The fields for the Line Counter Specifications are defined in the text that follows.

PAGE (COLUMNS 1-2).
Refer to section 2 for a complete description.

LINE (COLUMNS 3-5).
Refer to section 2 for a complete description.

FORM TYPE (COLUMN 6).
This field must contain the letter L.

FILENAME (COLUMNS 7-14).
This field is used to specify the name of the printer file to which these Line Counter Specifications apply. The filename must also be described on the File Description Specifications and must be assigned to a printer. The entry in this field must be left-justified.

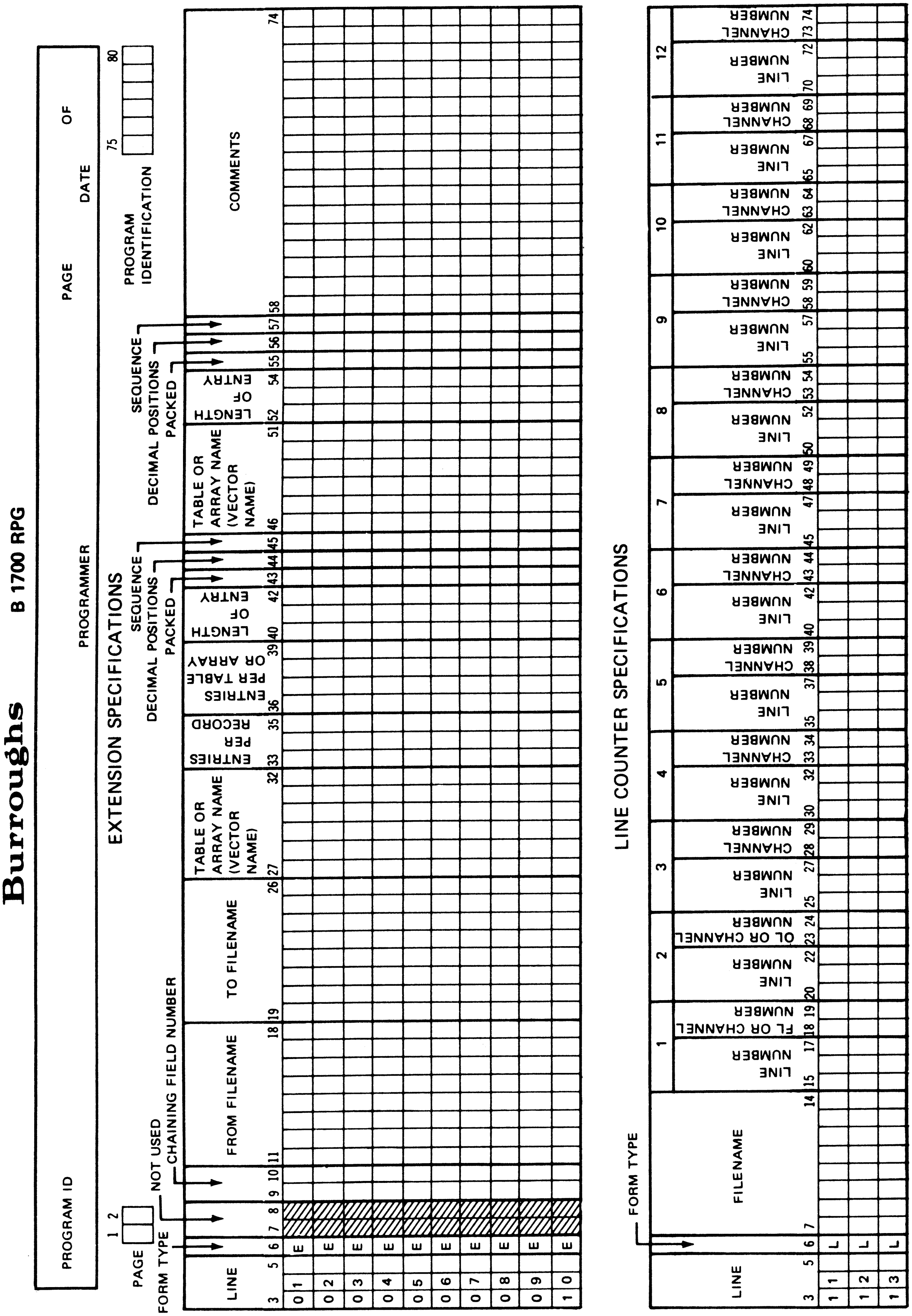
NOTE
This field may be blank, in which case the specifications apply to all printer files declared in the program.

LINE NUMBER, FL OR CHANNEL NUMBER (COLUMNS 15-19).
These fields have two possible meanings, depending upon the entry in columns 18-19.

If columns 18-19 contain the entry FL, the entry in columns 15-17 specifies the length (in print lines) of each page. The LINE NUMBER field entry must be between one and 112, inclusive, and be right-justified in the field (leading zeros are optional).

If the LINE NUMBER field is left blank, a form length of 66 lines is assumed (11 inches long @ 6 lines per inch).

If columns 18-19 contain a numeric entry between one and 12, inclusive, the entry in columns 15-17 specifies the number of lines from the top of the form to be associated with the CHANNEL NUMBER entry designated



Burroughs B 1700 RPG

PROGRAM ID	PROGRAMMER	PAGE	DATE	OF

PAGE 1 2 □□

PROGRAM IDENTIFICATION 75 80 □□□□□□

EXTENSION SPECIFICATIONS

LINE 3 5	FORM TYPE 6	NOT USED 7 8	CHAINING FIELD NUMBER 9 10	FROM FILENAME 11 18	TO FILENAME 19 26	TABLE OR ARRAY NAME (VECTOR NAME) 27 32	ENTRIES PER RECORD 33 35	ENTRIES PER TABLE OR ARRAY 36 39	LENGTH OF ENTRY 40 42	PACKED 43	DECIMAL POSITIONS 44	SEQUENCE 45	TABLE OR ARRAY NAME (VECTOR NAME) 46 51	LENGTH OF ENTRY 52 54	PACKED 55	DECIMAL POSITIONS 56	SEQUENCE 57	COMMENTS 58 74
0 1	E	▨																
0 2	E	▨																
0 3	E	▨																
0 4	E	▨																
0 5	E	▨																
0 6	E	▨																
0 7	E	▨																
0 8	E	▨																
0 9	E	▨																
1 0	E	▨																

LINE COUNTER SPECIFICATIONS

			1		2		3		4		5		6		7		8		9		10		11		12	
LINE 3 5	FORM TYPE 6	FILENAME 7 14	LINE NUMBER 15 17	FL OR CHANNEL NUMBER 18 19	LINE NUMBER 20 22	OL OR CHANNEL NUMBER 23 24	LINE NUMBER 25 27	CHANNEL NUMBER 28 29	LINE NUMBER 30 32	CHANNEL NUMBER 33 34	LINE NUMBER 35 37	CHANNEL NUMBER 38 39	LINE NUMBER 40 42	CHANNEL NUMBER 43 44	LINE NUMBER 45 47	CHANNEL NUMBER 48 49	LINE NUMBER 50 52	CHANNEL NUMBER 53 54	LINE NUMBER 55 57	CHANNEL NUMBER 58 59	LINE NUMBER 60 62	CHANNEL NUMBER 63 64	LINE NUMBER 65 67	CHANNEL NUMBER 68 69	LINE NUMBER 70 72	CHANNEL NUMBER 73 74
1 1	L																									
1 2	L																									
1 3	L																									

Figure 6-1. Line Counter Specifications Form

in columns 18-19. The LINE NUMBER entry must be between one and 112, inclusive. Both entries must be right-justified in their respective fields, and leading zeros are optional.

LINE NUMBER, OL OR CHANNEL NUMBER (COLUMNS 20-24).
These fields have two possible meanings, depending upon the entry in columns 23-24.

If columns 23-24 contain the entry OL, the entry in columns 20-22 specifies the line number that will be considered the last line on the page. When the line specified as the overflow line is printed, the overflow indicator specified for the file is turned ON to indicate that the end of the page is near. When the overflow indicator is ON, the following actions will take place before the forms are advanced to the next page:

a. Detail lines still to be printed as part of the current program cycle will be completed.

b. Total lines will be printed.

c. Total lines conditioned by the overflow indicator for this file will be printed.

Since all these actions will take place <u>after</u> the overflow line is reached, the programmer should be certain that enough space is left between the overflow line and the bottom of the page to allow all the lines to be printed.

If the LINE NUMBER field is left blank, an overflow line of 60 is assumed (one inch from bottom of form @ 6 lines per inch).

If columns 23-24 contain a numeric entry between one and 12, inclusive, the entry in columns 20-22 specifies the number of lines from the top of the form to be associated with the CHANNEL NUMBER entry (columns 23-24). The LINE NUMBER field entry must be between one and 112, inclusive. Both entries must be right-justified in their respective fields, and leading zeros are optional.

LINE NUMBER AND CHANNEL NUMBER (COLUMNS 25-74).
The rest of the form is divided into 10 5-character fields, each consisting of a 3-character LINE NUMBER field and a 2-character CHANNEL NUMBER field. All fields are optional and must be left blank if they are not to be used.

The CHANNEL NUMBER fields may contain a numeric entry between one and 12, inclusive. This CHANNEL NUMBER entry is associated to the corresponding LINE NUMBER field entry (between one and 112, inclusive) and is used to relate a Channel Number to a particular position on each page of the output forms.

All entries must be right-justified in their respective fields. Leading zeros are optional.

PROGRAM IDENTIFICATION (COLUMNS 75-80).
Refer to section 2 for a complete description.

SECTION 7

INPUT SPECIFICATIONS

GENERAL.

Input Specifications describe the records within each file and fields within each record to be used as input data for the program. The two types of input specifications are:

a. Record type descriptions (columns 7-42) which define the various input records and their relationship to other records in the file.

b. Field descriptions (columns 43-70) which define each field within the records.

The coding form for the Input Specification is illustrated below (see figure 7-1).

Field description entries must start one line below the associated record type descriptions.

FIELD DEFINITIONS.

The fields for the Input Specifications form are defined in the text that follows.

PAGE (COLUMNS 1-2).
Refer to section 2 for a complete description.

LINE (COLUMNS 3-5).
Refer to section 2 for a complete description.

FORM TYPE (COLUMN 6).
This field must contain the letter I.

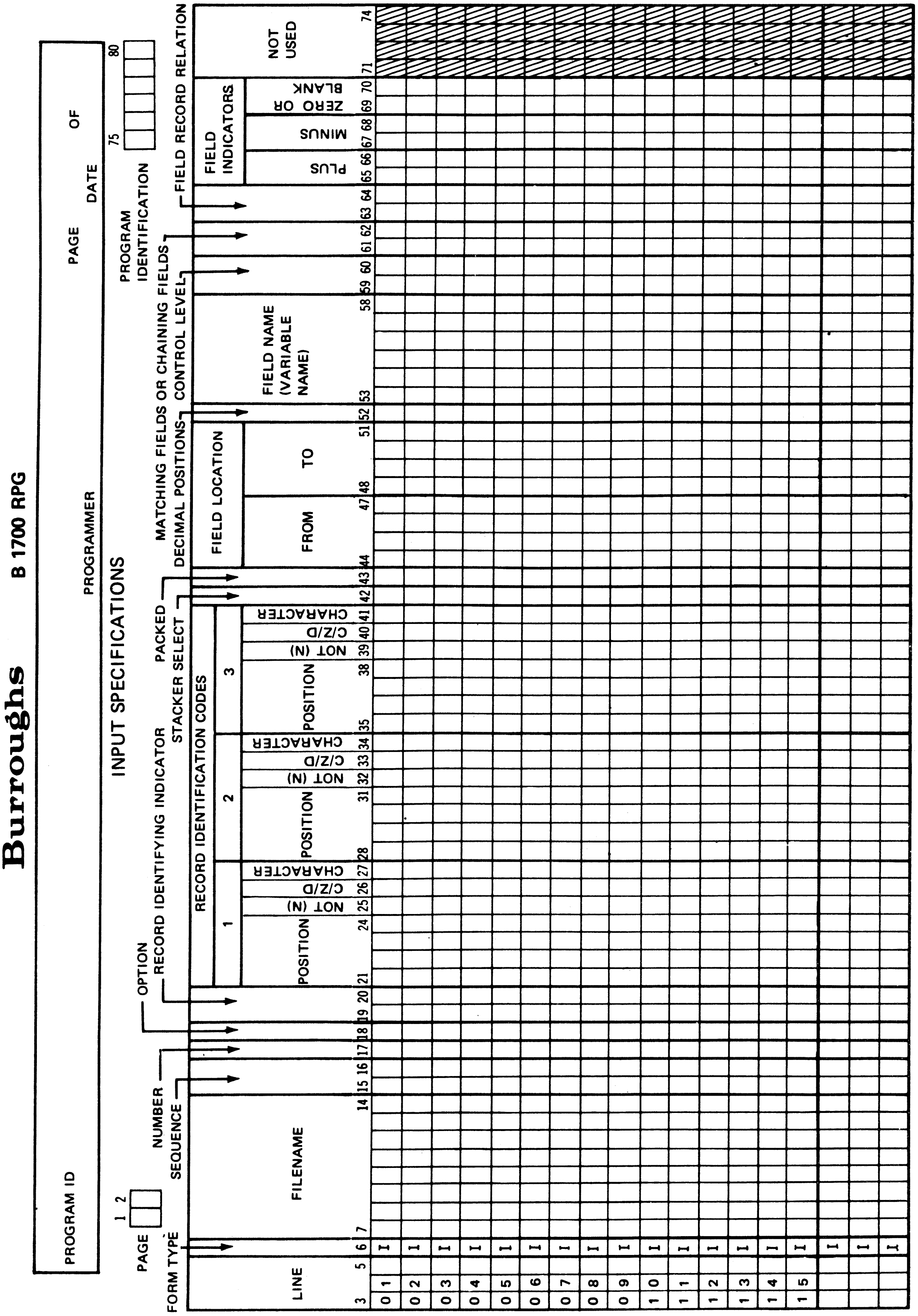
FILENAME (COLUMNS 7-14).
This field is used to identify the file to which the subsequent record type and field descriptions belong. The file specified must have been previously described on the File Description Specifications form. Every input file described in the File Description Specifications must be described on the Input Specification form.

The FILENAME entry must be the same as the one used in the File Description Specifications. It must appear on the first line containing information about the records in the file; if the entry is left blank, the last file name entered is assumed to be the file being described.

SEQUENCE (COLUMNS 15-16).
This field is used to specify a special sequence to different record types in a file. If this field contains an alphabetic entry, it specifies that the record types need not be in any special order.

NOTE

All chained and demand files must have an alphabetic entry in this field.



Burroughs B 1700 RPG

INPUT SPECIFICATIONS

PROGRAM ID

PROGRAMMER

PAGE OF

DATE

PAGE 1 2 □□

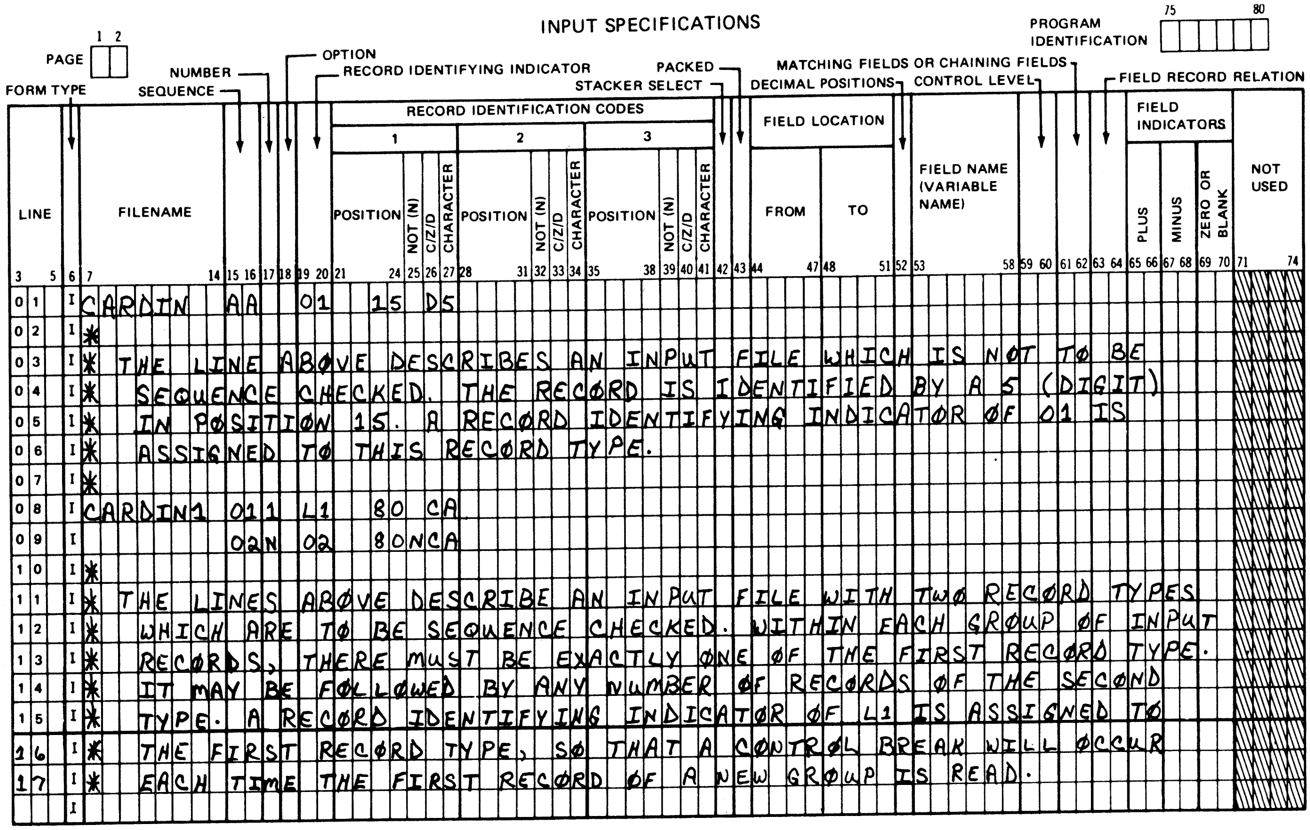
PROGRAM IDENTIFICATION 75 □□□□□□ 80

							RECORD IDENTIFICATION CODES														FIELD LOCATION							FIELD INDICATORS			
							1				2				3																
LINE	FORM TYPE	FILENAME	SEQUENCE	NUMBER	OPTION	RECORD IDENTIFYING INDICATOR	POSITION	NOT (N)	C/Z/D	CHARACTER	POSITION	NOT (N)	C/Z/D	CHARACTER	POSITION	NOT (N)	C/Z/D	CHARACTER	STACKER SELECT	PACKED	FROM	TO	DECIMAL POSITIONS	FIELD NAME (VARIABLE NAME)	CONTROL LEVEL	MATCHING FIELDS OR CHAINING FIELDS	FIELD RECORD RELATION	PLUS	MINUS	ZERO OR BLANK	NOT USED
3 5	6	7 14	15 16	17	18	19 20	21 24	25	26	27	28 31	32	33	34	35 38	39	40	41	42	43	44 47	48 51	52	53 58	59 60	61 62	63 64	65 66	67 68	69 70	71 74
0 1	I																														▨
0 2	I																														▨
0 3	I																														▨
0 4	I																														▨
0 5	I																														▨
0 6	I																														▨
0 7	I																														▨
0 8	I																														▨
0 9	I																														▨
1 0	I																														▨
1 1	I																														▨
1 2	I																														▨
1 3	I																														▨
1 4	I																														▨
1 5	I																														▨
	I																														▨
	I																														▨
	I																														▨

Figure 7-1. Input Specifications Form

Within each file, all record types having alphabetic entries in the SEQUENCE field must be specified before those with numeric entries.

An example of sequence coding (see figure 7-2) is illustrated below:



INPUT SPECIFICATIONS

PAGE 1 2 · PROGRAM IDENTIFICATION 75 80

FORM TYPE · FILENAME · SEQUENCE · NUMBER · OPTION · RECORD IDENTIFYING INDICATOR · RECORD IDENTIFICATION CODES 1 2 3 (POSITION, NOT (N), C/Z/D, CHARACTER) · STACKER SELECT · PACKED · FIELD LOCATION (FROM, TO) · DECIMAL POSITIONS · FIELD NAME (VARIABLE NAME) · CONTROL LEVEL · MATCHING FIELDS OR CHAINING FIELDS · FIELD RECORD RELATION · FIELD INDICATORS (PLUS, MINUS, ZERO OR BLANK) · NOT USED

```
LINE
3  5 6 7      14 15 16 17 18 19 20 21  24 25 26 27
01 ICARDIN  AA  01  15 D5
02 I*
03 I*  THE LINE ABØVE DESCRIBES AN INPUT FILE WHICH IS NØT TØ BE
04 I*   SEQUENCE CHECKED. THE RECØRD IS IDENTIFIED BY A 5 (DIGIT)
05 I*   IN PØSITIØN 15. A RECØRD IDENTIFYING INDICATØR ØF 01 IS
06 I*   ASSIGNED TØ THIS RECØRD TYPE.
07 I*
08 ICARDIN1 011 L1  80 CA
09 I        02N 02  80NCA
10 I*
11 I*  THE LINES ABØVE DESCRIBE AN INPUT FILE WITH TWØ RECØRD TYPES
12 I*   WHICH ARE TØ BE SEQUENCE CHECKED. WITHIN EACH GRØUP ØF INPUT
13 I*   RECØRDS, THERE MUST BE EXACTLY ØNE ØF THE FIRST RECØRD TYPE.
14 I*   IT MAY BE FØLLØWED BY ANY NUMBER ØF RECØRDS ØF THE SECØND
15 I*   TYPE. A RECØRD IDENTIFYING INDICATØR ØF L1 IS ASSIGNED TØ
16 I*   THE FIRST RECØRD TYPE, SØ THAT A CØNTRØL BREAK WILL ØCCUR
17 I*   EACH TIME THE FIRST RECØRD ØF A NEW GRØUP IS READ.
   I
```

Figure 7-2. Input Sequence Checking

When coding this field the programmer must not use the alphabetic entries ND or Rb (b equals blank), because the compiler may mistake them for the ND or R of an AND or OR line.

If this field contains a numeric entry, it assigns sequence numbers to the different types of records in a file. This allows the programmer to specify that one record type must appear before another record type within a sequenced group. The program will automatically check the designated order as the records are read.

If a record is encountered that is out of sequence, the program will halt. The system operator can order the program to resume, at which time it will ignore the record that is out of sequence and read the next record from the file. Refer to "SEQ" error, appendix E.

Records in an AND or OR line cannot have a SEQUENCE field entry; the entry from the previous line also applies to the line with the AND or OR entry.

NUMBER (COLUMN 17).
This field is used only if sequence checking is to be done (i.e., the SEQUENCE field contains a numeric entry), and indicates whether more than one record of the designated type may appear in each group of a sequenced input file (see figure 7-2). Valid entries for this field are:

Entry	Definition
Blank	Record types are not being sequence checked (SEQUENCE field contains alphabetic entry).
1	Not more than one record of this type will be present in each group.
N	Any number of records of this type will be present in each group.

Records in an AND or OR line cannot have a NUMBER field entry; the entry from the previous line also applies to the line with the AND or OR entry.

OPTION (COLUMN 18).
This field is used only if sequence checking is to be done (i.e., the SEQUENCE field contains a numeric entry), and indicates whether certain record types are optional. A zero entry specifies that a record of this type may or may not be present in each group of a sequenced file. If this field is left blank, each group must contain a certain number (as specified in the NUMBER field) of records of this type (see Figure 7-2).

Records in an AND or OR line cannot have an OPTION field entry; the entry from the previous line also applies to the line with the AND or OR entry. Valid entries are:

Entry	Definition
Blank	Record type must be present in each group.
0	Record type is optional, and may or may not be present in each group.

NOTE
If all record types in a file are designated as optional, no sequence errors will be detected.

RECORD IDENTIFYING INDICATOR (COLUMNS 19-20).
This field is used to assign an indicator to each record type. Valid entries are:

Entry	Definition
01-99	Record identifying indicator.
L1-L9	Control level indicator.
LR	Last record indicator.
H0-H9	Halt indicator.
**	Look ahead field.

The various indicators have special meanings, as defined below.

RECORD IDENTIFYING INDICATOR (01-99). Because each input file may contain different types of records (requiring different operations), record identifying indicators are used to signal to the rest of the program cycle the type of record just read. A unique record identifying indicator should be assigned to each record type; when a specific record type is read, its corresponding identifying indicator is turned ON and all other record identifying indicators are turned OFF. This indicator remains on for the rest of the current program cycle and may be used to condition various calculation and output operations, as desired.

Record identifying indicators do not have to be assigned in any order. If the same operations are to be performed on different record types, the same indicator may be assigned to more than one type (refer to OR relationship, RECORD IDENTIFICATION CODES field).

Record identifying indicators are not allowed in an AND line, but indicators may be specified for every record type that requires special processing in an OR relationship. Refer to RECORD IDENTIFICATION CODES field (AND lines) and VARIABLE NAME field (OR lines).

CONTROL LEVEL INDICATOR (L1-L9). A control level indicator is used instead of a record identifying indicator when a record type (rather than a control field) signals the start of a new control group. Refer to the CONTROL LEVEL field for a complete description of control level indicators.

LAST RECORD INDICATOR (LR). The last record indicator is used instead of a record identifying indicator when a record type (rather than automatic End-of-File) signals the end of processing. Final total operations are conditioned by this indicator.

HALT INDICATOR (H0-H9). A halt indicator is used instead of a record identifying indicator when the occurrence of a specific record type denotes an error condition requiring a program halt.

LOOK AHEAD FIELD (**). A pair of asterisks is used to indicate that fields named in the VARIABLE NAME field of subsequent lines are look ahead fields. A look ahead field allows the program to access information in a field of the next record that will be available for processing. Thus, the program may use information from the look ahead field to condition certain operations ahead of the time the record is actually read.

By using a look-ahead field, the program may do things such as:

a. Jobs that the matching record capability cannot handle, and

b. Determining when the last card of a control group is being processed.

RECORD IDENTIFYING CODES (COLUMNS 21-41).
When more than one record type is used within a file, only one record type will be selected for processing during each program cycle. The record identifying indicator for that record type will be turned ON when it is selected and will remain ON for the rest of the current program cycle.

In order to identify the various record types to the program for the purpose of record selection, each record type must have a unique code assigned to it. This code consists of a certain character or combination of characters occurring in certain positions of the record.

The RECORD IDENTIFICATION CODES field is used to describe the code for each record type. If all records are to be processed alike regardless of their type, or if all records are of the same type, this field should be left blank.

AND/OR RELATIONSHIPS. AND/OR relationships are described in the paragraphs that follow.

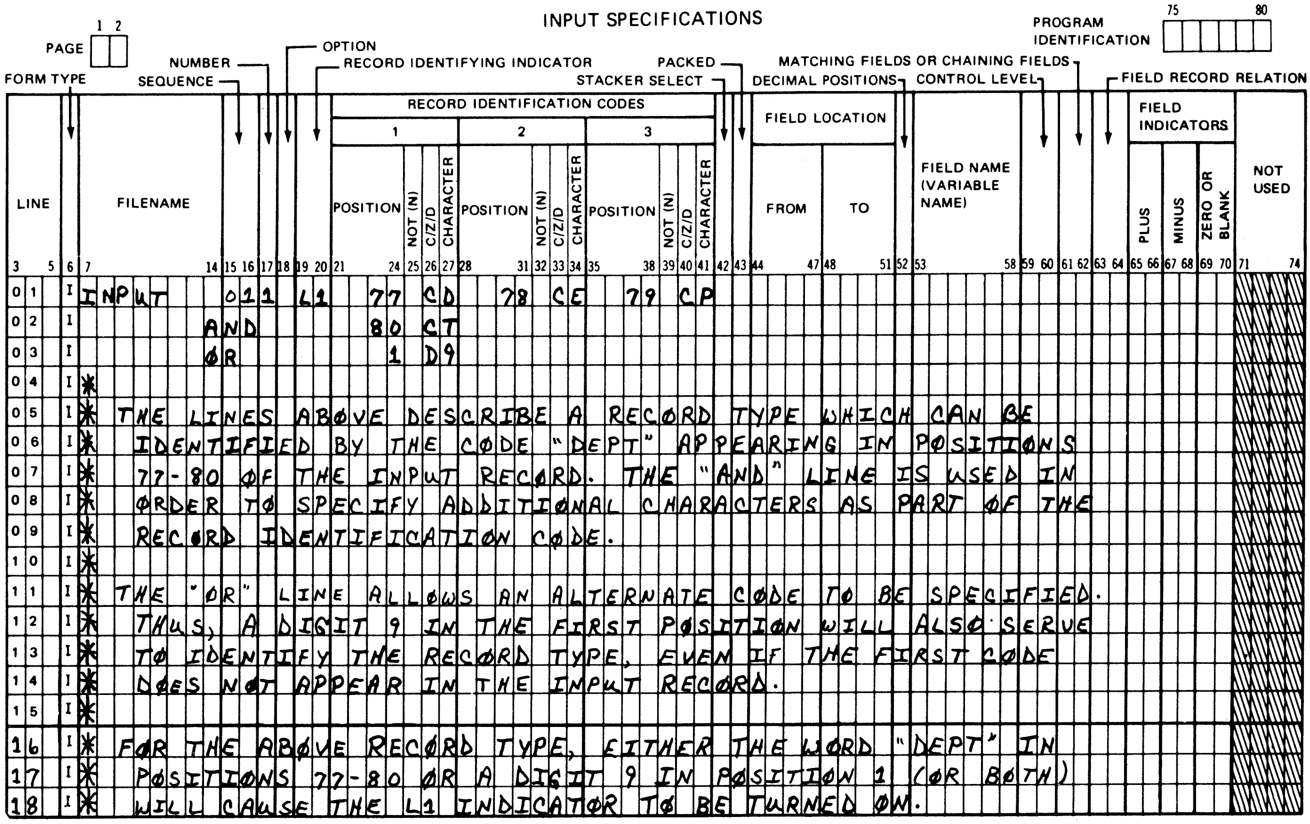
AND Line.

Up to three identification codes may be described on one specification line. An AND line may be used to describe identification codes consisting of more than three characters; and as many AND lines as are necessary to describe the code may be used. Each AND line allows up to three additional identification code characters to be specified. Figure 7-3 illustrates AND/OR relationships.

An AND line is identified by coding the word AND in columns 14-16 (see Figure 7-3). The record selected must contain all the characters specified as the record identification code in order to turn on the corresponding record identifying indicator.

OR Line.

In some cases, a particular record type may be identified by two or more different codes. For this condition, an OR line must be used to indicate that only one of the codes specified need be present to identify the record type. As many OR lines as are necessary to describe the code may be used.



INPUT SPECIFICATIONS

PAGE 1 2 · PROGRAM IDENTIFICATION 75–80 · FORM TYPE · NUMBER · SEQUENCE · OPTION · RECORD IDENTIFYING INDICATOR · PACKED · STACKER SELECT · DECIMAL POSITIONS · MATCHING FIELDS OR CHAINING FIELDS · CONTROL LEVEL · FIELD RECORD RELATION

LINE (3–5) · FILENAME (7–14) · RECORD IDENTIFICATION CODES 1 (POSITION 21–24, NOT (N) 25, C/Z/D 26, CHARACTER 27), 2 (POSITION 28–31, NOT (N) 32, C/Z/D 33, CHARACTER 34), 3 (POSITION 35–38, NOT (N) 39, C/Z/D 40, CHARACTER 41) · FIELD LOCATION FROM (44–47) TO (48–51) · FIELD NAME (VARIABLE NAME) (53–58) · FIELD INDICATORS PLUS (65–66) MINUS (67–68) ZERO OR BLANK (69–70) · NOT USED (71–74)

```
01 IINPUT   011 L1  77 CD  78 CE  79 CP
02 I       AND      80 CT
03 I       ØR        1 D9
04 I*
05 I*  THE LINES ABØVE DESCRIBE A RECØRD TYPE WHICH CAN BE
06 I*   IDENTIFIED BY THE CØDE "DEPT" APPEARING IN PØSITIØNS
07 I*   77-80 ØF THE INPUT RECØRD. THE "AND" LINE IS USED IN
08 I*   ØRDER TØ SPECIFY ADDITIØNAL CHARACTERS AS PART ØF THE
09 I*   RECØRD IDENTIFICATIØN CØDE.
10 I*
11 I*  THE "ØR" LINE ALLØWS AN ALTERNATE CØDE TØ BE SPECIFIED.
12 I*   THUS, A DIGIT 9 IN THE FIRST PØSITIØN WILL ALSØ SERVE
13 I*   TØ IDENTIFY THE RECØRD TYPE, EVEN IF THE FIRST CØDE
14 I*   DØES NØT APPEAR IN THE INPUT RECØRD.
15 I*
16 I*  FØR THE ABØVE RECØRD TYPE, EITHER THE WØRD "DEPT" IN
17 I*   PØSITIØNS 77-80 ØR A DIGIT 9 IN PØSITIØN 1 (ØR BØTH)
18 I*   WILL CAUSE THE L1 INDICATØR TØ BE TURNED ØN.
```

Figure 7-3. AND/OR Relationships

An OR line is identified by coding the word OR in columns 14-15 (see Figure 7-3).

RECORD IDENTIFICATION CODES FIELD. This field is subdivided into three subfields of seven columns each, allowing up to three code characters to be described on one line. The three subfields are taken to be in an AND relationship, and any that are not needed to specify code characters should be left blank. Each of the three subfields are divided into four entries, and coding is the same for all three subfields.

Position (Columns 21-24, 28-31, and 35-38).

These fields are used to give the locations in the record of each character in the record identification code. Entries must be numeric; between one and the record length specified, inclusive; and right-justified (leading zeros are optional).

NOT (N) (Columns 25, 32, and 39).

These fields are used to indicate whether the specified character must be present in the record at the designated position. Valid codes are:

Entry	Definition
Blank	Character must be present in the location specified by the POSITION entry.
N	Character must *not* be present in the location specified by the POSITION entry.

C/Z/D (Columns 26, 33, And 40).

These fields are used to indicate the portion of the character that should be used for comparison: the zone, the digit, or the entire character. Valid entries are:

Entry	Definition
C	Entire character.
Z	Zone portion.
D	Numeric (digit) portion.

CHARACTER (Columns 27, 34, and 41).

Any valid EBCDIC character may be used to identify the input record type. These fields are used to specify the character to be used for comparison as part of the identification code.

If none of the RECORD IDENTIFICATION codes are found on a record, the program will halt. The system operator may request resumption of the program, at which time it will ignore the record in error and read the next record from the same file. See "IDENT" error, appendix E.

STACKER SELECT (COLUMN 42).
This field is used to indicate the stacker into which the input card is to be placed after being read. Only card input or combined files may be stacker selected. Input files may be stacker selected only in the Input Specifications; combined files may be stacker selected either in the Input or Output-Format Specifications. If a combined file is stacker selected in both the Input and Output-Format Specifications, the Output-Format stacker specification overrides the stacker specified in the Input Specifications. Valid entries for this field are:

Entry	Definition
Blank	Cards automatically go to default stacker.
Numeric entry	Stacker into which card type is stacked.

Card types identified by OR lines may be stacker selected for a special stacker by an entry in this field; however, if the STACKER SELECT field entry is left blank, the card type selected by the OR line will go to the default stacker. AND lines may not have an entry in STACKER SELECT.

At execution time, any record types specifying a stacker number higher than that available on the device being used will go to the default stacker.

PACKED (COLUMN 43).
This field is used to specify that an input field is in packed decimal format. Valid entries are:

Entry	Definition
Blank	Field is in unpacked decimal format or is alphanumeric.
P	Field is in packed decimal format.

FIELD LOCATION (COLUMNS 44-51).
This field is used to describe the location of data fields within a record, and is divided into two subfields that specify the beginning (FROM) and ending (TO) positions of the data field. A field of only one character will have the same position number entered in both subfields.

The maximum length allowed for a numeric field is 31 digits, and the maximum allowable length for an alphanumeric field is 255 characters.

Entries in the FROM and TO subfields must be right-justified; leading zeros are optional.

DECIMAL POSITIONS (COLUMN 52).
This field is used to specify the number of positions to the right of the implied decimal point in a numeric field. This entry may not be blank for a numeric field; if the data field only contains integral values, a 0 should be entered to indicate no decimal positions. Valid entries are:

Entry	Definition
Blank	Alphanumeric field.
0-9	Number of decimal positions in a numeric field.

Any field to be used for arithmetic operations or to be edited must be numeric. The number of decimal positions specified cannot exceed the length of the field (as specified in the FIELD LOCATION field).

VARIABLE NAME (COLUMNS 53-58).
This field is used to assign an identifier (name) to an input data field. All fields that will be referenced by the program must be named. Names must be assigned in accordance with the rules for forming field names as described in section 2. A previously defined vector name may be used, which allows loading of the vector during input. Refer to appendix I for a complete discussion of this method of vector loading. A separate line must be used for each field description.

All fields within one record type should have unique names; if two or more fields within the same record have identical names, only the last one defined is used. However, fields from different record types may have the same name, but all names not uniquely defined must have the same length and data type (decimal position entry). These fields do not have to occur in the same location in each record. If matching fields are specified, duplicate variable names should not be used.

OR RELATIONSHIP. To eliminate duplicate coding of identical fields within different record types, the OR relationship may be used. An OR relationship is illustrated below (see figure 7-4). OR lines may be used when:

a. Two or more record types have identical fields in the same record positions (see Figure 7-4), or

b. Two or more record types have some fields which are identical and some fields which differ (see Figure 7-9).

As many OR lines may be used as needed for each sequenced group of records. An OR line is identified by the word OR coded in columns 14-15.

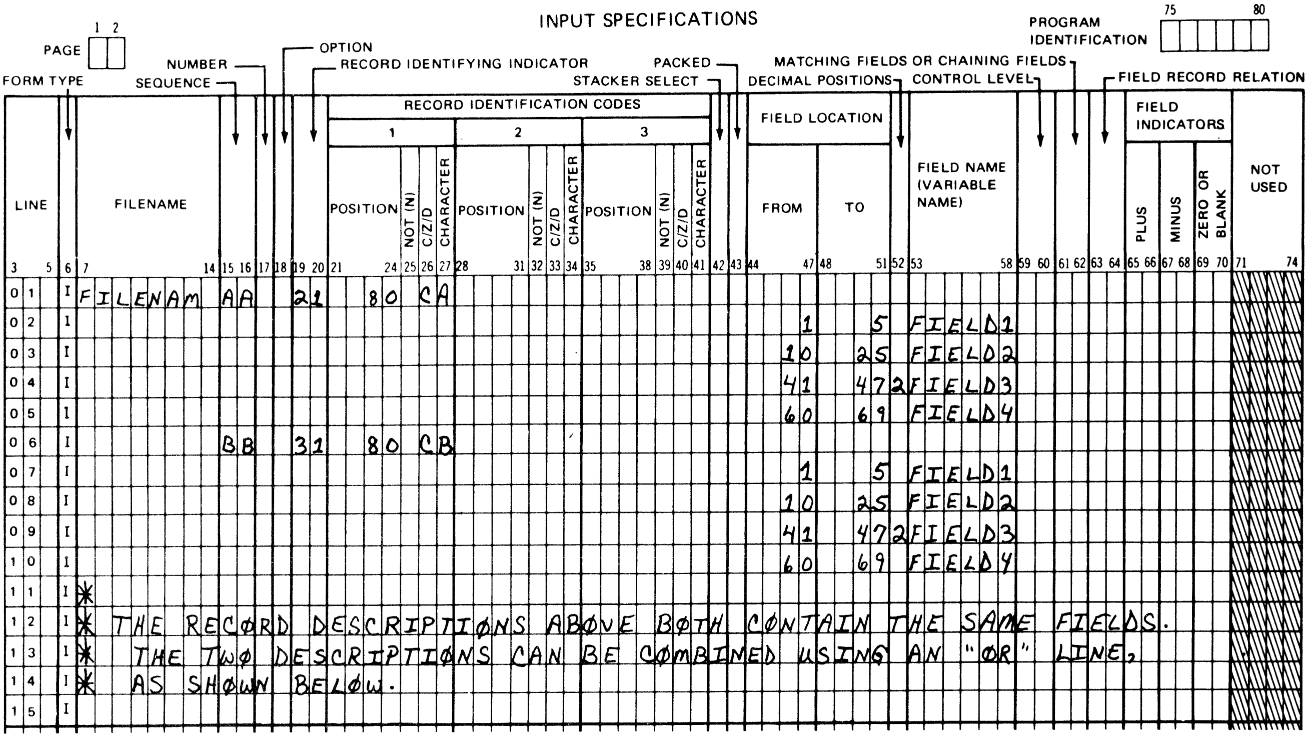
SPECIAL WORDS. The following special words are reserved for use as variable names:

PAGE
PAGE1
PAGE2
UDATE
UMONTH
UDAY
UYEAR

Each special word has a specifically defined usage, as described in the paragraphs that follow:

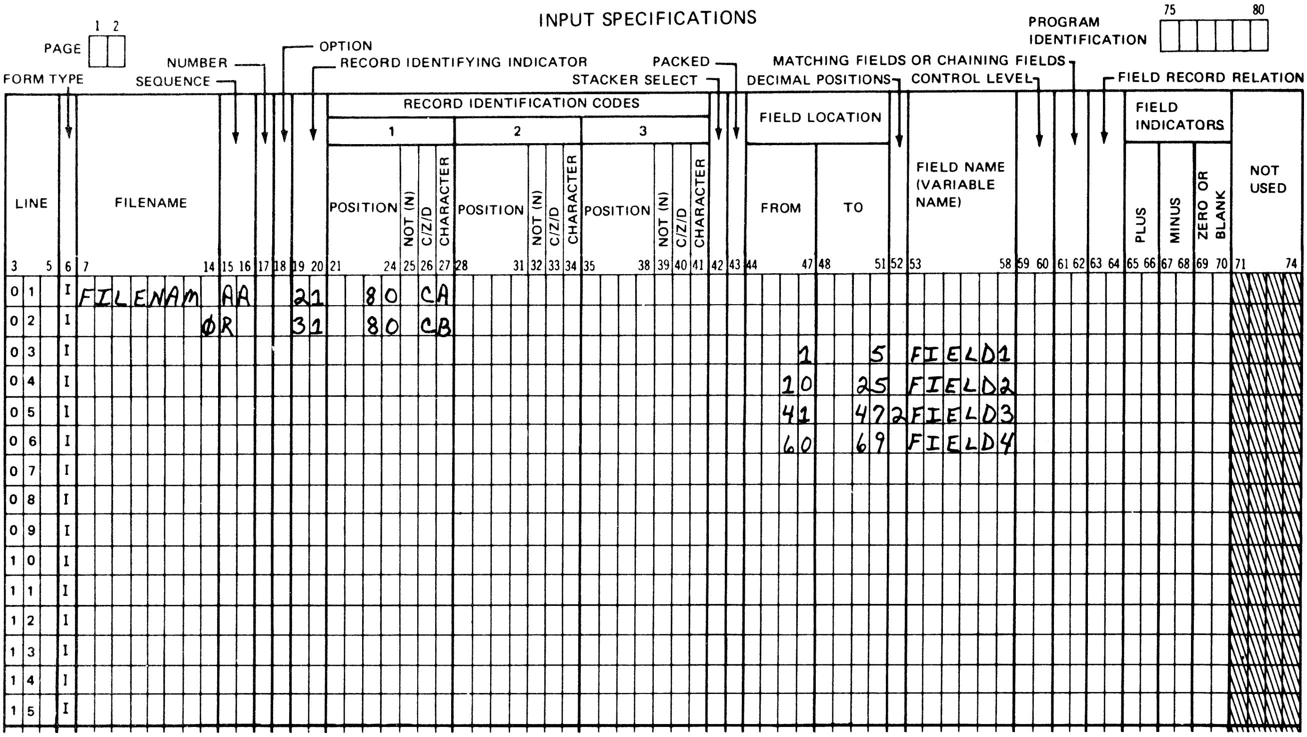
Page Fields (PAGE, PAGE1, and PAGE2).

If page numbering is to be done on output, the special word PAGE (or PAGE1 or PAGE2, for two more printer files) is used to indicate that page numbering is to be done. PAGE field coding is illustrated on the following page (see Figure 7-5).



INPUT SPECIFICATIONS

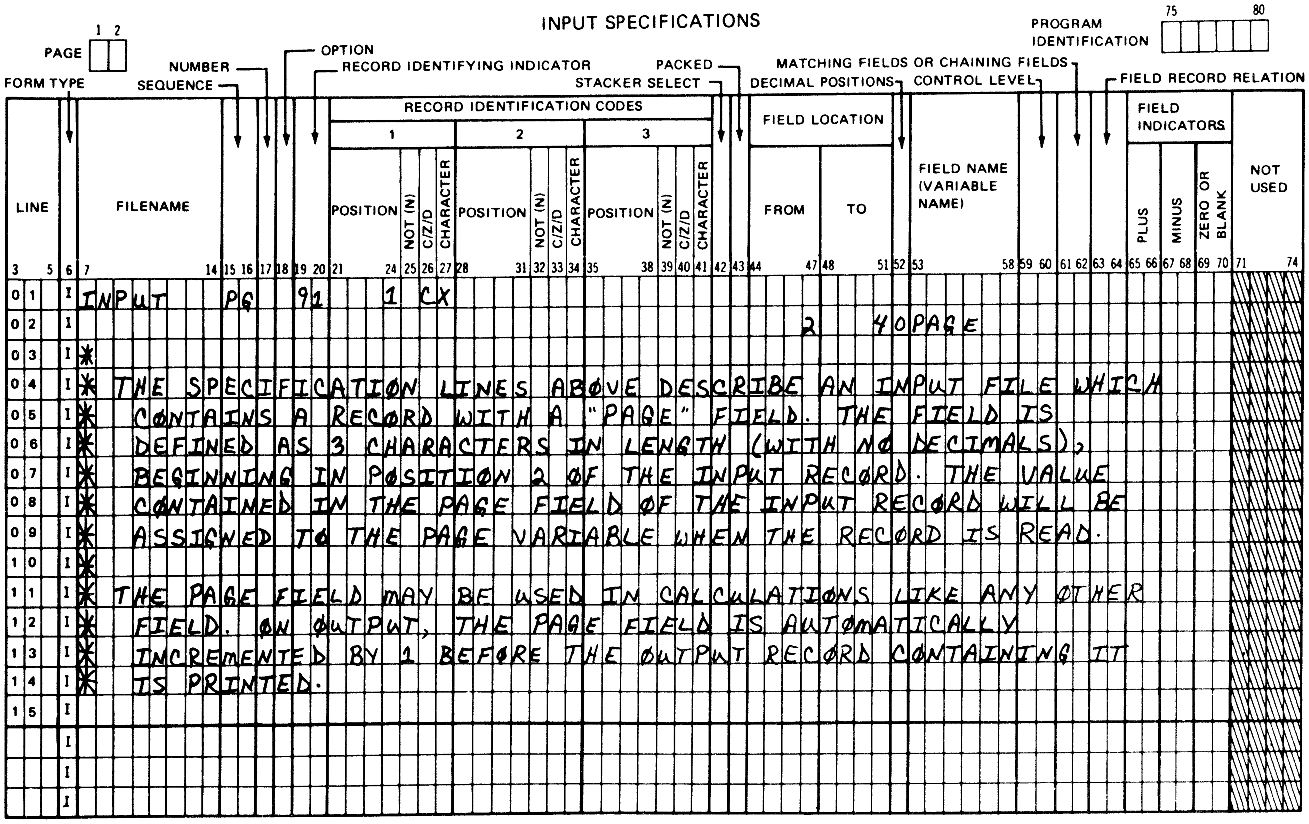
```
01 IFILENAM AA  21  80 CA
02 I                                        1   5 FIELD1
03 I                                       10  25 FIELD2
04 I                                       41  472FIELD3
05 I                                       60  69 FIELD4
06 I        BB  31  80 CB
07 I                                        1   5 FIELD1
08 I                                       10  25 FIELD2
09 I                                       41  472FIELD3
10 I                                       60  69 FIELD4
11 I*
12 I* THE RECØRD DESCRIPTIØNS ABØVE BØTH CØNTAIN THE SAME FIELDS.
13 I*  THE TWØ DESCRIPTIØNS CAN BE CØMBINED USING AN "ØR" LINE,
14 I*  AS SHØWN BELØW.
15 I
```



INPUT SPECIFICATIONS

```
01 IFILENAM AA  21  80 CA
02 I       ØR   31  80 CB
03 I                                        1   5 FIELD1
04 I                                       10  25 FIELD2
05 I                                       41  472FIELD3
06 I                                       60  69 FIELD4
07 I
08 I
09 I
10 I
11 I
12 I
13 I
14 I
15 I
```

Figure 7-4. OR Relationship



INPUT SPECIFICATIONS

PAGE 1 2 — PROGRAM IDENTIFICATION 75 80

FORM TYPE — NUMBER — SEQUENCE — OPTION — RECORD IDENTIFYING INDICATOR — RECORD IDENTIFICATION CODES (1, 2, 3: POSITION, NOT (N), C/Z/D, CHARACTER) — PACKED — STACKER SELECT — FIELD LOCATION (FROM, TO) — DECIMAL POSITIONS — FIELD NAME (VARIABLE NAME) — CONTROL LEVEL — MATCHING FIELDS OR CHAINING FIELDS — FIELD RECORD RELATION — FIELD INDICATORS (PLUS, MINUS, ZERO OR BLANK) — NOT USED

LINE 3 5 — 6 — FILENAME 7 14

```
  01 IINPUT   PG  91   1 CX
  02 I                                        2   40PAGE
  03 I*
  04 I* THE SPECIFICATIØN LINES ABØVE DESCRIBE AN INPUT FILE WHICH
  05 I*   CØNTAINS A RECØRD WITH A "PAGE" FIELD. THE FIELD IS
  06 I*   DEFINED AS 3 CHARACTERS IN LENGTH (WITH NØ DECIMALS),
  07 I*   BEGINNING IN PØSITIØN 2 ØF THE INPUT RECØRD. THE VALUE
  08 I*   CØNTAINED IN THE PAGE FIELD ØF THE INPUT RECØRD WILL BE
  09 I*   ASSIGNED TØ THE PAGE VARIABLE WHEN THE RECØRD IS READ.
  10 I*
  11 I* THE PAGE FIELD MAY BE USED IN CALCULATIØNS LIKE ANY ØTHER
  12 I*   FIELD. ØN ØUTPUT, THE PAGE FIELD IS AUTØMATICALLY
  13 I*   INCREMENTED BY 1 BEFØRE THE ØUTPUT RECØRD CØNTAINING IT
  14 I*   IS PRINTED.
  15 I
     I
     I
     I
```

Figure 7-5. PAGE Field Coding

This feature allows a page number to be entered through a field in an input record, the field called PAGE (or PAGE1 or PAGE2). The page number printed will be one greater than the page number contained in the PAGE field of the input record. The field may be defined as any length, but it must contain no decimal positions. Unless otherwise specified, it is assumed to be four characters in length with no decimal positions (see Figure 7-5). The PAGE field may be used in calculations like any other field.

The same PAGE entry (PAGE, PAGE1, or PAGE2) must not be used for two different output files.

Date Fields (UDATE, UMONTH, UDAY, and UYEAR).

Four special words allow the program to obtain the current value of the date as maintained by the MCP. The following rules apply to date fields:

a. UDATE gives a 6-digit numeric field in one of the following formats (depending upon the entry in the INVERTED PRINT field of the Control Card):

   1) Domestic (MMDDYY), or

2) International/Sterling (DDMMYY).

b. The other three fields; UMONTH, UDAY, and UYEAR; each give a 2-digit field representing the month, day, and year, respectively.

c. These fields may not be changed by any operations within the program; thus, they are usually used only in compare, test, and output operations.

CONTROL LEVEL (COLUMNS 59-60).
This field is used to assign control level indicators to input files (for files other than chained and demand input files). Any field may be assigned a control level indicator, in which case it is known as a Control Field. Control Fields are checked each program cycle for a change in information; when data in the field changes, a Control Break occurs. A group of records with the same information in the Control Field is known as a Control Group. Valid entries for this field are:

Entry	Definition
L1-L9	Control level indicator assigned to field.
Blank	No control level indicator assigned.

A Control Break occurs when a record containing a Control Field is read and the information in that Control Field is different from the information in the same Control Field of the previous record. When a Control Break occurs, the designated control level indicator turns ON, along with all control level indicators lower than it. For example, if control level indicator L5 is turned ON, L4, L3, L2, and L1 are also automatically turned ON. Control level indicator L0 is always ON and cannot be assigned to a Control Field.

Control level indicators are used to condition operations such as:

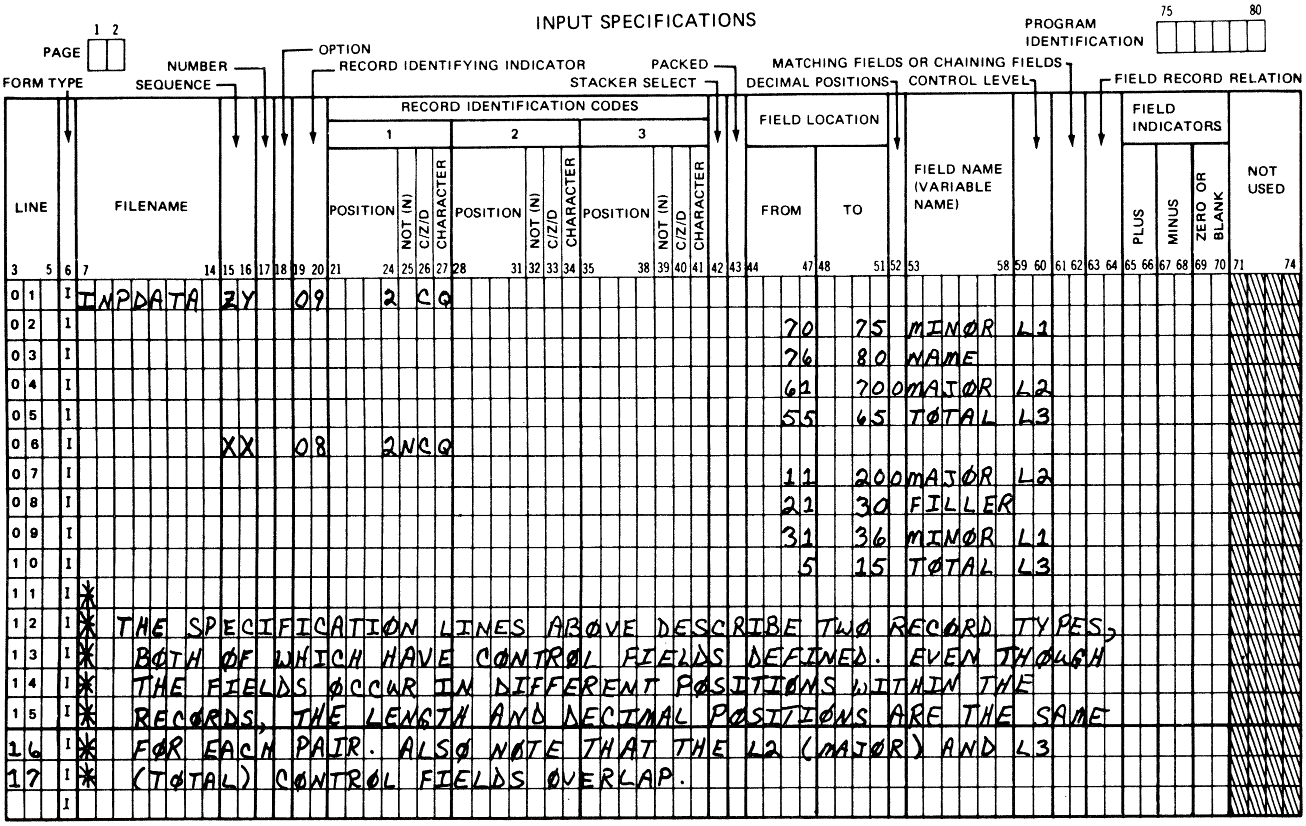
a. Calculations that must be performed when a Control Group changes (totals, etc.),

b. Operations that must be performed on the first card of a new Control Group, or

c. Summary punching or total printing that must be performed for each Control Group.

The following rules must be observed when assigning control level indicators:

a. The same control level indicator may be used in different record types or files; however, the Control Fields associated with that indicator must be of the same length and type. (See figure 7-6).



INPUT SPECIFICATIONS

PAGE 1 2 · PROGRAM IDENTIFICATION 75 80 · FORM TYPE · NUMBER · SEQUENCE · OPTION · RECORD IDENTIFYING INDICATOR · PACKED · STACKER SELECT · MATCHING FIELDS OR CHAINING FIELDS · DECIMAL POSITIONS · CONTROL LEVEL · FIELD RECORD RELATION

LINE · FILENAME · RECORD IDENTIFICATION CODES 1 2 3: POSITION, NOT (N), C/Z/D, CHARACTER · FIELD LOCATION: FROM, TO · FIELD NAME (VARIABLE NAME) · FIELD INDICATORS: PLUS, MINUS, ZERO OR BLANK · NOT USED

```
3 5 6 7      14 15 16 17 18 19 20 21  24 25 26 27 28  31 32 33 34 35  38 39 40 41 42 43 44  47 48  51 52 53   58 59 60 61 62 63 64 65 66 67 68 69 70 71  74
01 IINPDATA ZY  09   2 CQ
02 I                                       70  75 MINØR L1
03 I                                       76  80 NAME
04 I                                       61  700MAJØR L2
05 I                                       55  65 TØTAL L3
06 I        XX  08   2NCQ
07 I                                       11  200MAJØR L2
08 I                                       21  30 FILLER
09 I                                       31  36 MINØR L1
10 I                                        5  15 TØTAL L3
11 I*
12 I* THE SPECIFICATIØN LINES ABØVE DESCRIBE TWØ RECØRD TYPES,
13 I*  BØTH ØF WHICH HAVE CØNTRØL FIELDS DEFINED. EVEN THØUGH
14 I*  THE FIELDS ØCCUR IN DIFFERENT PØSITIØNS WITHIN THE
15 I*  RECØRDS, THE LENGTH AND DECIMAL PØSITIØNS ARE THE SAME
16 I*  FØR EACH PAIR. ALSØ NØTE THAT THE L2 (MAJØR) AND L3
17 I*  (TØTAL) CØNTRØL FIELDS ØVERLAP.
   I
```

Figure 7-6. Control Fields in Two Record Types

b. Field names have no effect upon the control level indicator assigned; therefore, Control Fields in different record types may have both the same name and the same indicator assigned.

c. The total number of positions which may be assigned as Control Fields within each record type is limited to a maximum of 255 characters.

d. Within one record type, Control Fields may overlap.

e. Numeric Control Fields are treated as though they have no decimal positions.

f. For numeric Control Fields, only the digit portion of each character is compared; thus, negative numbers are treated the same as positive numbers.

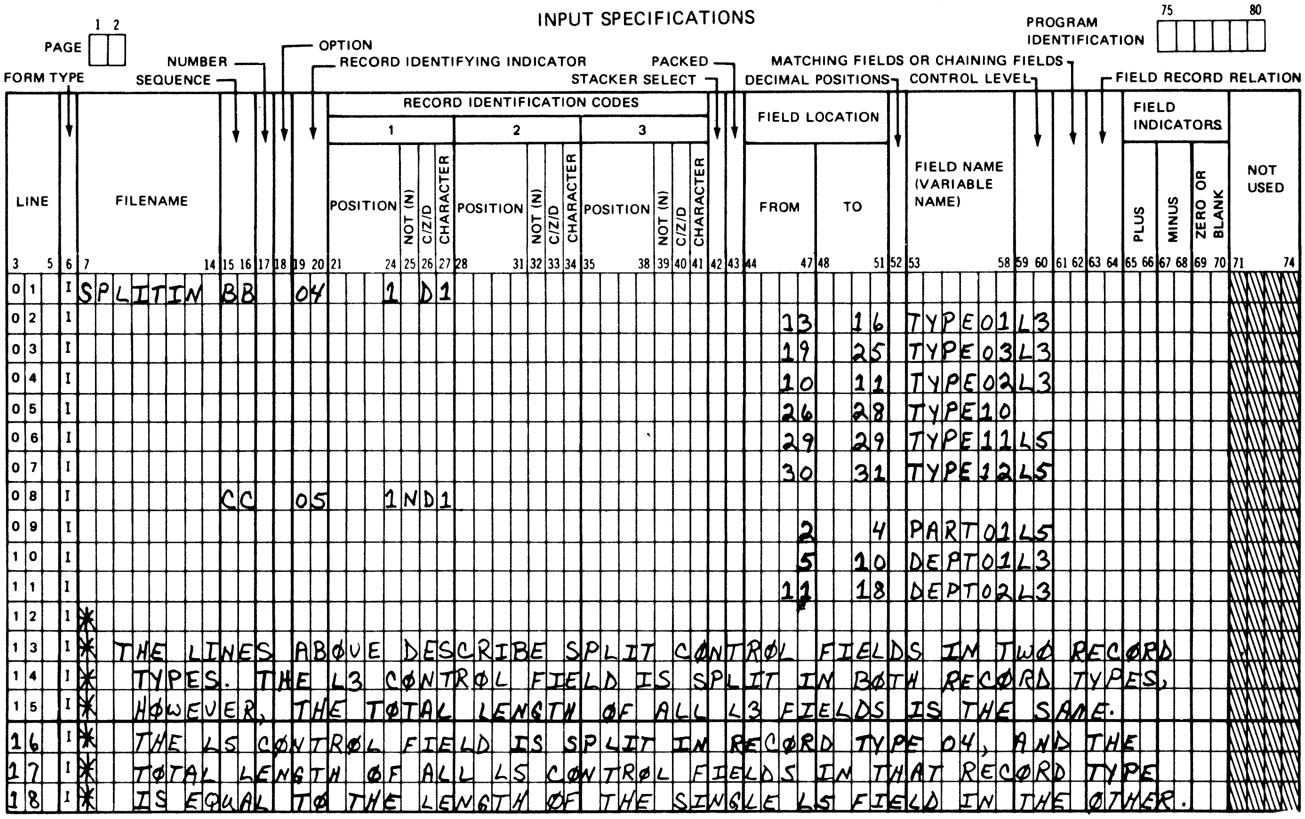
g. All Control Fields with the same control level indicator are considered numeric if any one of those fields is described as numeric.

h. Control levels need not be written in any special sequence, and gaps are permitted in the control levels assigned.

i. Control Fields are initialized to hexadecimal zeros.

j. Total calculations and total output operations are bypassed until the first cycle following a cycle in which a record specifying Control Fields is selected. This prevents a Control Break from occurring the first time a record with Control Fields is read (the input Control Fields would usually not be equal to the value contained in the initialized Control Fields).

SPLIT CONTROL FIELDS. If the same control level indicator is assigned to more than one field within the same record type, the Control Field created is known as a Split Control Field. All fields so designated (those having the same control level within the same record type) are combined by the program in the order specified in the Input Specifications and are treated as one Control Field. Split Control Fields is illustrated below (see figure 7-7).


INPUT SPECIFICATIONS
PAGE
PROGRAM IDENTIFICATION
FORM TYPE
NUMBER
SEQUENCE
OPTION
RECORD IDENTIFYING INDICATOR
RECORD IDENTIFICATION CODES
PACKED
STACKER SELECT
DECIMAL POSITIONS
MATCHING FIELDS OR CHAINING FIELDS
CONTROL LEVEL
FIELD RECORD RELATION
FIELD LOCATION
FIELD INDICATORS
LINE
FILENAME
POSITION
NOT (N)
C/Z/D
CHARACTER
FROM
TO
FIELD NAME (VARIABLE NAME)
PLUS
MINUS
ZERO OR BLANK
NOT USED

```
01 ISPLITIN BB  04   1 D1
02 I                                       13  16 TYPE01L3
03 I                                       19  25 TYPE03L3
04 I                                       10  11 TYPE02L3
05 I                                       26  28 TYPE10
06 I                                       29  29 TYPE11L5
07 I                                       30  31 TYPE12L5
08 I        CC  05   1ND1
09 I                                        2   4 PART01L5
10 I                                        5  10 DEPT01L3
11 I                                       11  18 DEPT02L3
12 I*
13 I*  THE LINES ABØVE DESCRIBE SPLIT CØNTRØL FIELDS IN TWØ RECØRD
14 I*   TYPES. THE L3 CØNTRØL FIELD IS SPLIT IN BØTH RECØRD TYPES,
15 I*   HØWEVER, THE TØTAL LENGTH ØF ALL L3 FIELDS IS THE SAME.
16 I*   THE L5 CØNTRØL FIELD IS SPLIT IN RECØRD TYPE 04, AND THE
17 I*   TØTAL LENGTH ØF ALL L5 CØNTRØL FIELDS IN THAT RECØRD TYPE
18 I*   IS EQUAL TØ THE LENGTH ØF THE SINGLE L5 FIELD IN THE ØTHER.
```

Figure 7-7. Split Control Fields

The following special rules must be observed for Split Control Fields.

a. The same control level indicator may be used for Split Control Fields in different record types if the field names used are different. The length of various portions of a Split Control Field in one record type may be different than the corresponding portions in another record type; in fact, a Control Field may be split in one record type but not in another. However, the total length of the Control Fields (whether split or not) must be the same in both record types.

b. If one portion of a Split Control Field is numeric, the entire field is considered numeric.

c. Any one portion of a numeric Split Control Field may not exceed 31 characters; however, the total length of all fields assigned to one control level indicator (within each record type) may be as great as 255 characters.

d. A mixture of packed and unpacked Control Fields is allowed.

e. No other specification lines may come between lines describing Split Control Fields.

MATCHING OR CHAINING FIELDS (COLUMNS 61-62).
This field is used to indicate matching fields and sequence checking. Valid entries are:

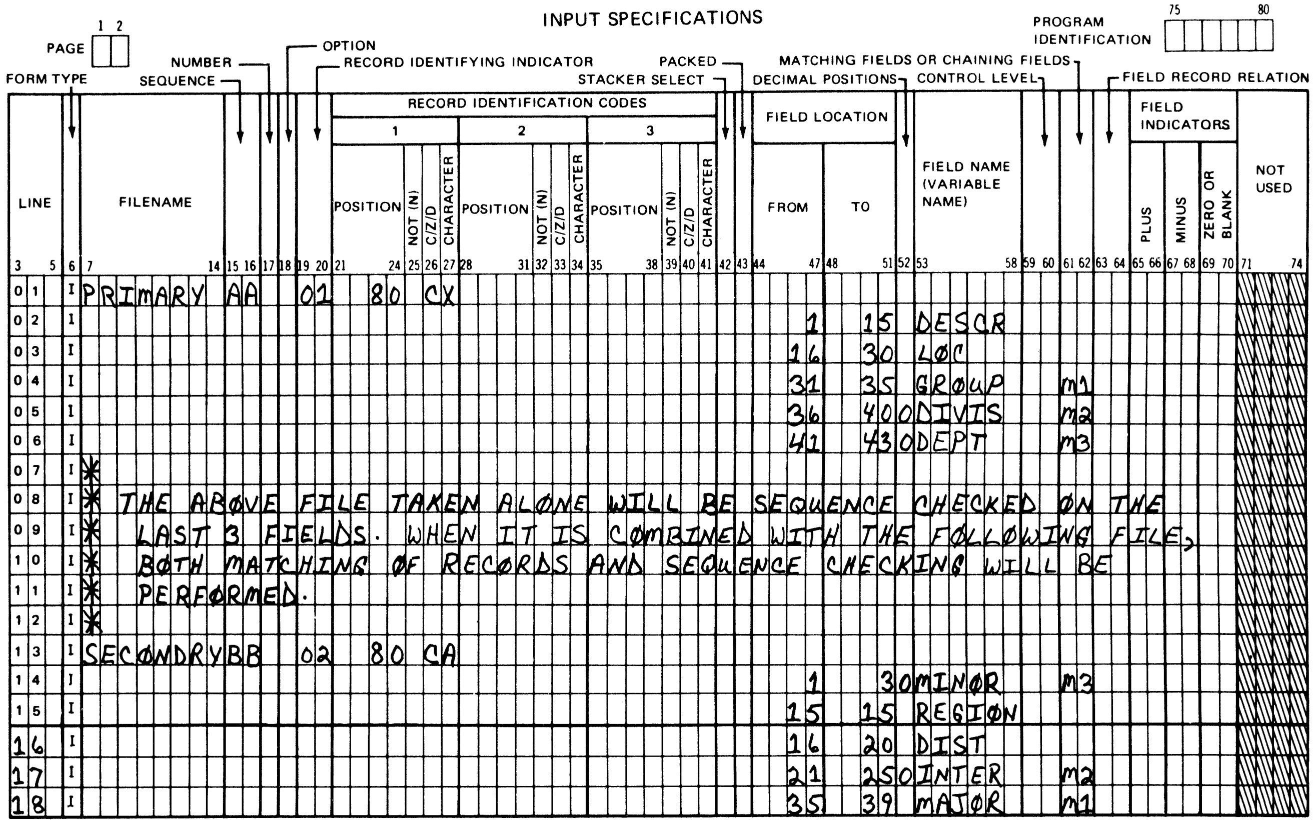
Entry	Definition
Blank	No matching is done.
M1-M9	Matching to be performed when two or more INPUT or COMBINED files specify matching fields. Sequence-checking to be performed when only one INPUT or COMBINED file specifies matching fields.

Matching fields and sequence checking cannot be specified for chained, demand, or table files. The entry in this field designates:

a. Matching fields and sequence checking when two or more INPUT or COMBINED files specify the same matching field value (M1-M9), or

b. Sequence checking only, when only one INPUT or COMBINED file specifies a matching field (M1-M9).

Matching fields and sequence checking is illustrated below (see figure 7-8).



```
01 IPRIMARY AA  01  80 CX
02 I                                        1  15 DESCR
03 I                                       16  30 LØC
04 I                                       31  35 GRØUP   m1
05 I                                       36  400DIVIS   m2
06 I                                       41  430DEPT    m3
07 I*
08 I* THE ABØVE FILE TAKEN ALØNE WILL BE SEQUENCE CHECKED ØN THE
09 I*  LAST 3 FIELDS. WHEN IT IS CØMBINED WITH THE FØLLØWING FILE,
10 I*  BØTH MATCHING ØF RECØRDS AND SEQUENCE CHECKING WILL BE
11 I*  PERFØRMED.
12 I*
13 ISECØNDRYBB  02  80 CA
14 I                                        1  30 MINØR   m3
15 I                                       15  15 REGIØN
16 I                                       16  20 DIST
17 I                                       21  250INTER   m2
18 I                                       35  39 MAJØR   m1
```

Figure 7-8. Matching Fields and Sequence Checking

MATCHING FIELDS. MATCHING field values allow comparison of records from two or more input or combined files to determine when the records match. As many fields as desired (even entire records) can be selected for matching, and as many as nine different fields may be designated as Match Fields (M1-M9). The matching record indicator (MR) will turn ON whenever the Match Fields from a primary file are exactly the same as the Match Fields from a secondary file. The matching field values (M1-M9) are used to specify which fields in each record are to be matched, and cause MR to turn ON when a match occurs. M1-M9 are not indicators; MR is used to condition those operations that should be done only when records match.

The following rules must be observed when assigning MATCHING field values:

a. All Match Fields must be in the same sequence during input, because sequence checking is automatically done on all fields designated as matching fields. A sequence error in any field will cause a program halt. When the system operator resumes program operation, the record in error is ignored and the next record from the same file is read. Refer to "MSEQ" error, appendix E.

b. At least one record type from two files must have Match Fields specified if the files are ever to be matched.

c. All fields given the same MATCHING field value (M1-M9) must be of the same length and type.

d. Overlapping of different Match Fields within one record type is allowed; however, the total length of all Match Fields for one record type must not exceed 255 characters.

e. The same number of Match Fields must be specified for all records used in matching, and the same MATCHING field values must be used for all types.

f. When more than one Match Field is designated for a record type, all fields specified are combined in order by ascending sequence of MATCHING field values and are treated as one contiguous Match Field.

g. Split Match Fields are not allowed; thus, the same MATCHING field value may not be used more than once in one record type.

h. Numeric Match Fields are treated as though they have no decimal positions.

i. For numeric Match Fields, only the digit portion of each character is compared; thus, negative numbers are treated the same as positive numbers.

j. All Match Fields with the same MATCHING field value are considered numeric if any one of the fields is numeric.

k. When more than one MATCHING field value is used for matching records, all Match Fields must match before the MR indicator turns ON.

l. Field names have no effect upon the MATCHING field values assigned; therefore, Match Fields in different record types may have both the same name and the same MATCHING field value assigned.

Processing of matching records proceeds as follows:

a. When a record from the primary file matches a record from the secondary file, the primary file record is processed first.

b. When records do not match, the record with the lowest (ascending files) or highest (descending files) Match Field value is processed first.

c. A record type which has no MATCHING field specification is processed immediately after the record it follows, and the MR indicator is not turned ON. If such a record is the first one in a file, it is processed first (even if it is not in the primary file).

d. Matching records allow the program to enter data from the primary record into the matching secondary record, since the primary record is processed first. Transfer of data from secondary records into matching primary records may be done through the use of look ahead fields.

SEQUENCE CHECKING.
MATCHING field values allow sequence checking of fields within one record type when only one input or combined file specifies MATCHING fields. As many as nine (M1-M9) fields within the record may be selected for sequence checking. When a record is encountered which has a field or fields out of sequence, the program will halt. When the system operator resumes program operation, the record in error is ignored and the next record from the same file is read.

The following rules must be observed when assigning MATCHING field values for sequence checking:

a. All fields designated for sequence checking must be in the same order, either ascending or descending.

b. When more than one field is designated for sequence checking, all fields specified are combined in order by ascending sequence of MATCHING field values and are treated as one contiguous field.

c. Split sequence fields are not allowed; thus, the same MATCHING field value may not be used more than once in the record.

d. Numeric fields are treated as though they have no decimal positions.

e. For numeric fields, only the digit portion of each character is compared; thus, negative numbers are treated the same as positive numbers.

f. All sequence fields are considered numeric if any one of the fields is numeric.

NOTE
Refer to appendix G for the use of this field in IBM's RPG.

FIELD RECORD RELATION (COLUMNS 63-64).
The following rules should be observed when assigning field record relation indicators.

a. All fields within each record type which specify field record relations should be specified after those which do not.

b. Fields within one record type which specify the same field record relation indicator may be entered in any order; however, the most efficient data storage is obtained when they are written as a group on specification lines following one another.

c. All portions of a Split Control Field must be assigned the same field record relation indicator and must be written as a group on specification lines following one another.

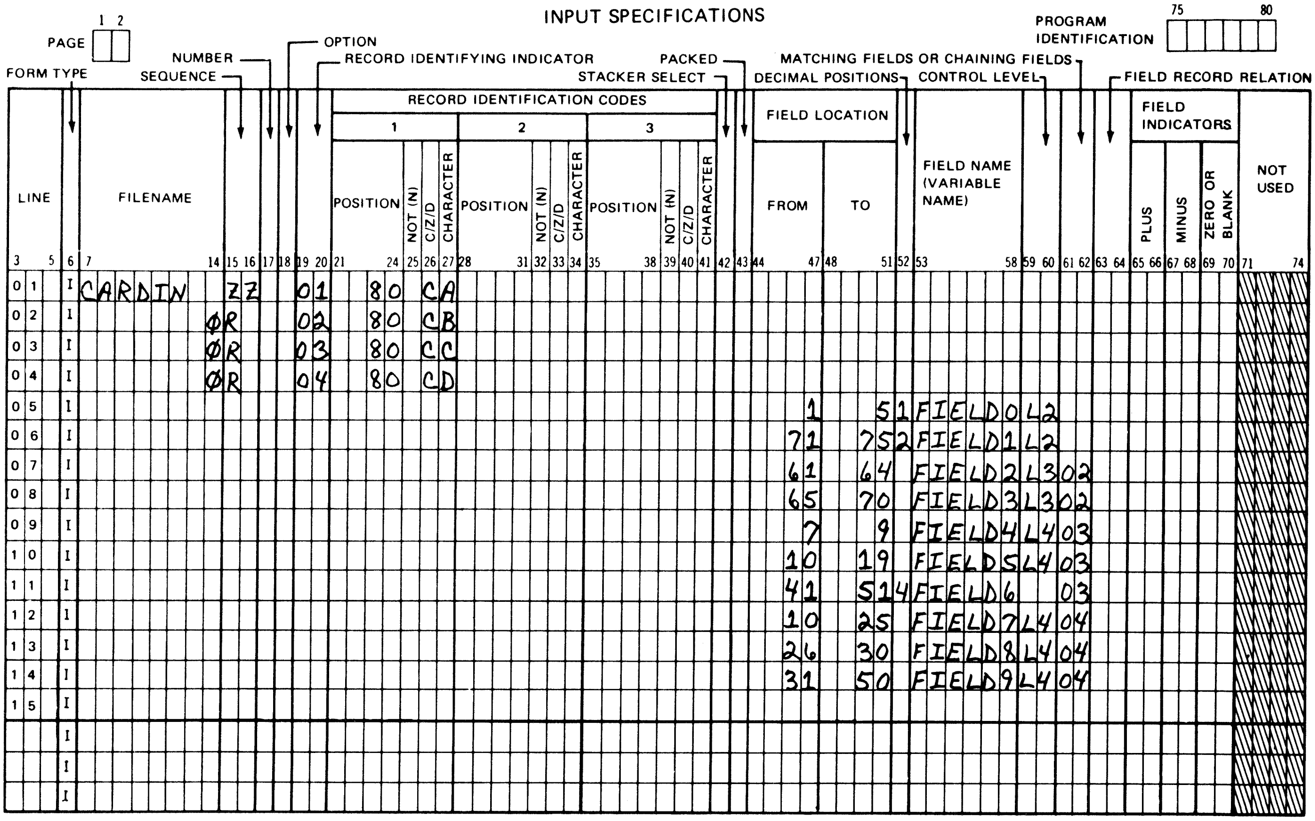
If field record relation indicators are not to be assigned, this field should be left blank (see figure 7-9). The acceptable entries are as follows:

Entry	Definition
Blank	No field record relations.
01-99	Record identifying indicator assigned to a record type.
L1-L9	Control level indicator defined elsewhere.
MR	Matching record indicator.
U1-U8	External indicator defined elsewhere.
H0-H9	Halt indicator defined elsewhere.

RECORD IDENTIFYING INDICATORS (01-99). When several record types have been defined in an OR relationship, all fields defined apply to all record types. In many cases, however, not all of the record types defined have exactly the same fields. The FIELD RECORD RELATION field allows the programmer to specify that some fields only apply to certain record types and not to others. If the FIELD RECORD RELATION field is left blank, the associated field applies to all record types to which it is subordinate. However, by placing the same entry found in the RECORD IDENTIFYING INDICATOR field of one record type in the FIELD RECORD RELATION field, the field is identified as applying only to the corresponding record type.

CONTROL FIELDS and MATCH FIELDS may also be related to a particular record type in an OR relationship by a FIELD RECORD RELATION field entry.

When two CONTROL FIELDS or two MATCH FIELDS have the same control level indicator or matching field value, respectively, only one of the two fields may have a field record relation indicator assigned. Thus, only the specification with the field record relation indicator is used when that indicator is ON; when none of the field record relation indicators is ON, only the specification without any field record relation indicator is used.



INPUT SPECIFICATIONS

PAGE 1 2

PROGRAM IDENTIFICATION 75 80

LINE 3-5	FORM TYPE 6	FILENAME 7-14	SEQUENCE 15-16	NUMBER 17	OPTION 18	RECORD IDENTIFYING INDICATOR 19-20	RECORD IDENTIFICATION CODES 1 POSITION 21-24	1 NOT (N) 25	1 C/Z/D 26	1 CHARACTER 27	2 POSITION 28-31	2 NOT (N) 32	2 C/Z/D 33	2 CHARACTER 34	3 POSITION 35-38	3 NOT (N) 39	3 C/Z/D 40	3 CHARACTER 41	STACKER SELECT 42	PACKED 43	FIELD LOCATION FROM 44-47	FIELD LOCATION TO 48-51	DECIMAL POSITIONS 52	FIELD NAME (VARIABLE NAME) 53-58	CONTROL LEVEL 59-60	MATCHING FIELDS OR CHAINING FIELDS 61-62	FIELD RECORD RELATION 63-64	FIELD INDICATORS PLUS 65-66	FIELD INDICATORS MINUS 67-68	FIELD INDICATORS ZERO OR BLANK 69-70	NOT USED 71-74
01	I	CARDIN	ZZ			01	80		C	A																					
02	I		ØR			02	80		C	B																					
03	I		ØR			03	80		C	C																					
04	I		ØR			04	80		C	D																					
05	I																				1	5	1	FIELD0	L2						
06	I																				71	75	2	FIELD1	L2						
07	I																				61	64		FIELD2	L3	02					
08	I																				65	70		FIELD3	L3	02					
09	I																				7	9		FIELD4	L4	03					
10	I																				10	19		FIELD5	L4	03					
11	I																				41	51	4	FIELD6		03					
12	I																				10	25		FIELD7	L4	04					
13	I																				26	30		FIELD8	L4	04					
14	I																				31	50		FIELD9	L4	04					
15	I																														
	I																														
	I																														
	I																														

Figure 7-9. Field Record Relation

CONTROL LEVEL (L1-L9) AND MATCHING RECORD (MR) INDICATORS. Certain fields may only be used when a special condition, such as a control break or matching records, occurs. To indicate the condition under which data will be accepted from a particular field, the control level (L1-L9) or matching record (MR) indicator is used. Data from the associated field will be accepted only when the specified indicator is ON.

EXTERNAL INDICATORS (U1-U8). External indicators are used primarily to condition files in the File Description Specifications (see columns 71-72 - FILE CONDITION). However, they may also be used to condition fields even though file conditioning is not specified. Data from the associated field will be accepted only when the specified indicator is ON.

HALT INDICATORS (H0-H9). A halt indicator is used to associate a field with a record type in an OR relationship which has the same halt indicator specified in the RECORD IDENTIFYING INDICATOR field.

FIELD INDICATORS (COLUMNS 65-70).
This field is divided into three subfields to allow testing of a field for a condition of plus (positive), minus (negative), or zero. The indicator specified turns ON if the condition tested is True, and turns

OFF if the condition tested is False. Thus, these indicators may be used to condition certain calculation or output operations. Valid entries are:

Entry	Definition
Blank	No field indicators used.
01-99	Field indicator.
H0-H9	Halt indicator.

The FIELD INDICATOR field is subdivided as described in the paragraphs that follow.

PLUS (COLUMNS 65-66). Any valid indicator specified in this field will turn ON if the corresponding data field is greater than zero (numeric field only) or greater than blank (numeric or alphanumeric field).

MINUS (COLUMNS 67-68). Any valid indicator specified in this field will turn ON if the corresponding data field is less than zero (numeric field only) or less than blank (numeric or alphanumeric field).

ZERO OR BLANK (COLUMNS 69-70). Any valid indicator specified in this field will turn ON if the corresponding data field is zero (numeric field only) or blank (numeric or alphanumeric field).

The following rules must be observed when assigning and using field indicators:

a. All field indicators are OFF at the beginning of the program, and remain OFF until the condition being tested is satisfied on the input record just read.

b. A field may be assigned more than one indicator; however, only the indicator specified for the condition with a true result will be turned ON. All other indicators assigned to the field will be turned OFF.

c. The state of a field indicator assigned to fields in different record types is always determined by the last record selected.

d. The state of a field indicator assigned to more than one field within one record type is determined by the last field to which it is assigned.

e. When different field indicators are assigned to fields in different record types, a field indicator will remain ON (or OFF) until another record of the same type is selected.

f. Indicators specified in the FIELD INDICATOR field may be affected by the SETON or SETOF operators.

g. If a halt indicator specified in the FIELD INDICATOR field is turned ON as a result of the corresponding condition being true, the program will halt *after* the input record which caused it to turn ON has been completely processed.

NOT USED (COLUMNS 71-74).
This field is ignored by the compiler and should be left blank.

PROGRAM IDENTIFICATION (COLUMNS 75-80).
Refer to secton 2 for a complete description.

SECTION 8

CALCULATION SPECIFICATIONS

GENERAL.

Calculation Specifications are used to describe the operations to be performed on the data and to specify the order in which those operations are to be performed. Calculation Specifications are of three types, which must appear in the following order:

a. Detail calculations.
b. Total calculations.
c. Subroutines.

Within each grouping, operations are performed in the order in which they are written. The Calculation Specifications coding form is illustrated below (see figure 8-1).

Each specification line describes one operation, and is divided into three functional parts:

a. Conditions under which the operation is to be performed (columns 7-17).

b. The kind of operation to be performed and the data which is to be operated upon (columns 18-53).

c. Tests to be made upon the results of the operation (columns 54-59).

FIELD DEFINITIONS.

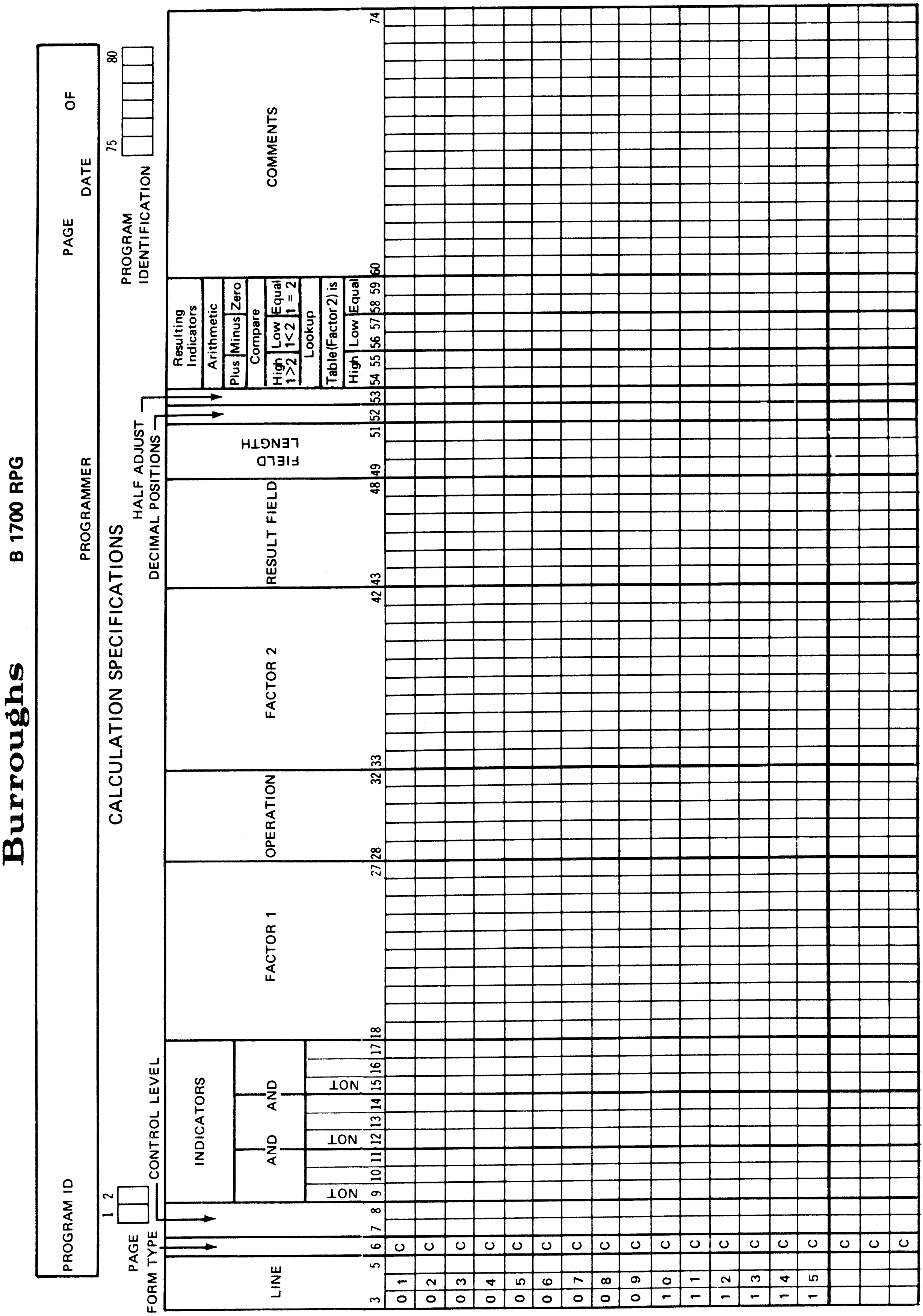
The fields for the Calculation Specifications form are defined in the text that follows.

PAGE (COLUMNS 1-2).
Refer to section 2 for a complete description.

LINE (COLUMNS 3-5).
Refer to section 2 for complete description.

FORM TYPE (COLUMN 6).
A C must appear in this field.

CONTROL LEVEL (COLUMNS 7-8).
The valid entries for this field are listed below and each entry is described in the paragraphs that follow.



Burroughs B 1700 RPG

PROGRAM ID PROGRAMMER PAGE DATE OF

CALCULATION SPECIFICATIONS

PROGRAM IDENTIFICATION 75 80

PAGE 1 2

LINE	FORM TYPE	CONTROL LEVEL	INDICATORS					FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	Resulting Indicators						COMMENTS
				AND		AND									Arithmetic Compare Lookup						
															Plus High 1>2 Table(Factor2) is High		Minus Low 1<2 Low		Zero Equal 1 = 2 Equal		
			NOT		NOT		NOT														
3 5	6	7 8	9	10 11	12	13 14	15 16 17	18 27	28 32	33 42	43 48	49 51	52	53	54	55	56	57	58	59	60 74
0 1	C																				
0 2	C																				
0 3	C																				
0 4	C																				
0 5	C																				
0 6	C																				
0 7	C																				
0 8	C																				
0 9	C																				
1 0	C																				
1 1	C																				
1 2	C																				
1 3	C																				
1 4	C																				
1 5	C																				
	C																				
	C																				
	C																				

Figure 8-1. Calculation Specifications Form

Entry	Definition
Blank	Calculation operation is performed during every cycle, depending upon conditions specified in columns 9-17 (INDICATORS).
L0-L9	Calculation operation is performed only when the designated control break occurs.
LR	Calculation operation is performed after the last record has been processed.
SR	Calculation operation is part of a subroutine.
AN, OR	Establishes AND and OR relationship between lines of indicators.

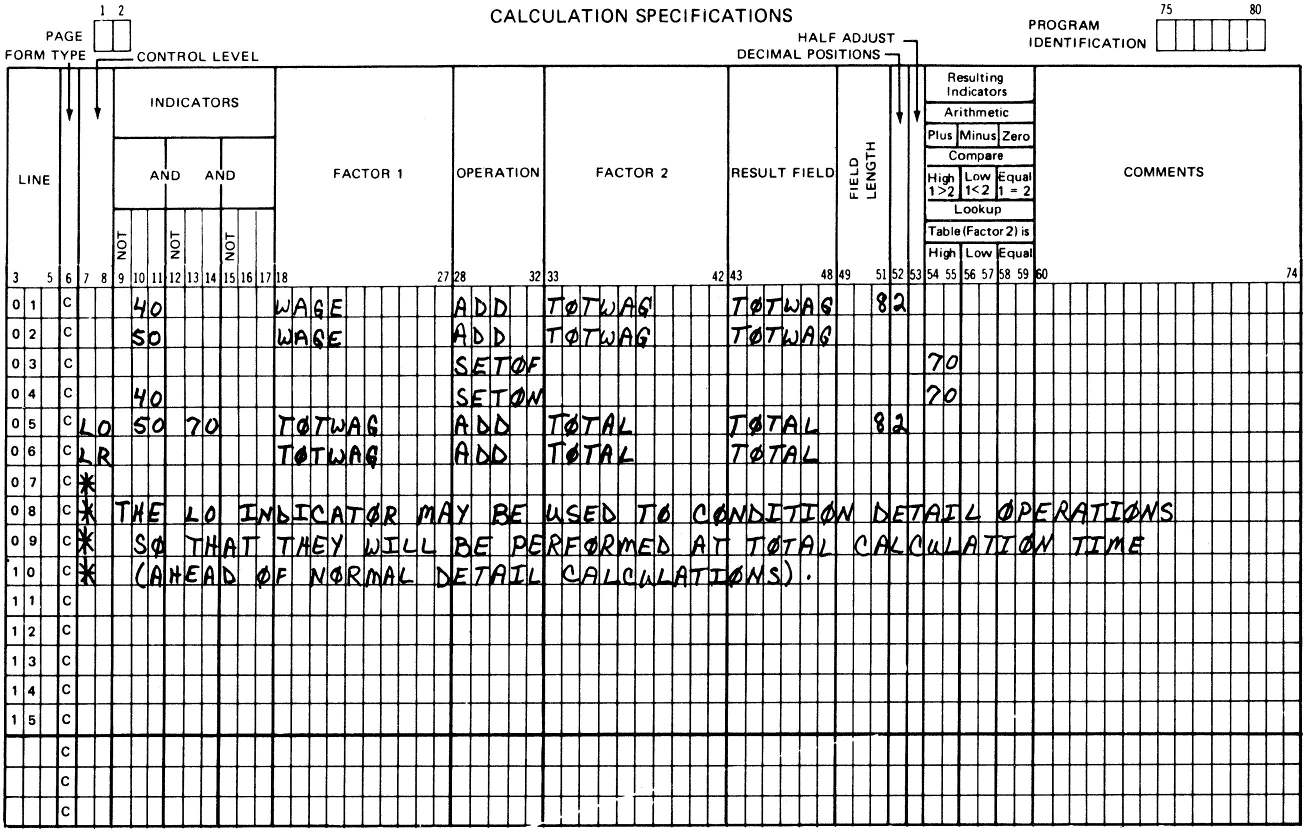
CONTROL.LEVEL INDICATORS (L0-L9). Control level indicators are used to condition operations that are to occur only at a Control Break. If a control level indicator is specified in this field, the operation described on the same specification line is done only when the designated indicator is ON. When a Control Break for a certain level occurs, all lower control level indicators also turn ON. However, when a control level indicator used as a record identifying indicator turns ON to indicate a specific record type, or when a control level indicator is turned ON by a SETON operation, all lower-level indicators remain OFF.

The L0 indicator remains ON during the entire program. If no other control level indicators are assigned, but it is desired to perform total calculation and total output operations, the L0 indicator may be used to condition those operations. L0 indicator coding is illustrated below (see figure 8-2).

LAST RECORD INDICATOR (LR). The LR indicator automatically turns ON when the last record has been read and processed (all control level indicators L1-L9 also turn ON). This indicator is used to condition those operations that are to be performed at End-of-File.

SUBROUTINES (SR). The SR entry is not an indicator, rather it is used to indicate that the specification on the same line is part of a subroutine. All subroutine lines must be specified last.

AND/OR LINES (AN, OR). This field may also be used to specify that lines of indicators are in an AND or OR relationship. There is no limit to the number of AND or OR lines which may be specified for each operation. The last line of a group in an AND or OR relationship contains the Operation Code and all operands. All previous lines in the group must contain blanks in columns 18-59. The first line of a group



CALCULATION SPECIFICATIONS

PAGE □□ (1 2) PROGRAM IDENTIFICATION □□□□□□ (75 80)

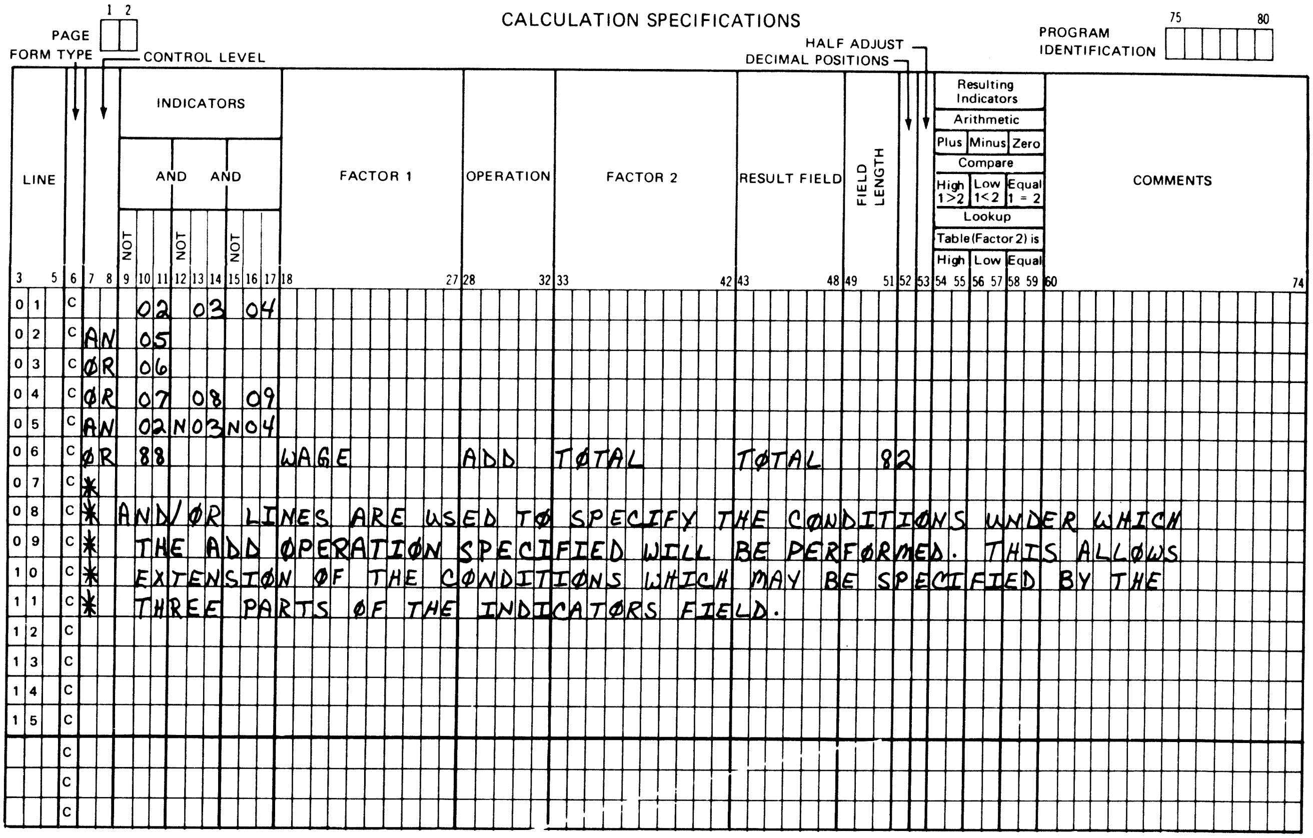
LINE (3-5)	FORM TYPE (6)	CONTROL LEVEL (7-8)	INDICATORS (9-11)	AND (12-14)	AND (15-17)	FACTOR 1 (18-27)	OPERATION (28-32)	FACTOR 2 (33-42)	RESULT FIELD (43-48)	FIELD LENGTH (49-51)	DECIMAL POSITIONS (52)	HALF ADJUST (53)	Plus / High 1>2 (54-55)	Minus / Low 1<2 (56-57)	Zero / Equal 1 = 2 (58-59)	COMMENTS (60-74)
01	C		40			WAGE	ADD	TØTWAG	TØTWAG	8	2					
02	C		50			WAGE	ADD	TØTWAG	TØTWAG							
03	C						SETØF						70			
04	C		40				SETØN						70			
05	C	L0	50	70		TØTWAG	ADD	TØTAL	TØTAL	8	2					
06	C	LR				TØTWAG	ADD	TØTAL	TØTAL							
07	C	*														
08	C	*	THE L0 INDICATØR MAY BE USED TØ CØNDITIØN DETAIL ØPERATIØNS													
09	C	*	SØ THAT THEY WILL BE PERFØRMED AT TØTAL CALCULATIØN TIME													
10	C	*	(AHEAD ØF NØRMAL DETAIL CALCULATIØNS).													
11	C															
12	C															
13	C															
14	C															
15	C															
	C															
	C															
	C															

Figure 8-2. L0 Indicator

may contain an L0-L9, LR, or SR entry if the entire group is conditioned by a control level indicator or is part of a subroutine. AND/OR relationship is illustrated below (see figure 8-3).

INDICATORS (COLUMNS 9-17).
This field is divided into three subfields such that up to three indicators on each line may be specified to condition a calculation operation. Each subfield is divided into two parts, as follows:

a. NOT (one column).



```
01 C   02 03 04
02 CAN 05
03 CØR 06
04 CØR 07 08 09
05 CAN 02N03N04
06 CØR 88      WAGE      ADD  TØTAL     TØTAL   82
07 C*
08 C* AND/ØR LINES ARE USED TØ SPECIFY THE CØNDITIØNS UNDER WHICH
09 C*   THE ADD ØPERATIØN SPECIFIED WILL BE PERFØRMED. THIS ALLØWS
10 C*   EXTENSIØN ØF THE CØNDITIØNS WHICH MAY BE SPECIFIED BY THE
11 C*   THREE PARTS ØF THE INDICATØRS FIELD.
12 C
13 C
14 C
15 C
```

Figure 8-3. AND/OR Relationship

b. INDICATOR (two columns).

The NOT portion is used to specify that the associated indicator must be OFF in order for the operation to occur. If this condition is desired, an N must be entered in the NOT position. Otherwise, the NOT portion must be left blank.

The INDICATOR portion is used to specify the indicator to be tested for ON (NOT = blank) or OFF (NOT = N). The following entries are allowed in this portion of the INDICATORS field:

Entry	Definition
Blank	Operation not conditioned by any indicator.
01-99	Operation conditioned by indicator used elsewhere in the program.
L1-L9	Operation conditioned by control level indicator previously assigned.
LR	Operation conditioned by last record indicator.

Entry	Definition
MR	Operation conditioned by matching record indicator.
H0-H9	Operation conditioned by halt indicator used elsewhere in the program.
U1-U8	Operation conditioned by external indicator previously defined.
OA-OG, OV	Operation conditioned by overflow indicator previously defined.

All three indicators on one line are in an AND relationship. All indicators on one line (or grouped lines), plus the Control Level indicator (if used) must be ON or OFF as specified, in order for the associated operation to take place.

NUMERIC INDICATORS (01-99). The numeric indicators (01-99) may be used as follows:

a. To condition calculation operations that are only to be performed for specific input record types (record identifying indicators).

b. To condition calculation operations that are only to be done when an input field meets certain conditions (field indicators).

c. To condition calculation operations according to the results of previous operations in the Calculation Specifications (resulting indicators).

CONTROL LEVEL INDICATORS (L1-L9). Control level indicators assigned in the CONTROL LEVEL field (Input Specifications) or the RESULTING INDICATORS field (Calculation Specifications) may be used to condition operations that are to be performed only on the first record of a new Control Group.

LAST RECORD INDICATOR (LR). The last record indicator is used to condition operations that are to be performed at End-of-Job.

MATCHING RECORDS INDICATOR (MR). The matching records indicator is used to condition operations that are to be performed only when matching input records are found.

HALT INDICATORS (H0-H9). Halt indicators previously assigned in the FIELD INDICATORS field (Input Specifications) or the RESULTING INDICATORS field (Calculation Specifications) may be used to condition operations that are to be performed only when an error condition occurs.

Since the program does not halt until *after* the record in error has been completely processed, some operations must be prevented in order to avoid erroneous results. By using the halt indicators in conjunction with a N in the NOT portion of the INDICATORS field, an operation may be inhibited when the specified halt indicator is ON.

EXTERNAL INDICATORS (U1-U8). External indicators may be used to condition operations that are to be performed only when the specified indicator has been set at execution time. See FILE CONDITION field (section 4) for complete description of external indicators.

OVERFLOW INDICATORS (OA-OG, OV). Overflow indicators previously assigned in the OVERFLOW INDICATOR field (File Description Specifications) may be used to condition operations that are to be performed when the overflow line on a printer file has been reached.

The following special rules must be observed in specifying indicators:

a. When a control level indicator (L1-L9) is used in the INDICATORS field and not in the CONTROL LEVEL field, the operation conditioned by the designated indicator is performed only for the record which caused the Control Break or any higher-level Control Break.

b. All operations conditioned by control level indicators specified in the CONTROL LEVEL field are performed before those conditioned by control level indicators specified in the INDICATORS field.

c. If a control level indicator is specified in the CONTROL LEVEL field and the matching record indicator (MR) is specified in the INDICATORS field, MR indicates a matching condition in the previous record (not the record just read, which caused the Control Break). After all operations conditioned by control level indicators in the CONTROL LEVEL field are done, MR then indicates the matching condition on the record just read.

FACTOR 1 (COLUMNS 18-27).
This field is used to supply the data to be operated upon by the Operation Code specified in columns 28-32. Allowable entries are:

a. The name of any field previously defined.

b. A literal (alphanumeric or numeric).

c. The name of a subroutine (BEGSR operation only).

d. A table or array name (vector name).

e. The special names UDATE, UMONTH, UDAY, UYEAR, PAGE, PAGE1, and PAGE2.

f. A label (TAG or ENDSR operation only).

Entries in this field must be left-justified, and are dependent upon the particular operation being specified (refer to appendix C).

OPERATION (COLUMNS 28-32).
This field is used to specify the proper Operation Code for the type of operation to be performed using FACTOR 1 and FACTOR 2. The Operation Code must be left-justified in the field.

Operations are performed in the order in which they appear on the Calculation Specifications sheet; however, all operations conditioned by control level indicators specified in the CONTROL LEVEL field (except those which belong to subroutines) must appear after those operations not conditioned by control level indicators.

Refer to appendix C for a complete list of Operation Codes and their functions.

FACTOR 2 (COLUMNS 33-42).
This field is used to supply the data to be operated upon by the Operation Code specified in columns 28-32. Allowable entries are:

a. The name of any field previously defined.

b. A literal (alphanumeric or numeric).

c. The name of a subroutine (EXSR operation only).

d. A table or array name (vector name).

e. The special names UDATE, UMONTH, UDAY, UYEAR, PAGE, PAGE1, and PAGE2.

f. A label (GOTO operation only).

g. A filename (DEBUG, DSPLY, or FORCE operation only).

Entries in this field must be left-justified, and are dependent upon the particular operation being specified (refer to appendix C).

RESULT FIELD (COLUMNS 43-48).
This field is used to name the field, vector, or vector element that will be used to store the results of the operation specified. The use of this field is dependent upon the particular operation being specified (refer to appendix C). Any field, vector, or vector element previously defined in the Input or Calculation Specifications may be used as a RESULT FIELD.

Definition of a new field is also allowed, and is accomplished by entering a field name that has not been previously used. If a new field is defined, the FIELD LENGTH and DECIMAL POSITIONS must also be declared for that field. The field name specified must be entered left-justified, must begin with an alphabetic character, and may contain no imbedded blanks or special characters.

FIELD LENGTH (COLUMNS 49-51).

This field is used to specify the length of a result field that has not been previously defined. Numeric fields are limited to a maximum length of 31 characters, while alphanumeric fields may be as long as 255 characters. Allowable entries for this field are:

Entry	Definition
Blank	RESULT FIELD previously defined.
1-255	RESULT FIELD length.

It is allowable to enter the length of a field that has been previously defined; however, the length and number of decimal positions specified must be the same as that previously defined.

Entries in this field must be right-justified, and leading zeros are optional.

DECIMAL POSITIONS (COLUMN 52).

This field is used to specify the number of positions to the right of the implied decimal point in a numeric result field. If the RESULT FIELD is alphanumeric, this field must be left blank. If the RESULT FIELD is a numeric field which has been previously defined, an entry is optional. If an entry is made, however, it must agree with the previous entry specified for the field. The valid entries are:

Entry	Definition
Blank	Alphanumeric field (or numeric field previously defined).
0-9	Number of decimal positions in a numeric field.

Previously undefined numeric result fields must have the number of decimal positions specified; if the field only contains integral values, a 0 should be entered to indicate no decimal positions.

The number of decimal positions specified must not exceed the length of the RESULT FIELD, as specified by the FIELD LENGTH entry.

HALF ADJUST (COLUMN 53).

This field is used to indicate whether or not the contents of the RESULT FIELD are to be rounded. Rounding is accomplished by adding 5 (or -5 for negative values) to the digit to the right of the last decimal position (as specified by the DECIMAL POSITIONS entry). Then all digits to the right of the last decimal position are dropped. Valid entries for this field are:

Entry	Definition
Blank	Do not half adjust
H	Half adjust.

Entries in this field are only allowable for numeric result fields.

RESULTING INDICATORS (COLUMNS 54-59).
This field may be used for the following purposes:

a. To test the value of the RESULT FIELD after an arithmetic operation.

b. To check the result of a CHAIN or READ operation.

c. To check the result of a COMP, LOKUP, or TESTZ operation.

d. To specify the indicators that are to be set by a SETON or SETOF operation.

Valid entries for this field are:

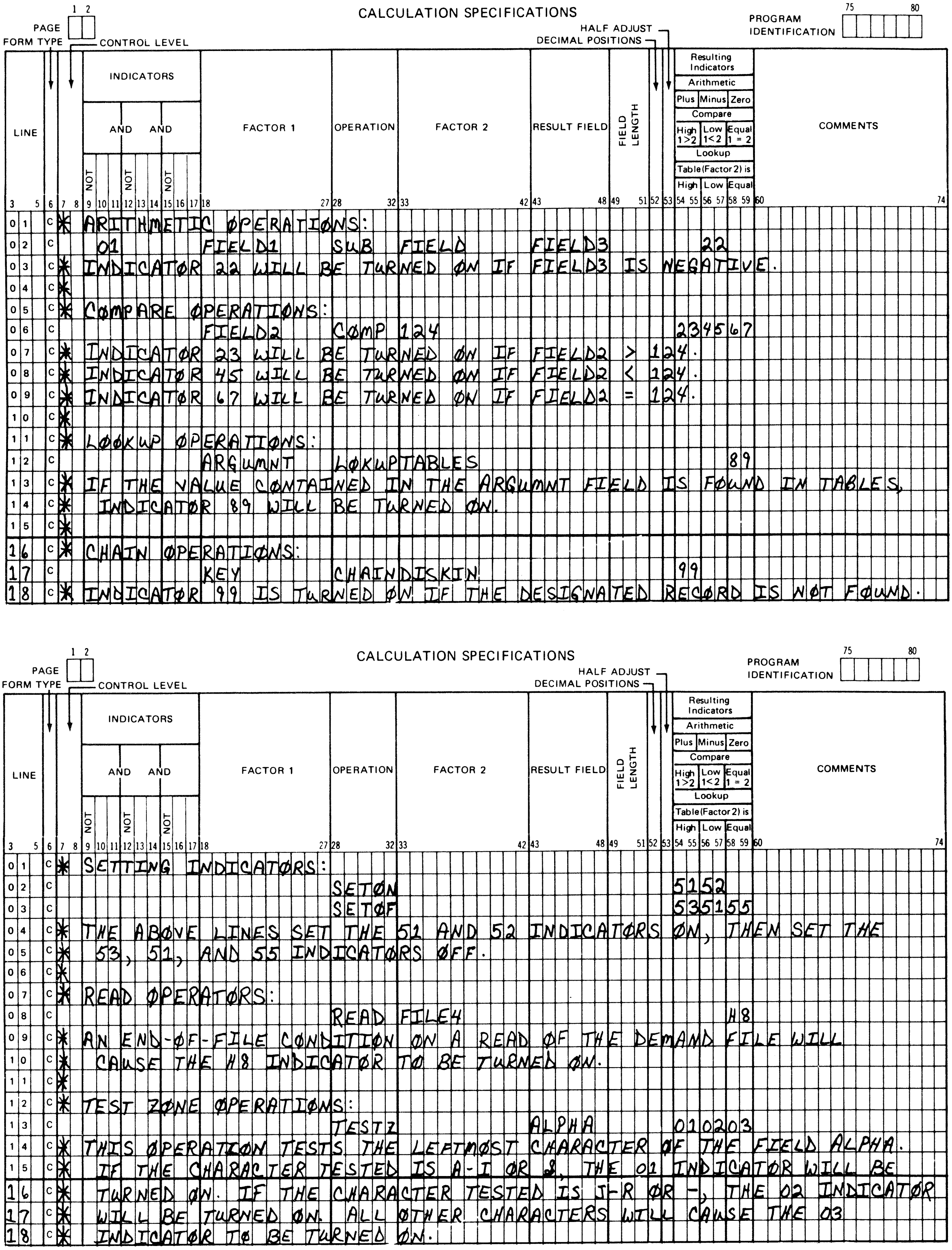
Entry	Definition
01-99	Any numeric indicator.
L1-L9	Any control level indicator.
LR	Last record indicator.
H0-H9	Any halt indicator.
OA-OG, OV	Any overflow indicator.

Examples of Resulting indicators are illustrated below (see figure 8-4). This figure can be used in conjunction with the text that follows.

NOTE
Resulting Indicators may not be specified when the RESULT field is an array name. The only exception being a look-up operation.

ARITHMETIC RESULTS. The arithmetic results of this field are described in the paragraphs that follow.

Plus (Columns 54-55).
Any indicator entered in this field will be turned ON if the result of an arithmetic operation is positive.



CALCULATION SPECIFICATIONS

```
01 C* ARITHMETIC ØPERATIØNS:
02 C   01      FIELD1    SUB  FIELD     FIELD3       22
03 C* INDICATØR 22 WILL BE TURNED ØN IF FIELD3 IS NEGATIVE.
04 C*
05 C* CØMPARE ØPERATIØNS:
06 C           FIELD2    CØMP 124                  234567
07 C* INDICATØR 23 WILL BE TURNED ØN IF FIELD2 > 124.
08 C* INDICATØR 45 WILL BE TURNED ØN IF FIELD2 < 124.
09 C* INDICATØR 67 WILL BE TURNED ØN IF FIELD2 = 124.
10 C*
11 C* LØØKUP ØPERATIØNS:
12 C           ARGUMNT   LØKUPTABLES                   89
13 C* IF THE VALUE CØNTAINED IN THE ARGUMNT FIELD IS FØUND IN TABLES,
14 C*  INDICATØR 89 WILL BE TURNED ØN.
15 C*
16 C* CHAIN ØPERATIØNS:
17 C           KEY       CHAINDISKIN               99
18 C* INDICATØR 99 IS TURNED ØN IF THE DESIGNATED RECØRD IS NØT FØUND.
```

CALCULATION SPECIFICATIONS

```
01 C* SETTING INDICATØRS:
02 C                    SETØN                     5152
03 C                    SETØF                     535155
04 C* THE ABØVE LINES SET THE 51 AND 52 INDICATØRS ØN, THEN SET THE
05 C*  53, 51, AND 55 INDICATØRS ØFF.
06 C*
07 C* READ ØPERATØRS:
08 C                    READ FILE4                    H8
09 C* AN END-ØF-FILE CØNDITIØN ØN A READ ØF THE DEMAND FILE WILL
10 C*  CAUSE THE H8 INDICATØR TØ BE TURNED ØN.
11 C*
12 C* TEST ZØNE ØPERATIØNS:
13 C                    TESTZ          ALPHA      010203
14 C* THIS ØPERATIØN TESTS THE LEFTMØST CHARACTER ØF THE FIELD ALPHA.
15 C*  IF THE CHARACTER TESTED IS A-I ØR &, THE 01 INDICATØR WILL BE
16 C*  TURNED ØN. IF THE CHARACTER TESTED IS J-R ØR -, THE 02 INDICATØR
17 C*  WILL BE TURNED ØN. ALL ØTHER CHARACTERS WILL CAUSE THE 03
18 C*  INDICATØR TØ BE TURNED ØN.
```

Figure 8-4. Resulting Indicators

Minus (Columns 56-57).
Any indicator entered in this field will be turned ON if the result of an arithmetic operation is negative.

Zero (Columns 58-59).
Any indicator entered in this field will be turned ON if the result of an arithmetic operation is zero.

COMP OPERATION RESULTS. The COMP results for this field are described in the paragraphs that follow.

High (Columns 54-55).
Any indicator entered in this field will be turned ON if FACTOR 1 is greater than FACTOR 2

Low (Columns 56-57).
Any indicator entered in this field will be turned ON if FACTOR 1 is less than FACTOR 2.

Equal (Columns 58-59).
Any indicator entered in this field will be turned ON if FACTOR 1 is equal to FACTOR 2.

LOKUP OPERATION RESULTS. The results for LOKUP operation are described in the text that follows.

High (Columns 54-55).
Any indicator entered in this field will be turned ON if FACTOR 2 is greater than FACTOR 1.

Low (Columns 56-57).
Any indicator entered in this field will be turned ON if FACTOR 2 is less than FACTOR 1.

Equal (Columns 58-59).
Any indicator entered in this field will be turned ON if FACTOR 2 is equal to FACTOR 1.

CHAIN OPERATION RESULTS. An entry is not required in these fields. However, if the desired record is not found and subsequent operations assume that it was, erroneous results may occur.

If an indicator is specified in columns 54-55 and the desired record is not found, the specified indicator will turn ON. No output is permitted to a chained update file when the record is not found, unless a record is to be added.

NOTE

> Columns 56-57 should be left blank; however, it *may* contain an entry, which *must* be the same entry as in columns 54-55. Columns 58-59 must be blank.

READ OPERATION RESULTS. An indicator should be specified in columns 58-59, if not, an End-of-File condition will cause the program to be discontinued. If End-of-File is reached and an indicator is specified, that indicator will be turned ON. Any subsequent attempts to READ from the same file will cause the program to be discontinued.

TESTZ OPERATION RESULTS. The results of TESTZ Operation are described in the text that follows.

Plus (Columns 54-55).
Any indicator entered in this field will be turned ON if the character under test is A-I or &.

Minus (Columns 56-57).
Any indicator entered in this field will be turned ON if the character under test is J-R or -.

Zero (Columns 58-59).
Any indicator entered in this field will be turned ON if the character under test is anything other than A-R, &, or -.

SETON AND SETOF OPERATIONS. Up to three indicators may be turned ON (SETON) or OFF (SETOF) on one specification line. The following rules must be observed when using the SETON and SETOF operations:

a. The following indicators may not be specified for either operation: 1P, MR, LO, or U1-U8.

b. Setting a control level indicator (L1-L9) ON or OFF does not affect any other control level indicator.

c. All control level and record identifying operations are automatically turned OFF after detail output operations are completed, regardless of any previous SETON or SETOF operations.

d. If any halt indicators (H0-H9) are set ON and are not turned OFF before the detail output operations finish, the program will halt.

e. If the LR indicator is turned ON by a SETON operation which is conditioned by a control level indicator (columns 7-8 on Calculation Specifications sheet), the program will stop after all total output operations are completed. If the LR indicator is turned ON by a SETON operation not conditioned by a control level indicator, the program stops after the next total output operation is completed.

COMMENTS (COLUMNS 60-74).
This field is available for inclusion of comments and documentary remarks, and may contain any valid EBCDIC characters.

PROGRAM IDENTIFICATION (COLUMNS 75-80).
Refer to section 2 for a complete description.

SECTION 9

OUTPUT-FORMAT SPECIFICATIONS

GENERAL.

Output-Format Specifications describe the records within each file and fields within each record to be used as output from the program. There are two types of Output-Format Specifications, as follows:

a. Record type descriptions (columns 7-31) which define the various output records and their relationship to other records in the file.

b. Field descriptions (columns 23-74) which define each field within the records.

Field description entries must start one line below the associated record type descriptions.

The standard Output-Format Specification form is illustrated below (see figure 9-1).

FIELD DEFINITIONS.

The fields for the Output-Format Specifications form are defined in the text that follows.

PAGE (COLUMNS 1-2).
Refer to section 2 for complete description.

LINE (COLUMNS 3-5).
Refer to section 2 for complete description.

FORM TYPE (COLUMN 6).
An alpha "O" must appear in this field.

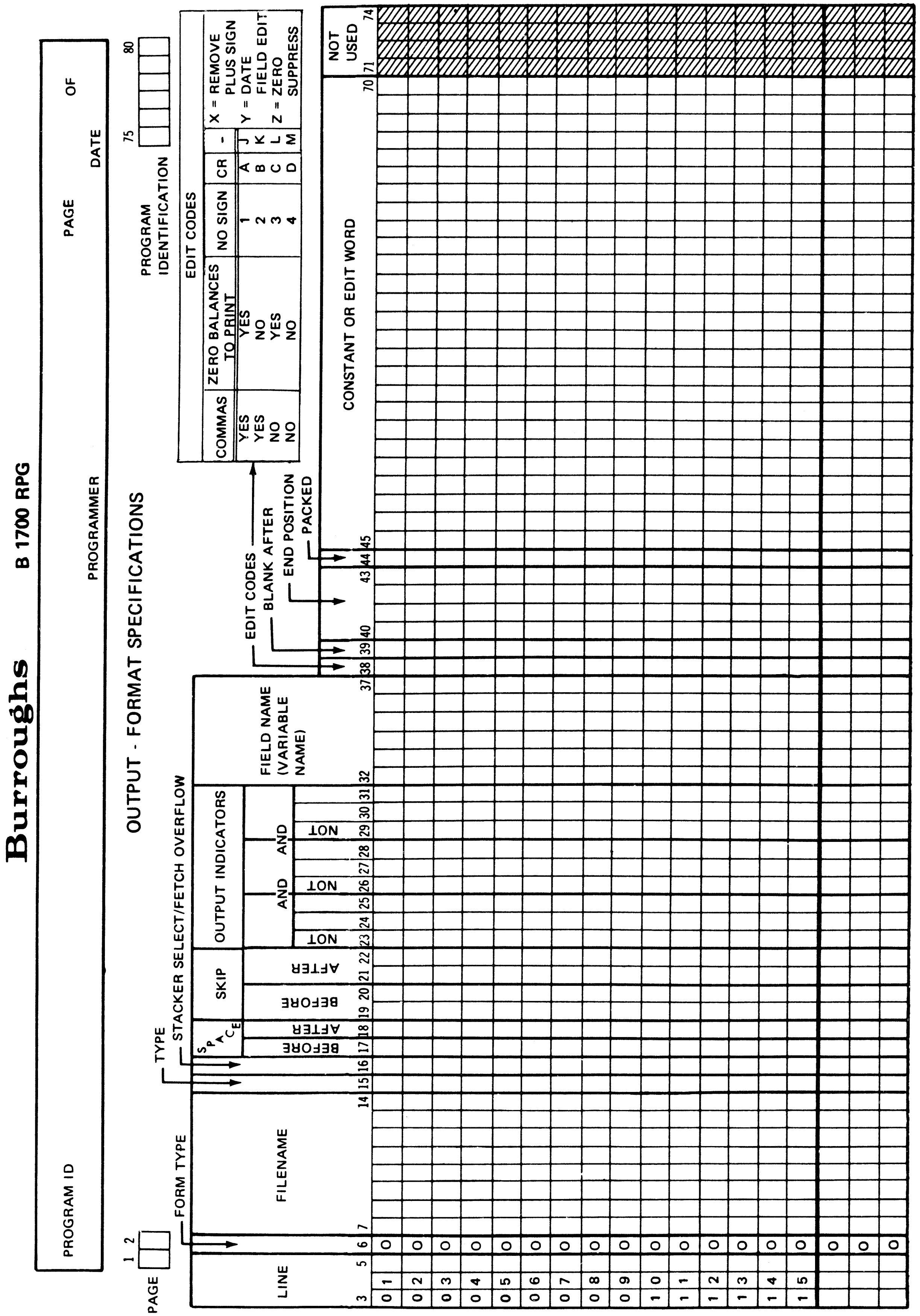
FILENAME (COLUMNS 7-14).
This field is used to identify the file to which the subsequent record type and field description entries belong. The file specified must have been previously described on the File Description Specifications form. Every output file described in the File Description Specifications must also be described on the Output-Format Specifications form.

TYPE (COLUMN 15).
This field is used to specify the type of output record to be written. These records contain such information as:

a. Heading records contain such information as page headings.

b. Detail records usually contain some type of data obtained directly from input records and calculation operations. Detail records are written once during every cycle, depending upon conditioning indicators.

c. Total records usually contain totals accumulated from a group



Burroughs B 1700 RPG

OUTPUT - FORMAT SPECIFICATIONS

PROGRAM ID

PROGRAMMER

PAGE OF

DATE

PAGE 1 2

PROGRAM IDENTIFICATION 75 80

EDIT CODES

COMMAS	ZERO BALANCES TO PRINT	NO SIGN	CR	-
YES	YES	1	A	J
YES	NO	2	B	K
NO	YES	3	C	L
NO	NO	4	D	M

X = REMOVE PLUS SIGN
Y = DATE FIELD EDIT
Z = ZERO SUPPRESS

LINE 3 5

FORM TYPE 6

FILENAME 7 14

TYPE 15

STACKER SELECT/FETCH OVERFLOW 16

SPACE BEFORE 17 AFTER 18

SKIP BEFORE 19 20 AFTER 21 22

OUTPUT INDICATORS NOT 23 24 25 AND NOT 26 27 28 AND NOT 29 30 31

FIELD NAME (VARIABLE NAME) 32 37

EDIT CODES 38

BLANK AFTER 39

END POSITION 40 43

PACKED 44

CONSTANT OR EDIT WORD 45 70

NOT USED 71 74

0 1	O	
0 2	O	
0 3	O	
0 4	O	
0 5	O	
0 6	O	
0 7	O	
0 8	O	
0 9	O	
1 0	O	
1 1	O	
1 2	O	
1 3	O	
1 4	O	
1 5	O	
	O	
	O	
	O	

Figure 9-1. Output-Format Specifications Form

of input records. Total records are written only during cycles in which a control break occurs, depending upon conditioning indicators.

d. Exception records are written during calculation time, through use of the EXCPT operation code.

Valid entries for this field are:

Entry	Definition
H	Heading records.
D	Detail records.
T	Total records.
E	Exception records (written during calculation time).

If an update file has total output specified, the record updated will be the last record read (i.e., the record that caused the Control Break). Exception records may not be specified for a combined file.

Specification lines should be written in one of the following ways:

a. Code the output records for each file in this order: heading, detail, total, and exception, or

b. Code all heading records for all files, then all detail records for all files, and so forth. This method is preferred, and generates more efficient code.

RECORD ADDITION (COLUMNS 16-18).
If a record is to be added to an input, output, or update file (disk files only), the word ADD must be entered in these columns.

STACKER SELECT/FETCH OVERFLOW (COLUMN 16).
This field may be used to specify:

a. The stacker into which the output card is to be placed after being punched, or

b. That the overflow routine can be invoked at this point for a printer file.

Valid entries for this field are:

Entry	Definition
Blank	Cards automatically go to default stacker.
numeric entry	Stacker into which card type is stacked.
F	Fetch overflow (printer files only).

STACKER SELECT. Only card output or combined files may be stacker selected. Output files may be stacker selected only in the Output-Format Specifications; combined files may be stacker selected either in the Input or Output-Format Specifications. If a combined file is stacker selected in both the Input and Output-Format Specifications, the Output-Format stacker specification overrides the stacker specified in the Input Specifications.

Stacker selection on the basis of matching records should only be specified for detail output lines, because the MR indicator is on only at detail output time to signal the matching status. Record types identified by OR lines may be stacker selected for a special stacker by an entry in this field; however, if the STACKER SELECT field entry is left blank, the card type selected by the OR line will go to the default stacker. AND lines may not have an entry in STACKER SELECT.

At execution time, any record types specifying a stacker number higher than that available on the device being used will go to the default stacker.

FETCH OVERFLOW. When the overflow line has been sensed on a printer file, the following actions take place:

a. All remaining detail lines in the current program cycle are printed.

b. All remaining total lines in the current program cycle are printed.

c. All lines conditioned by an overflow indicator are printed (total first, then detail).

d. Forms advance to the next page if a skip to top-of-page has been specified in one of the total or detail overflow lines.

If, however, it is desired to print overflow lines ahead of the usual time, a fetch overflow routine may be specified. This may be initiated any time after the overflow line has been reached. When overflow is caused in this manner, the following actions take place:

a. All total lines conditioned by the overflow indicator are printed.

b. Forms advance to the next page if a skip to top-of-page has been specified in a total line conditioned by an overflow indicator.

c. Heading lines conditioned by the overflow indicator are printed.

d. The line that fetched the overflow routine is printed.

e. Any detail and/or total lines left to be printed in the current program cycle are printed.

The overflow routine that is fetched does not automatically cause forms to advance. A skip to top-of-page must also be specified on a line conditioned by the overflow indicator.

Fetched overflow can be specified for any detail, total, or exception line, except those conditioned by an overflow indicator.

SPACE (COLUMNS 17-18).
This field is used to specify forms spacing for printer output files. It is divided into two subfields such that spacing BEFORE or AFTER printing may be specified. If both fields (SPACE and SKIP) are blank, single spacing after printing is assumed.

NOTE

> If any of the four subfields contains an entry, all desired skipping and spacing should be specified.

Valid entries for this field are:

Entry	Definition
Blank	Single space.
0	Space suppress.
1-9	Space 1-9 lines, as specified.

SKIP (COLUMNS 19-22).
This field is used to specify forms skipping for printer output files. It is divided into two subfields such that skipping BEFORE or AFTER printing may be specified. Valid entries for this field are:

Entry	Definition
0-99	Line number for skip - 0-99.
A0-A9	Line number for skip - 100-109.
B0-B2	Line number for skip - 110-112.

A skip entry must not be greater than the line number of the overflow line as specified in the Line Counter Specifications.

If both skipping and spacing are specified on the same line, the following order is used:

a. SKIP BEFORE.
b. SPACE BEFORE.
c. SKIP AFTER.
d. SPACE AFTER.

Spacing to or beyond the overflow line causes the overflow indicator to turn ON. However, skipping beyond the overflow line to a line on the next page does not cause the overflow indicator to turn ON.

Skipping and spacing AFTER printing is more efficient than skipping and spacing BEFORE printing.

Different SPACE and SKIP entries may be specified for OR lines; however, if these entries are blank for an OR line, spacing and skipping is done according to the specifications on the preceding line.

NOTE

> Refer to appendix G for the use of this field in IBM's RPG.

OUTPUT INDICATORS (COLUMNS 23-31).
This field is divided into three subfields such that up to three indicators on each line may be specified to condition an output operation. Each subfield is divided into two parts, as follows:

a. NOT (one column).
b. INDICATOR (two columns).

The NOT portion is used to specify that the associated indicator must be OFF in order for the operation to occur. If this condition is desired, an N must be entered in the NOT portion. Otherwise, the NOT portion must be left blank.

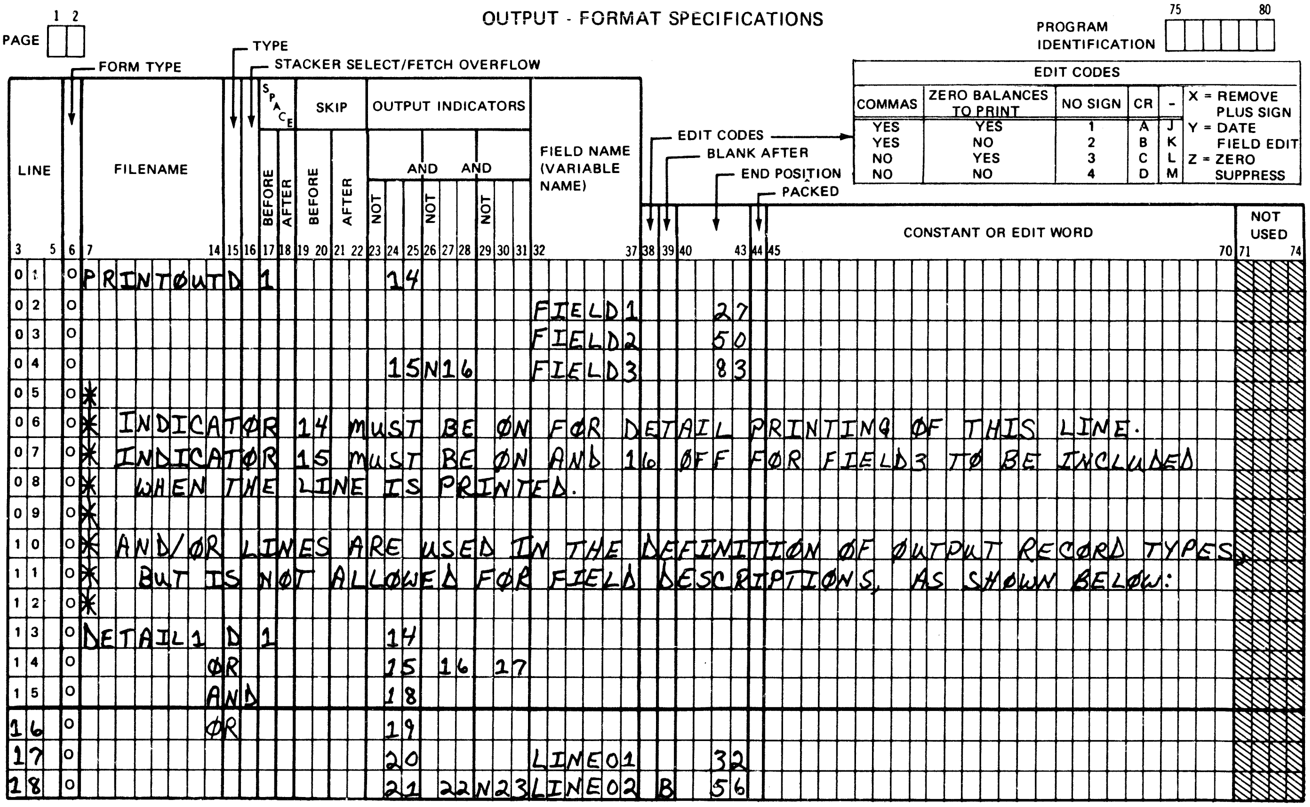
The INDICATOR portion is used to specify the indicator to be tested for ON (NOT = blank) or OFF (NOT = N). The following entries are allowed in this portion of the INDICATORS field:

Entry	Definition
Blank	Operation not conditioned by any indicator.
01-99	Operation conditioned by indicator used elsewhere in the program.
L0-L9	Operation condition by control level indicator previously assigned.
LR	Operation conditioned by last record indicator.
MR	Operation conditioned by matching record indicator.
H0-H9	Operation conditioned by halt indicator used elsewhere in the program.
U1-U8	Operation conditioned by external indicator previously set.

Entry	Definition
OA-OG, OV	Operation conditioned by overflow indicator previously set.
1P	Operation conditioned by first page indicator.

All three indicators on one line are in an AND relationship. All indicators on one line (or grouped lines) must be ON or OFF as specified in order for the associated operation to take place. At least one of the indicators specified for an output operation must be positive (i.e., NOT entry left blank).

An indicator specified on the line describing the record type will condition the entire output record. An indicator used to condition a field within the record is placed on the same line as the field description (see figure 9-2).


PAGE 1 2
PROGRAM IDENTIFICATION 75 80
FORM TYPE
TYPE
STACKER SELECT/FETCH OVERFLOW
LINE
FILENAME
SPACE BEFORE AFTER
SKIP BEFORE AFTER
OUTPUT INDICATORS AND AND NOT NOT NOT
FIELD NAME (VARIABLE NAME)
EDIT CODES
BLANK AFTER
END POSITION
PACKED
CONSTANT OR EDIT WORD
NOT USED
3 5 6 7 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 37 38 39 40 43 44 45 70 71 74

OUTPUT - FORMAT SPECIFICATIONS

EDIT CODES

COMMAS	ZERO BALANCES TO PRINT	NO SIGN	CR	-	
YES	YES	1	A	J	X = REMOVE PLUS SIGN
YES	NO	2	B	K	Y = DATE FIELD EDIT
NO	YES	3	C	L	Z = ZERO SUPPRESS
NO	NO	4	D	M	

```
01 OPRINTØUTD 1      14
02 O                         FIELD1    27
03 O                         FIELD2    50
04 O                 15N16   FIELD3    83
05 O*
06 O*  INDICATØR 14 MUST BE ØN FØR DETAIL PRINTING ØF THIS LINE.
07 O*  INDICATØR 15 MUST BE ØN AND 16 ØFF FØR FIELD3 TØ BE INCLUDED
08 O*    WHEN THE LINE IS PRINTED.
09 O*
10 O*  AND/ØR LINES ARE USED IN THE DEFINITTIØN ØF ØUTPUT RECØRD TYPES,
11 O*    BUT IS NØT ALLØWED FØR FIELD DESCRIPTIØNS, AS SHØWN BELØW:
12 O*
13 ODETAIL1 D 1      14
14 O       ØR        15 16 17
15 O       AND       18
16 O       ØR        19
17 O                 20      LINE01    32
18 O                 21 22N23LINE02 B  56
```

Figure 9-2. Output Indicators

NUMERIC INDICATORS (01-99). The numeric indicators (01-99) may be used as follows:

a. To condition output operations that are only to be performed for specific input record types (record identifying indicators).

b. To condition output operations that are to be done when an input field meets certain conditions (field indicators), or

c. To condition output operations according to the results of previous operations in the Calculation Specifications (resulting indicators).

CONTROL LEVEL INDICATORS (L0-L9). Control level indicators may be used to condition output operations that are to be performed only on the first record of a new Control Group.

The L0 indicator remains ON during the entire program. If no other control level indicators are assigned, but it is desired to perform total calculation and total output operations, the L0 indicator may be used to condition those operations.

LAST RECORD INDICATOR (LR). The last record indicator is used to condition output operations that are to be performed at End-of-Job.

MATCHING RECORDS INDICATOR (MR). The matching records indicator is used to condition output operations that are to be performed only when matching input records are found.

HALT INDICATORS (H0-H9). Halt indicators previously assigned in the FIELD INDICATORS field (Input Specifications) or the RESULTING INDICATORS field (Calculation Specifications) may be used to condition output operations that are to be performed only when an error condition occurs.

Since the program does not halt until *after* the record in error has been completely processed, some operations must be prevented in order to avoid erroneous output. By using halt indicators in conjunction with a N in the NOT portion of the INDICATORS field, an operation may be inhibited when the specified halt indicator is ON.

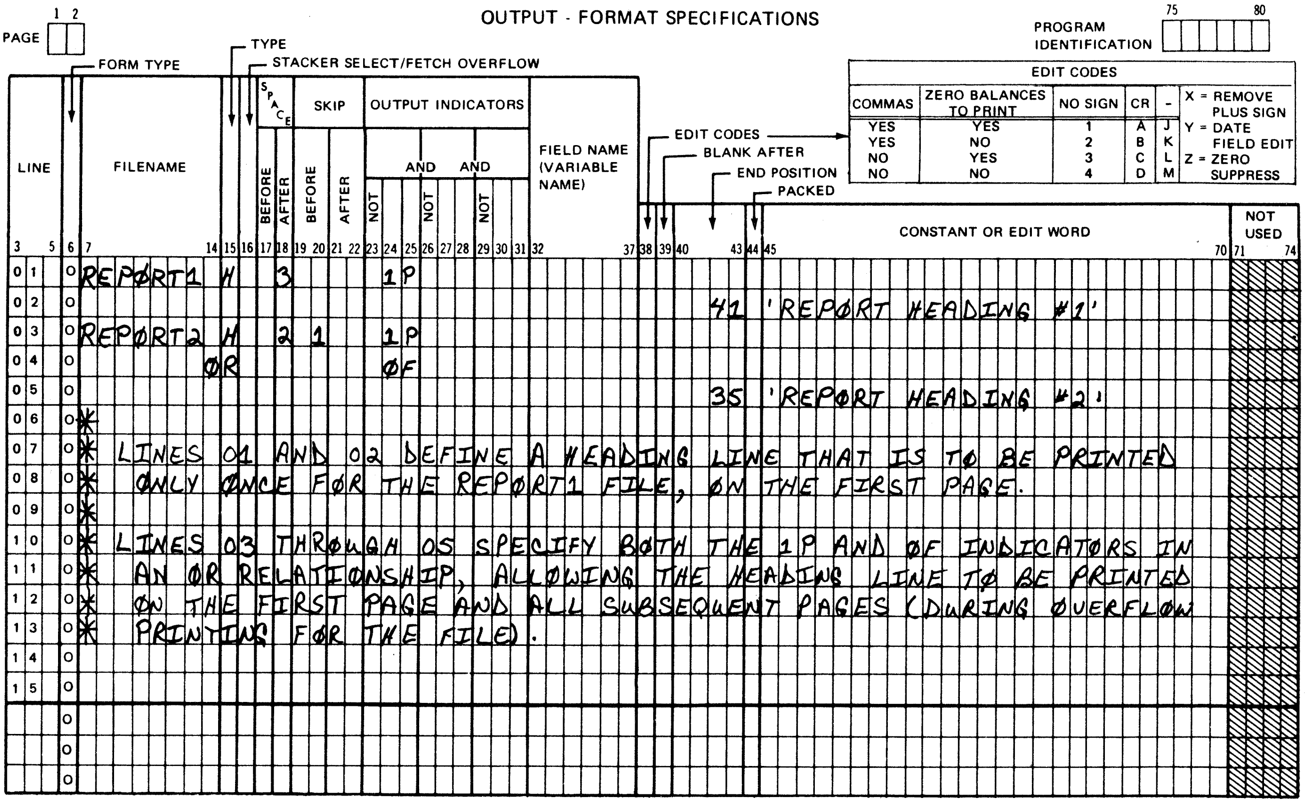
EXTERNAL INDICATORS (U1-U8). If an output file specified conditioning by an external indicator in the File Description Specifications (EXTERNAL INDICATORS field), every output record described for the file must be conditioned by the same external indicator.

OVERFLOW INDICATORS (OA-OG, OV). Overflow indicators previously assigned in the OVERFLOW INDICATOR field (File Description Specifications) may be used to condition output operations that are to be performed when the overflow line on a printer file has been reached.

Overflow indicators which have not been previously assigned in the File Description Specifications (except OF and OV, which are compiler-defined) may not be used in the Output-Format Specifications. Forms advancing at End-of-Page are handled automatically if no overflow indicators are assigned to the file. Any specification line not conditioned by an overflow indicator which designates a skip to the next page turns OFF all overflow indicators before the skip takes place.

No more than one overflow indicator may be associated with a group of output indicators in an AND or OR relationship, and it must be the same indicator assigned to the file in the File Description Specifications.

FIRST PAGE INDICATOR (1P). The first page indicator is used to allow printing of heading information on the first page of a printer output file. In conjunction with an overflow indicator in an OR relationship, it may be used to allow printing of the same heading information on all pages. The use of the 1P indicator is illustrated below (see figure 9-3).



OUTPUT - FORMAT SPECIFICATIONS

COMMAS	ZERO BALANCES TO PRINT	NO SIGN	CR	-	
YES	YES	1	A	J	X = REMOVE PLUS SIGN
YES	NO	2	B	K	Y = DATE FIELD EDIT
NO	YES	3	C	L	Z = ZERO SUPPRESS
NO	NO	4	D	M	

```
01 OREPØRT1 H  3     1P
02 O                                   41 'REPØRT HEADING #1'
03 OREPØRT2 H  2 1   1P
04 O       ØR        ØF
05 O                                   35 'REPØRT HEADING #2'
06 O*
07 O* LINES 01 AND 02 DEFINE A HEADING LINE THAT IS TØ BE PRINTED
08 O*   ØNLY ØNCE FØR THE REPØRT1 FILE, ØN THE FIRST PAGE.
09 O*
10 O* LINES 03 THRØUGH 05 SPECIFY BØTH THE 1P AND ØF INDICATØRS IN
11 O*   AN ØR RELATIØNSHIP, ALLØWING THE HEADING LINE TØ BE PRINTED
12 O*   ØN THE FIRST PAGE AND ALL SUBSEQUENT PAGES (DURING ØVERFLØW
13 O*   PRINTING FØR THE FILE).
14 O
15 O
```

Figure 9-3. 1P Indicator

The first page indicator can be used only to condition heading or detail lines, and cannot be specified for total or exception lines.

AND/OR LINES. If it is necessary to specify more than three indicators to condition an output operation, an AND line may be used. The word AND must be entered in columns 14-16, and the additional indicators entered in their respective fields. The conditions specified for all indicators in an AND relationship must be met before the associated output operation will take place.

OR lines (OR in columns 14-15) may be used to group indicators such that only one of the conditions specified must be met for the associated output operation to take place. Both AND and OR lines may be used together to condition an output record (but not a field). A maximum of three indicators in an AND relationship (on one line) may be used to condition a field. See Figure 9-2 for an example of the usage of AND and OR lines.

There is no limit to the number of AND or OR lines which may be specified.

VARIABLE NAME (COLUMNS 32-37).
This field is used to assign an identifier (name) to an output data field. The identifier used must have previously been defined in the Input Specifications (VARIABLE NAME field), the Extension Specifications (VECTOR NAME field), or the Calculation Specifications (RESULT FIELD field). Also, certain special names may be used. A separate line must be used for each field description. Fields may be listed in any order within each record type, since their location is determined by the entry in the END POSITION field. If fields overlap, only the last field specified appears in the output record.

SPECIAL WORDS. The following special words are reserved for use as variable names:

PAGE
PAGE1
PAGE2
UDATE
UMONTH
UDAY
UYEAR
*PLACE
*PRINT

Each special word has a specifically defined usage, as described below:

Page Fields (PAGE, PAGE1, and PAGE2).

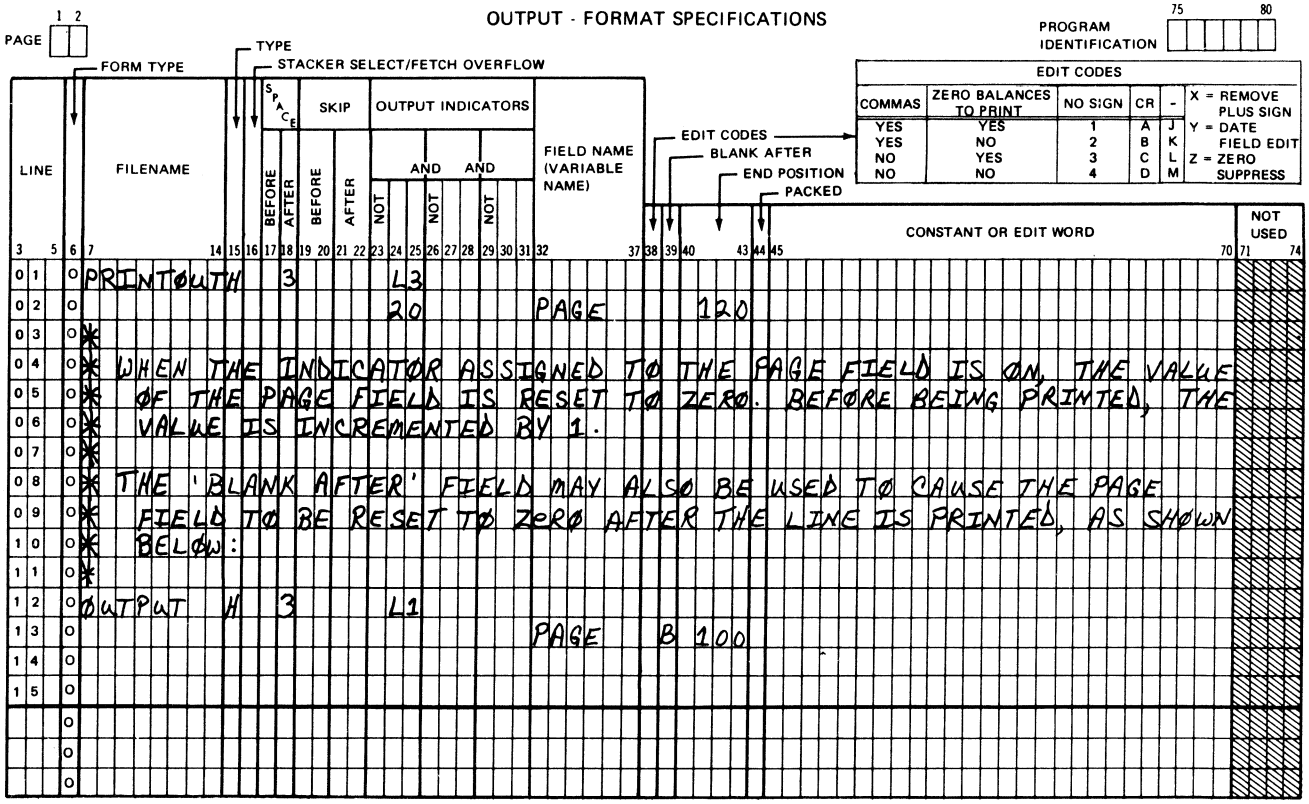
If page numbering is to be done on output, the special word PAGE (or PAGE1, PAGE2 for two more printer files) is used to indicate that page numbering is to be done automatically. When a PAGE field is named in this field without being previously defined in the Input or Calculation Specifications, it is assumed to be four characters in length with no decimal positions. On output, leading zeros are suppressed and the sign (if negative) is printed in the rightmost position unless an edit word or edit code is specified.

The page number begins at one (unless otherwise specified), and is automatically incremented by one for each new page.

The page number may be reset at any point during the program by setting the PAGE field to zero before it is printed (see Figure 9-4). This may be accomplished in two ways:

a. Use the BLANK AFTER specification to cause the field to be cleared to zero after printing, or

b. Assign an OUTPUT INDICATOR to the PAGE field. If the indicator is ON, the field will be set to zero before normal incrementation takes place.



OUTPUT - FORMAT SPECIFICATIONS

PAGE 1 2 — PROGRAM IDENTIFICATION 75 80

FORM TYPE — TYPE — STACKER SELECT/FETCH OVERFLOW — SPACE (BEFORE, AFTER) — SKIP (BEFORE, AFTER) — OUTPUT INDICATORS (NOT, AND NOT, AND NOT) — FIELD NAME (VARIABLE NAME) — EDIT CODES — BLANK AFTER — END POSITION — PACKED — CONSTANT OR EDIT WORD — NOT USED

EDIT CODES

COMMAS	ZERO BALANCES TO PRINT	NO SIGN	CR	-	
YES	YES	1	A	J	X = REMOVE PLUS SIGN
YES	NO	2	B	K	Y = DATE FIELD EDIT
NO	YES	3	C	L	Z = ZERO SUPPRESS
NO	NO	4	D	M	

```
LINE 6 7 FILENAME 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 FIELD NAME 37 38 39 40 43 44 45 CONSTANT OR EDIT WORD 70 71 NOT USED 74

01 OPRINTØUTH  3     L3
02 O                 20      PAGE     120
03 O*
04 O* WHEN THE INDICATOR ASSIGNED TØ THE PAGE FIELD IS ØN, THE VALUE
05 O*  ØF THE PAGE FIELD IS RESET TØ ZERØ. BEFØRE BEING PRINTED, THE
06 O*  VALUE IS INCREMENTED BY 1.
07 O*
08 O* THE 'BLANK AFTER' FIELD MAY ALSØ BE USED TØ CAUSE THE PAGE
09 O*  FIELD TØ BE RESET TØ ZERØ AFTER THE LINE IS PRINTED, AS SHØWN
10 O*  BELØW:
11 O*
12 OØUTPUT  H  3     L1
13 O                         PAGE   B 100
14 O
15 O
   O
   O
   O
```

Figure 9-4. PAGE Specification

The same PAGE entry (PAGE, PAGE1, or PAGE2) must not be used for two different output files.

Date Fields (UDATE, UMONTH, UDAY, and UYEAR).

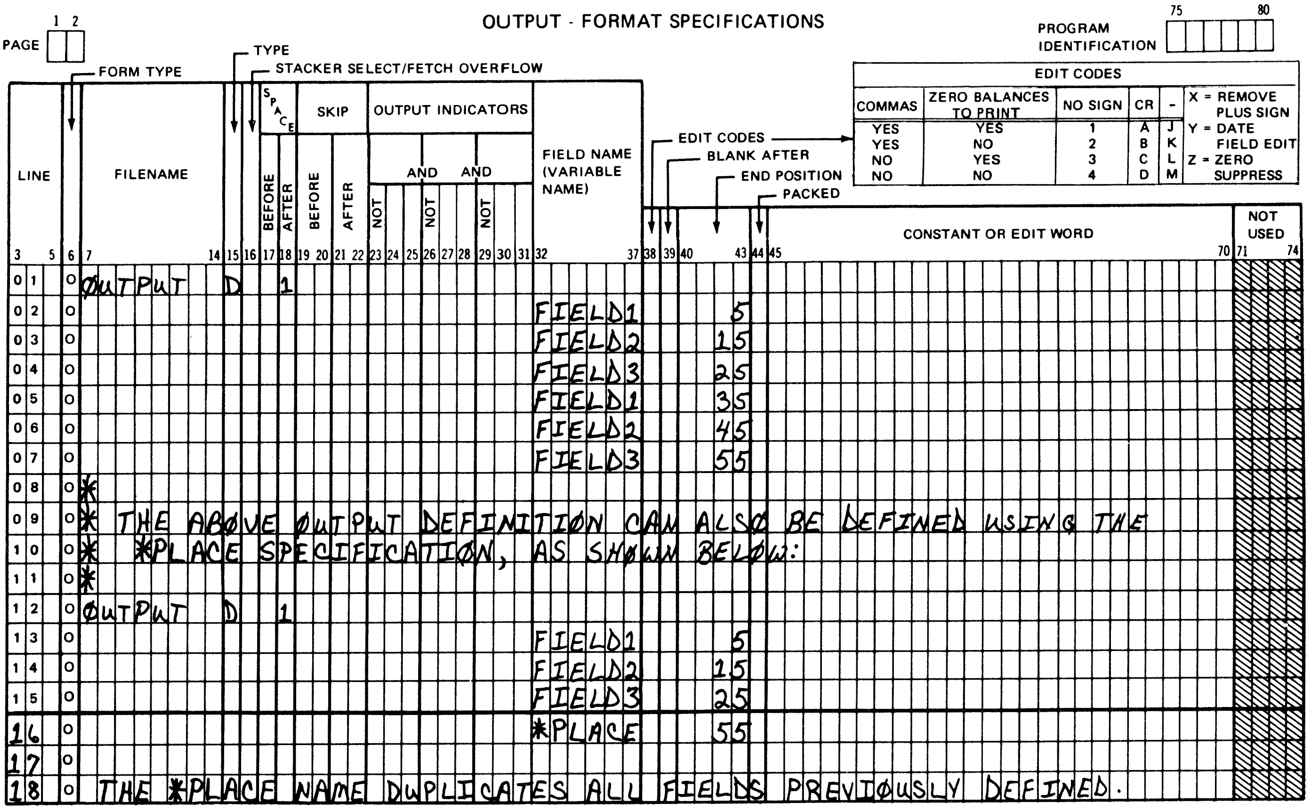
Four special words allow the program to obtain the current value of the date as maintained by the MCP. Refer to Section 7 - VARIABLE NAME field for rules applying to date fields.

*PLACE Specification.

The special word *PLACE allows writing of the same field or fields in more than one place in an output record without having to specify the field names and end positions more than once. The designated fields are written in the same relative positions ending in the position specified for the *PLACE entry. It is possible to obtain the same results in two ways (see Figure 9-5):

a. Define each field and its corresponding end position for every time it is to appear in the output records, or

b. Use the special word *PLACE.



OUTPUT - FORMAT SPECIFICATIONS

PAGE 1 2 — PROGRAM IDENTIFICATION 75 80

LINE — FORM TYPE — FILENAME — TYPE — STACKER SELECT/FETCH OVERFLOW — SPACE (BEFORE, AFTER) — SKIP (BEFORE, AFTER) — OUTPUT INDICATORS (NOT, AND NOT, AND NOT) — FIELD NAME (VARIABLE NAME) — EDIT CODES — BLANK AFTER — END POSITION — PACKED — CONSTANT OR EDIT WORD — NOT USED

COMMAS	ZERO BALANCES TO PRINT	NO SIGN	CR	-	
YES	YES	1	A	J	X = REMOVE PLUS SIGN
YES	NO	2	B	K	Y = DATE FIELD EDIT
NO	YES	3	C	L	Z = ZERO SUPPRESS
NO	NO	4	D	M	

```
01 O ØUTPUT    D  1
02 O                          FIELD1      5
03 O                          FIELD2     15
04 O                          FIELD3     25
05 O                          FIELD1     35
06 O                          FIELD2     45
07 O                          FIELD3     55
08 O*
09 O* THE ABØVE ØUTPUT DEFINITIØN CAN ALSØ BE DEFINED USING THE
10 O*   *PLACE SPECIFICATIØN, AS SHØWN BELØW:
11 O*
12 O ØUTPUT    D  1
13 O                          FIELD1      5
14 O                          FIELD2     15
15 O                          FIELD3     25
16 O                          *PLACE     55
17 O
18 O  THE *PLACE NAME DUPLICATES ALL FIELDS PREVIØUSLY DEFINED.
```

Figure 9-5. *PLACE Specification

Both methods will produce identical results, but use of the *PLACE entry saves extra coding.

The following rules must be observed when using the *PLACE specification:

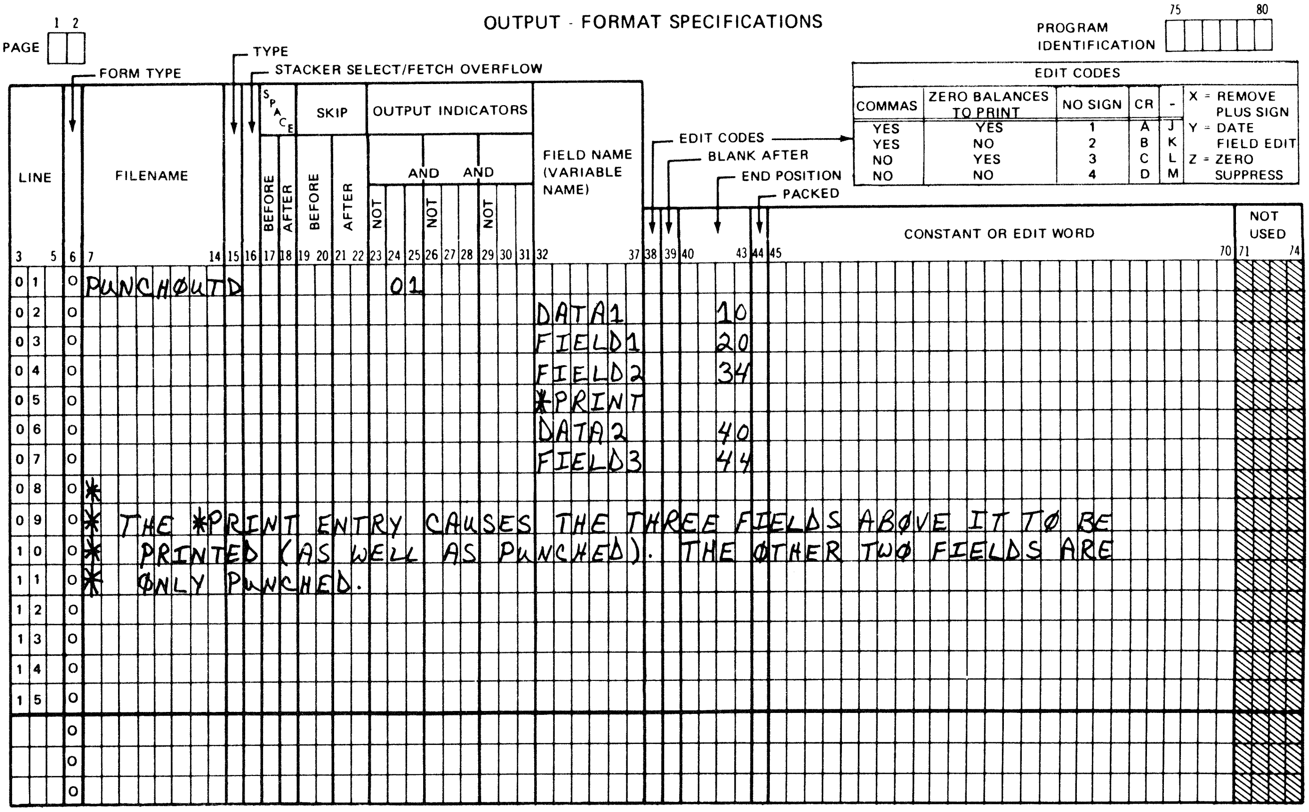
a. All fields within the record type written above the *PLACE entry are repeated according to the *PLACE' specification, not just the one line above.

b. An end position must be given for every *PLACE specification.

c. An additional *PLACE entry (on a separate line) must be used every time the fields are to be repeated.

*PRINT Specification.
The special word *PRINT is used to cause card interpreting after punching (for those peripheral devices so equipped). Printing is done at the top of the cards in the same column positions as the fields are punched. The *PRINT specification may be used only once for each rec-

ord and must appear after all fields on the card which are to be printed. The *PRINT specification may be conditioned by indicators in the OUTPUT INDICATORS field; all other fields must be left blank (see figure 9-6).



OUTPUT - FORMAT SPECIFICATIONS

PAGE □□ PROGRAM IDENTIFICATION □□□□□□

EDIT CODES

COMMAS	ZERO BALANCES TO PRINT	NO SIGN	CR	-	
YES	YES	1	A	J	X = REMOVE PLUS SIGN
YES	NO	2	B	K	Y = DATE FIELD EDIT
NO	YES	3	C	L	Z = ZERO SUPPRESS
NO	NO	4	D	M	

LINE (3-5)	FORM TYPE (6)	FILENAME (7-14)	OUTPUT INDICATORS (23-31)	FIELD NAME (32-37)	END POSITION (40-43)
01	O	PUNCHØUTD	01		
02	O			DATA1	10
03	O			FIELD1	20
04	O			FIELD2	34
05	O			*PRINT	
06	O			DATA2	40
07	O			FIELD3	44
08	O	*			
09	O	* THE *PRINT ENTRY CAUSES THE THREE FIELDS ABØVE IT TØ BE			
10	O	* PRINTED (AS WELL AS PUNCHED). THE ØTHER TWØ FIELDS ARE			
11	O	* ØNLY PUNCHED.			
12	O				
13	O				
14	O				
15	O				

Figure 9-6. *PRINT Specification

EDIT CODES (COLUMN 38).
This field is used to specify editing of a numeric output field. Certain editing operations have been provided, for which it is not necessary to write an edit word. These operations (and their corresponding edit codes) are summarized in the EDIT CODES table on the Output-Format Specifications sheet.

Only numeric fields may be edited, and if an edit code is specified, the CONSTANT OR EDIT WORD field must be left blank; however, if check protect or floating dollar sign is required, enter '*' or '$', respectively, in columns 45-47.

BLANK AFTER (COLUMN 39).
This field is used to specify that a field is to be reset after the output operation is finished. Alphanumeric fields will be cleared to blanks and numeric fields will be cleared to zeros. Valid entries for this field are:

Entry	Definition
Blank	Field is not to be cleared.
B	Field is to be cleared after the output operation is complete.

END POSITION (COLUMNS 40-43).
This field is used to specify the location of a field within an output record. Only the location of the rightmost position is specified; the number of positions to the left of that location is determined by the length specified previously for the field. Space must also be left to allow for any editing symbols (edit codes or edit words) that may be entered.

To print fields on a card in a position other than the one assigned by using the *PRINT specification, code an asterisk in column 40. The *PRINT specification interprets the cards exactly as punched; however, this is not always desirable. To print the fields in positions other than would be assigned by the *PRINT specification:

a. Enter the field name to be printed in the VARIABLE NAME field, and

b. Enter an asterisk in column 40, and

c. Enter the end position for the field in columns 41-43 (limited to a maximum of 96).

All lines with an asterisk in column 40 should follow all lines specifying punching only and all *PRINT lines for each record.

PACKED (COLUMN 44).
This field is used to specify that an output field will be written in packed decimal format. Packed decimal fields cannot be printed. Valid entries for this field are:

Entry	Definition
Blank	Field is to be written in alphanumeric or unpacked decimal format.
P	Field is to be written in packed decimal format.

This field must be blank if an asterisk appears in column 40 of the same specification line or for fields that precede *PRINT for a card file or *PLACE for a printer file.

CONSTANT OR EDIT WORD (COLUMNS 45-70).
Constants or Edit Words are entered in this field. A description of each is provided in the text that follows.

CONSTANTS. Constants are literals usually used for such things as page headings, and contain information that is not changed by any operations. Constants will appear exactly as written in the output record, in the position specified. The following rules must be observed when forming constants:

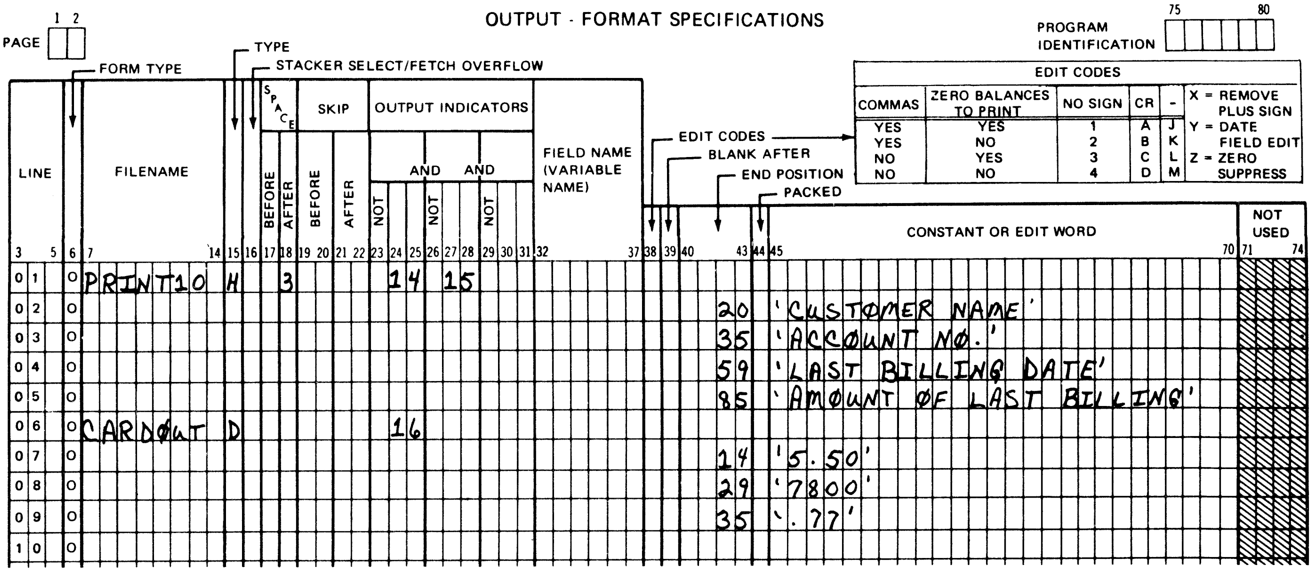
a. The VARIABLE NAME entry must be left blank.

b. Constants must be left-justified, enclosed in apostrophes, and no more than 24 characters in length.

c. If an apostrophe is to appear in the constant, two consecutive apostrophes must be coded.

d. Numeric data may be used as a constant, but must still be enclosed in apostrophes.

Some samples of output constants are illustrated below (see figure 9-7).



OUTPUT - FORMAT SPECIFICATIONS

PAGE □□ (1 2)

PROGRAM IDENTIFICATION □□□□□□ (75 80)

EDIT CODES					
COMMAS	ZERO BALANCES TO PRINT	NO SIGN	CR	-	X = REMOVE PLUS SIGN Y = DATE FIELD EDIT Z = ZERO SUPPRESS
YES	YES	1	A	J	
YES	NO	2	B	K	
NO	YES	3	C	L	
NO	NO	4	D	M	

LINE (3-5)	FORM TYPE (6)	FILENAME (7-14)	TYPE (15)	STACKER SELECT/FETCH OVERFLOW (16)	SPACE BEFORE (17)	SPACE AFTER (18)	SKIP BEFORE (19-20)	SKIP AFTER (21-22)	OUTPUT INDICATORS (23-25)	AND (26-28)	AND (29-31)	FIELD NAME (VARIABLE NAME) (32-37)	EDIT CODES (38)	BLANK AFTER (39)	END POSITION (40-43)	PACKED (44)	CONSTANT OR EDIT WORD (45-70)	NOT USED (71-74)
01	O	PRINT10	H			3			14	15								
02	O														20		'CUSTØMER NAME'	
03	O														35		'ACCØUNT NØ.'	
04	O														59		'LAST BILLING DATE'	
05	O														85		'AMØUNT ØF LAST BILLING'	
06	O	CARDØUT	D						16									
07	O														14		'5.50'	
08	O														29		'7800'	
09	O														35		'.77'	
10	O																	

Figure 9-7. Output Constants

EDIT WORDS. Edit words are used instead of an edit code (see column 38 - EDIT CODE) when a special edit format is desired. The following rules must be observed when forming edit words:

a. The VARIABLE NAME entry must not be blank.

b. Edit words must be left-justified, enclosed in apostrophes, and no more than 24 characters in length.

c. Any character in the RPG character set (refer to section 2) is valid, but certain characters have special uses.

d. The number of replaceable characters (0, *, blank, $) in the edit word must not be less than the length of the field to be edited.

e. Column 38 (EDIT CODE) must be left blank.

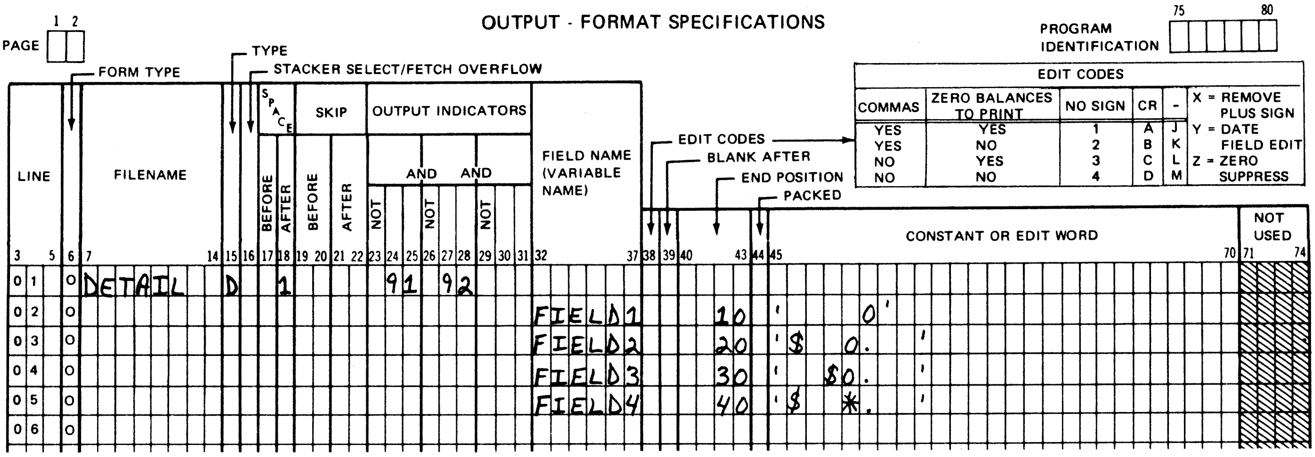
A replaceable character is a part of the edit word that does not require an extra position in the output field. The replaceable entries are as follows:

Entry	Definition
Blank 0	Zero suppression.
*	Check protect.
$	Floating dollar sign only.

Fixed dollar sign, decimal points, commas, ampersands, negative signs, and constant information all require extra space in the output field.

NOTE
One extra space in the edit word must be provided for the floating dollar sign, or no dollar sign will appear if the output field is full.

Some samples of edit words are illustrated below (see figure 9-8).



OUTPUT - FORMAT SPECIFICATIONS

EDIT CODES					
COMMAS	ZERO BALANCES TO PRINT	NO SIGN	CR	-	X = REMOVE PLUS SIGN Y = DATE FIELD EDIT Z = ZERO SUPPRESS
YES	YES	1	A	J	
YES	NO	2	B	K	
NO	YES	3	C	L	
NO	NO	4	D	M	

```
  01 ODETAIL  D  1     91 92
  02 O                         FIELD1    10 '    0'
  03 O                         FIELD2    20 '$  0.  '
  04 O                         FIELD3    30 '  $0.  '
  05 O                         FIELD4    40 '$  *.  '
  06 O
```

PAGE
1 2
PROGRAM IDENTIFICATION
75 80
FORM TYPE
TYPE
STACKER SELECT/FETCH OVERFLOW
SPACE
SKIP
OUTPUT INDICATORS
LINE
FILENAME
BEFORE
AFTER
BEFORE
AFTER
AND
AND
NOT
NOT
NOT
FIELD NAME (VARIABLE NAME)
EDIT CODES
BLANK AFTER
END POSITION
PACKED
CONSTANT OR EDIT WORD
NOT USED
3 5 6 7 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 37 38 39 40 43 44 45 70 71 74

Figure 9-8. Edit Words

Table 9-1 provides the field values and edited values for the FIELD NAMES in figure 9-8.

Table 9-1

Field Values and Edited Values
for FIELD NAMES in Figure 9-8

Field Name	Field Value	Edited Field
FIELD1	00011	bbb11
FIELD2	00124	$bb1.24
FIELD3	01439	b$14.39
FIELD4	00974	$**9.74

NOT USED (COLUMNS 71-74).
This field is ignored by the compiler and should be left blank.

PROGRAM IDENTIFICATION (COLUMNS 75-80).
Refer to section 2 for a complete description.

SECTION 10

DOLLAR CARD SPECIFICATIONS

GENERAL.
Dollar Card Specifications allow the RPG Compiler or Translator to accommodate various extensions to other manufacturers RPG and RPG II languages, which cannot be handled on the other specification forms. Dollar Cards also allow certain compiler-control options to be set or reset during compilation.

Dollar Cards may appear anywhere within the source deck, as required. The standard Dollar Card Specifications form is illustrated below (see figure 10-1).

FIELD DEFINITIONS.
The fields for the Dollar Card Specifications form are described in the text that follows.

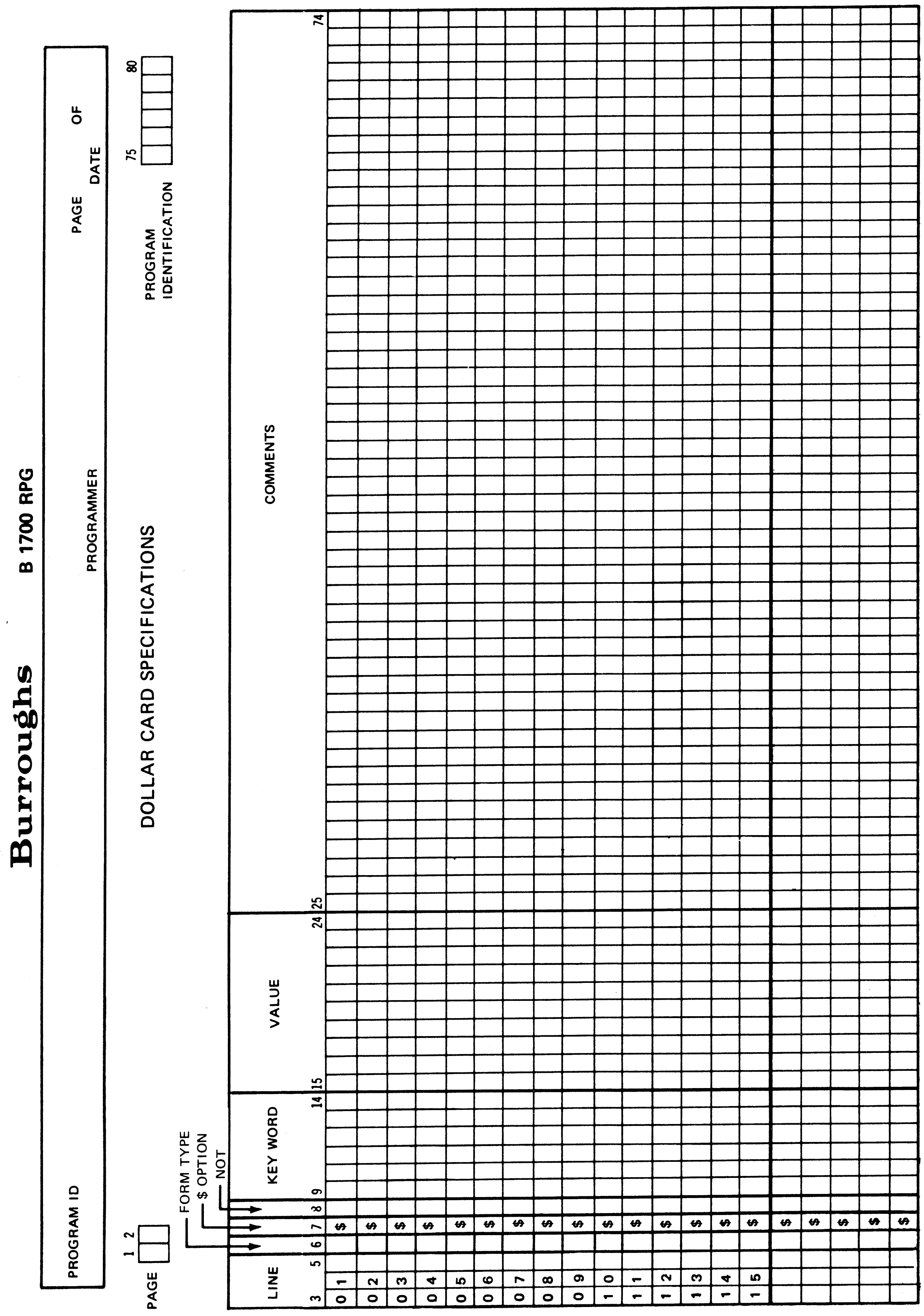
PAGE (COLUMNS 1-2).
Refer to section 2 for a complete description.

LINE (COLUMNS 3-5).
Refer to section 2 for a complete description.

FORM TYPE (COLUMN 6).
This field may be left blank or contain the form type to align with the associated form that the $ option was inserted in.

$ OPTION (COLUMN 7).
A $ must appear in this field.

NOT (COLUMN 8).
This field is used to specify that the option entered in the KEY WORD field is set ON (NOT = blank) or OFF (NOT = N). Certain options cannot be turned OFF; these are indicated under the KEY WORD entry description. Valid entries for this field are:



Burroughs

B 1700 RPG

PROGRAM ID

PROGRAMMER

PAGE OF

DATE

DOLLAR CARD SPECIFICATIONS

PAGE 1 2 □□

PROGRAM IDENTIFICATION 75 □□□□□□ 80

LINE 3 5	FORM TYPE 6	$ OPTION 7	NOT 8	KEY WORD 9 14	VALUE 15 24	COMMENTS 25 74
01		$				
02		$				
03		$				
04		$				
05		$				
06		$				
07		$				
08		$				
09		$				
10		$				
11		$				
12		$				
13		$				
14		$				
15		$				
		$				
		$				
		$				
		$				
		$				

Figure 10-1. Dollar Card Specifications

Entry	Definition
Blank	Specified option is "set".
N	Specified option is "reset".

KEY WORD (COLUMNS 9-14).
This field is used to name the option to be set or reset (according to entry in NOT field). The option name must be entered left-justified in the field. Options fall into two categories: RPG extensions, and comiler-directing options.

B 1700 RPG EXTENSIONS. The following options may appear only within the file description specifications, and must immediately precede the specification line describing the file to which they apply.

> NOTE
> None of the following operations may be "reset".

a. PACKID - specifies the pack name of a disk file. Similar to $ FAMILY and $ FILEID. Default of blank name and MCP assumes systems pack. This entry should be included to insure correct handling of Files by MCP.

b. FAMILY - specifies the external family name (MFID) associated with the file. The VALUE field contains the name (one to 10 characters left-justified).

c. FILEID - specifies the external file identification (FID) associated with the file. The VALUE field contains the name (one to 10 characters, left-justified).

> NOTE
> The internal name of each file is the 7-character filename assigned in the File Description Specifications. In the absence of FAMILY or FILEID specifications, the internal name will become the family name and with the addition of the File ID the external name will become the internal name/FILEID.

d. AREAS - specifies the maximum number of areas to be allocated for the file (disk files only). The VALUE field contains an integral value (1 to 40, right-justified, leading zeros optional). The default value assigned is 40, unless specified otherwise.

e. RPERA - specifies the maximum number of logical records that will be written in each disk area. The VALUE field contains an integral value (right-justified, leading zeros optional). The default value assigned is 500 unless specified otherwise.

f. OPEN - explicit open allows for all files to be opened at BOJ. Default is an implicit open (when files are actually called for).

g. CLOSE - explicit close allows all input serial files to remain opened until End-of-Job. Default is the Implicit Close (files close at End-of-File).

h. AAOPEN - is a file time option used to set a bit in the MCP File Parameter Block and allocate all disk space areas at the beginning of the program.

i. ONEPAK - specifies that this particular file must be contained on one Disk.

j. CYL - allocates file areas starting on an integral cylinder boundary.

k. DRIVE - allocates a physical drive to that particular file. Not related to PACKID. Value field (CC24) must be 0-15. Option may not be reset.

l. REFORM - input and update disk files are assumed to have the block and record length declared on the file header unless the $ REFORM option is used. However, on input or update chained indexed file specifications "data keys in core" option, it may be desirable to also use $ REFORM to indicate to the compiler that it may juggle the blocking factor to optimize the speed of chaining. Under this condition, the blocking-record-length specified on the File Description Specifications must be the same as when the file was outputted. This combination will produce the fastest chaining possible.

COMPILER-DIRECTING OPTIONS. The following options may appear anywhere within the B 1700 RPG Source Program, and direct the compiler to perform specific functions. All of the following options (with the exception of SEG) may be "reset".

a. LIST - specifies that the compiler produce a single-spaced listing of source input, together with applicable error or warning messages. This option is "set" by default; resetting it will not suppress error or warning messages. The listing triple spaces each time form type (CC 6) changes and adds a single space before error or warning messages.

b. LOGIC - specifies that the compiler produce a single-spaced listing of each source specification line followed immediately by an intermediate code used to generate COBOL-S-Code. The listing is produced after the NAMES listing (if the NAMES option is set), and does not include addresses or bit configurations, but only the opcodes and logical operands of the program.

c. MAP - specifies that the compiler produce a single-spaced listing detailing the program's memory utilization. The MAP listing is produced after the LOGIC listing (if the LOGIC option is set).

d. NAMES - specifies that the compiler is to produce a single-spaced listing of all assigned indicators, filenames, and field names. The attributes associated with each file and field are also listed. The NAMES listing is produced immediately after the normal source input listing.

e. RSIGN - indicates to the compiler, the location of the sign in numeric data items. When set, all signs are assumed to be in the low-order (rightmost) position of the field; when reset, all signs are assumed to be in the high-order (leftmost) position of the field. This option may be set and reset at different points in the Input and Output-Format Specifications, allowing different fields to have different sign positions. If this option is used, it will override the SIGN POSITION specified in the Control Card Specifications.

f. NRSIGN - resets RSIGN option.

g. SEG - orders the compiler to begin placing code in an overlayable segment, identified by the integer in the VALUE field (right-justified, between 0 and 7, inclusive). Segmentation is done automatically by the compiler, so that this option is used only if the user wishes to vary the compiler-determined segmentation. Up to eight overlayable segments may be generated.

h. SUPR - specifies that the compiler is to suppress all warning messages (not error messages).

i. XMAP - specifies that the compiler print a single-spaced listing of all the code generated, complete with actual bit configurations and addresses. Combined with the listing produced by the LOGIC option, complete information about the generated code of the program is available. The XMAP listing is produced after the MAP listing (if the MAP option is set).

VALUE (COLUMNS 15-24).
This field is used to specify a value to be associated with the option entered in the KEY WORD field. Not all options require a value; those that do are so designated in the individual descriptions of each option (KEY WORD field).

All alphanumeric values must be entered left-justified in the VALUE field. All numeric values must be entered right-justified in the VALUE field; leading zeros are optional.

COMMENTS (COLUMNS 25-74).
This field is available for inclusion of comments and documentary remarks, and may contain any valid EBCDIC characters.

PROGRAM IDENTIFICATION (COLUMNS 75-80).
Refer to section 2 for a complete description.

SECTION 11

COMPILER OPERATION

GENERAL.

The Burroughs RPG Compiler is an integral part of the small systems software system. It is treated in many respects as merely another program to be executed and can be multiprogrammed with other programs, or with itself.

SOURCE INPUT.

An input card file labeled "RPG/CARD" is the only source required by the compiler. All source input must be punched in the EBCDIC character set. The source program may be read from a device other than cards, through use of label equation (the FILE Card).

CONTROL CARD SYNTAX.

The format of the compiler control cards is as follows:

```
                                   ┌     ( SYNTAX  ) ┐
    ? COMPILE program-name WITH RPG│ FOR { LIBRARY } │
                                   └     ( SAVE    ) ┘

    ? CHARGE charge-number

    ? MEMORY memory-size

    ? PRIORITY priority-number

    ? [RPG] FILE internal-file-name ≡ external-file-name file-
      attributes

    ? program-name FILE internal-file-name ≡ external-file-name
      file-attributes

    ? DATA RPG/CARD

      source specification cards

    ? END

   ┌ ?(DATA  )RPG/VECTOR ┐
   └  (DATAB )            ┘

     [compile-time table cards]

   [? END]
```

All of the above control cards are fully discussed in the Software Operational Guide.

> NOTE
>
> If the input file has been label-equated to a device other than a card reader, the DATA Card and the source specifica-

tion cards will not be part of the
call-out deck.

The following sample deck should be used to compile a source program contained in punched cards to library:

```
? COMPILE PROGRAM/TEST1 WITH RPG LIBRARY
? CHARGE 12345
? DATA RPG/CARD
  source specification cards
? END
```

The following sample deck could be used to compile a source program contained on magnetic tape for syntax:

```
? COMPILE PROG2 RPG SYNTAX
? RPG FILE SOURCE = RPG/SOURCE MAGTAPE
? END
```

VECTOR FILE INPUT.

The vector input files consist of compile-time and execution-time vector files, both of which are described in the text that follows.

COMPILE-TIME VECTOR FILES. If the program requires table or array files to be included during compilation, the compiler requires an input card file labeled "RPG/VECTOR". This file may be input from a medium other than cards, through use of label equation (the FILE Card). Card input may be punched in either the BCL or EBCDIC character set.

Vector files must be entered in the same order as specified in the Extension Specifications, and each file must contain exactly the number of records specified. No separators define the end of one vector and the beginning of the next; records are read into one vector until that vector is full, then filling of the next vector is begun from subsequent records.

The following sample deck could be used to compile a source program (with compile-time vector input) contained on cards to library:

```
? COMPILE TESTER/RPG PROGRAM WITH RPG FOR LIBRARY
? DATA RPG/CARD
  source specification cards
? END
? DATA RPG/VECTOR
  table file cards
? END
```

EXECUTION-TIME VECTOR FILES. If the program requires table or array files to be included at the beginning of object program execution, the program will require card files with the labels as specified by the programmer in the File Description and Extension Specifications. The files may be input from a medium other than cards, through use of label equation (the FILE Card). Card input may be punched in either the BCL or EBCDIC character set.

COMPILER FILE NAMES.

Table 11-1 lists the files that are used by the compiler for source input and compilation output:

Table 11-1

Files Used for Source Input and Compilation Output

File Type	Device	Internal File Name	External File Name
Source input	Card Reader	SOURCE	RPG/CARD
Table input	Card Reader	TABCRD	RPG/VECTOR
Output listing	Line Printer	LINE	RPG/PRINT

APPENDIX A

B 1700 RPG COLLATING SEQUENCE

Ascending Order

Character	Hexadecimal Equivalent
blank	40
[	4A
.	4B
<	4C
(	4D
+	4E
|	4F
&	50
]	5A
$	5B
*	5C
)	5D
;	5E
¬	5F
-	60
/	6A
,	6B
%	6C
—	6D
>	6E
?	6F
:	7A
#	7B
@	7C
'	7D
=	7E
"	7F
A	C1
B	C2
C	C3
D	C4
E	C5
F	C6
G	C7
H	C8
I	C9
!	D0
J	D1
K	D2
L	D3
M	D4
N	D5
O	D6
P	D7
Q	D8
R	D9
S	E2
T	E3
U	E4
V	E5
W	E6
X	E7
Y	E8
Z	E9
0	F0
1	F1
2	F2
3	F3
4	F4
5	F5
6	F6
7	F7
8	F8
9	F9

APPENDIX B

HEXADECIMAL-DECIMAL CONVERSION TABLE

The table in this appendix provides for direct conversion of decimal and hexadecimal numbers in the ranges:

Hexadecimal	Decimal
000 to FFF	0 to 4095

For numbers outside the range of the table, add the following values to the table figures:

Hexadecimal	Decimal
1000	4096
2000	8192
3000	12288
4000	16384
5000	20484
6000	24576
7000	28672
8000	32768
9000	36864
A000	40960
B000	45056
C000	49152
D000	53248
E000	57344
F000	61440

APPENDIX B (cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
010	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
020	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
030	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
040	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
050	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
060	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
070	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
080	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
090	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
0A0	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
0B0	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
0C0	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
0D0	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
0E0	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
0F0	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
100	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271
110	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287
120	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303
130	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
140	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335
150	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351
160	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367
170	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383
180	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399
190	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415
1A0	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431
1B0	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447
1C0	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463
1D0	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479
1E0	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495
1F0	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511

APPENDIX B (cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527
210	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543
220	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559
230	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575
240	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591
250	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607
260	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623
270	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639
280	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655
290	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671
2A0	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687
2B0	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703
2C0	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719
2D0	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735
2E0	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751
2F0	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767
300	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783
310	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799
320	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815
330	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831
340	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847
350	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863
360	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879
370	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895
380	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911
390	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927
3A0	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943
3B0	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959
3C0	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975
3D0	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991
3E0	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007
3F0	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

APPENDIX B (cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
400	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
410	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
420	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
430	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
440	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
450	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
460	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
470	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
480	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
490	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A0	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B0	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C0	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D0	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E0	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F0	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
500	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
510	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
520	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
530	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
540	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
550	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
560	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
570	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
580	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
590	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A0	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B0	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C0	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D0	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E0	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F0	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535

APPENDIX B (cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
600	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
610	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
620	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
630	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
640	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
650	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
660	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
670	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
680	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
690	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A0	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B0	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C0	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D0	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E0	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F0	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791
700	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
710	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
720	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
730	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
740	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
750	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
760	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
770	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
780	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
790	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A0	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B0	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C0	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D0	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E0	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F0	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047

APPENDIX B (cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
800	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
810	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
820	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
830	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
840	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
850	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
860	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
870	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
880	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
890	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A0	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B0	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C0	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D0	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E0	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F0	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
900	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
910	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
920	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
930	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
940	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
950	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
960	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
970	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
980	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
990	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A0	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B0	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C0	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D0	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E0	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F0	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

APPENDIX B (cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A00	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A10	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A20	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A30	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A40	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A50	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A60	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A70	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A80	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A90	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA0	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB0	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC0	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD0	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE0	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF0	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B00	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B10	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B20	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B30	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B40	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B50	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B60	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B70	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B80	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B90	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA0	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB0	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC0	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD0	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE0	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF0	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071

APPENDIX B (cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C00	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C10	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C20	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C30	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C40	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C50	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C60	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C70	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C80	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C90	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA0	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB0	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC0	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD0	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE0	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF0	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327
D00	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D10	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D20	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D30	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D40	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D50	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D60	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D70	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D80	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D90	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA0	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB0	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC0	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD0	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE0	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF0	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583

APPENDIX B (cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E00	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E10	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E20	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E30	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E40	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E50	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E60	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E70	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E80	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E90	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA0	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB0	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC0	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED0	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE0	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF0	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F00	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F10	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F20	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F30	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F40	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F50	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F60	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F70	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F80	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F90	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA0	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB0	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC0	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD0	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE0	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF0	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

APPENDIX C

OPERATION CODES

GENERAL.

Ten categories of Operation Codes are provided to allow most types of data manipulation required by the RPG programmer. Operation Codes are entered only in the OPERATION field of the Calculation Specifications, and specify what type of arithmetic or logical operation is to be performed on the associated operands.

ARITHMETIC OPERATIONS.

Arithmetic operations are only allowed on numeric data items. FACTOR 1 and FACTOR 2 must name numeric fields or contain numeric literals; the RESULT FIELD must be numeric. All fields involved in arithmetic operations must be less than 16 characters in length; truncation will occur on longer fields. All results will be signed.

NOTE

An array name may be entered in FACTOR 1 or FACTOR 2, as long as the RESULT FIELD also specifies an array. In this case, the designated arithmetic operation will be performed on each element of the array or arrays designated as factors, with the result being placed in the corresponding elements of the array designated in the RESULT FIELD. The operation is terminated when the end of the shortest array is reached.

ADD.

This operation adds the contents of FACTOR 2 to the contents of FACTOR 1 and stores the sum in the RESULT FIELD. FACTOR 1 and FACTOR 2 are not affected by this operation, unless one of them is also designated as the RESULT FIELD.

DIV.

This operation divides the contents of FACTOR 1 by the contents of FACTOR 2 and places the quotient in the RESULT FIELD. FACTOR 1 and FACTOR 2 are not affected by this operation, unless one of them is also designated as the RESULT FIELD.

Any remainder resulting from the divide operation will be lost unless the next operation specified is the MOVE REMAINDER operation (MVR); if so, the result of the divide operation cannot be half adjusted.

NOTE

If FACTOR 2 is equal to zero, the program will be discontinued by the MCP.

MULT.

This operation multiplies the contents of FACTOR 1 by the contents of FACTOR 2 and stores the product in the RESULT FIELD. FACTOR 1 and FACTOR 2 are not affected by this operation, unless one of them is also designated as the RESULT FIELD.

MVR.
This operation moves the REMAINDER from a previous DIV operation to the designated RESULT FIELD. The MVR operation must immediately follow the DIV operation, and FACTOR 1 and FACTOR 2 must be left blank. The RESULT FIELD for the MVR operation must be the same length as FACTOR 2 for the DIV operation; if the RESULT FIELD is shorter, truncation may occur.

SQRT.
This operation derives the SQUARE ROOT of the contents of FACTOR 2 and places it in the RESULT FIELD. FACTOR 1 must be left blank.

NOTE

> If the FACTOR 2 operand is negative, the program will display a "SQRT" error message and halt. The operator may resume processing, in which case the RESULT FIELD will set to zero.

SUB.
This operation subtracts the contents of FACTOR 2 from the contents of FACTOR 1 and places the difference in the RESULT FIELD. FACTOR 1 and FACTOR 2 are not affected by this operation, unless one of them is also designated as the RESULT FIELD.

XFOOT.
This operation is used to crossfoot total the elements of an array. All the elements of the array specified by FACTOR 2 are summed and the total is placed in the RESULT FIELD. FACTOR 1 must be left blank.

Z-ADD.
This operation sets the RESULT FIELD to zeros, and adds the contents of FACTOR 2 to the RESULT FIELD. FACTOR 1 must be left blank. FACTOR 2 is not affected by this operation.

Z-SUB.
This operation sets the RESULT FIELD to zeros, then subtracts the contents of FACTOR 2 from the RESULT FIELD. FACTOR 1 must be left blank. This operation is used to change the sign of the field designated by FACTOR 2.

MOVE OPERATIONS.
These operations transfer the contents of FACTOR 2 to the RESULT FIELD. FACTOR 2 is not affected by these operations. FACTOR 1 must be left blank. RESULTING INDICATORS may not be specified. If FACTOR 2 and RESULT FIELD are identical (same length), MOVE and MOVEL operate the same. FACTOR 2 may be a field or literal; however, a RESULT FIELD may be a field or a vector.

NOTE

> An array name may be entered in FACTOR 2 as long as the RESULT FIELD also specifies an array. In this case, the des-

> ignated move operation will be performed on each element of the array designated in FACTOR 2, with the result being placed in the corresponding elements of the array designated in the RESULT FIELD. The operation is terminated when the end of the shorter array is reached.

For the purpose of MOVE, MOVEL defines the "sign" of the variable as (1) the algebraic sign if the variable is numeric, or (2) the low order zone if the variable is alpha.

Move operations may be specified between fields of different data types; however, decimal points are ignored in such moves.

MOVE
This operation moves characters from FACTOR 2 to the RESULT FIELD, starting with the rightmost character and continuing until either the source field is exhausted, or the destination field is filled. The sign of FACTOR 2 is moved to the sign position of the RESULT FIELD.

When an alphanumeric-to-numeric move is specified, the digit portion of each character is converted to its corresponding numeric character as it is moved to the RESULT FIELD. Blanks are transferred as zeros. The converse is true for numeric-to-alphanumeric moves.

MOVEL.
This operation moves, left-justified, characters from FACTOR 2 to the RESULT FIELD, starting with the leftmost character and continuing until either the source field is exhausted, or the destination field is filled.

When an alphanumeric-to-numeric move is specified, only the digit portion of each character is moved to the RESULT FIELD. Blanks are transferred as zeros. The sign is transferred only if the RESULT FIELD is numeric and not greater in length than FACTOR 2 or if the result is alpha and equal in length to FACTOR 2.

The sign of FACTOR 2 is the algebraic sign if it is a numeric data item or low order zone if FACTOR 2 is alpha. The algebraic sign is specified as the lower order digit if right signed or the high order digit if left signed.

MOVE_ZONE_OPERATIONS.
These operations are used to move the zone portion of only one character. Factor 2 is not affected by this operation, and only the zone portion of one character of the RESULT FIELD is affected. FACTOR 1 must be left blank.

NOTE

> An array name may be entered in FACTOR 2, as long as the RESULT FIELD also specifies an array. In this case, the designated move zone operation will be

performed on each element of the array designated in FACTOR 2, with the result being placed in the corresponding elements of the array designated in the RESULT FIELD. The operation is terminated when the end of the shorter array is reached.

MHHZO.
This operation moves the ZONE from the leftmost (HIGH) position of FACTOR 2 to the leftmost (HIGH) position of the RESULT FIELD. Both source and destination fields must be alphanumeric.

MHLZO.
This operation moves the ZONE from the leftmost (HIGH) position of FACTOR 2 to the rightmost (LOW) position of the RESULT FIELD. FACTOR 2 must be alphanumeric.

MLHZO.
This operation moves the ZONE from the rightmost (LOW) position of FACTOR 2 to the leftmost (HIGH) position of the RESULT FIELD. The RESULT FIELD must be alphanumeric.

MLLZO.
This operation moves the ZONE from the rightmost (LOW) position of FACTOR 2 to the rightmost (LOW) position of the RESULT FIELD.

COMPARE OPERATIONS.
These operations are used to test specified fields for certain conditions. The results of these operations are shown by the setting of the specified RESULTING INDICATORS. Fields are not affected by the operation.

COMP.
This operation compares FACTOR 1 with FACTOR 2, causing the RESULTING INDICATORS to be set as follows:

a. HIGH (columns 54-55) - FACTOR 1 > FACTOR 2
b. LOW (columns 56-57) - FACTOR 1 < FACTOR 2
c. EQUAL (columns 58-59) - FACTOR 1 = FACTOR 2

Both fields must be of the same type.

COMPARISON OF NUMERIC FIELDS. The comparison of numeric fields is based on their respective values considered purely as algebraic quantities. The length of the fields, in terms of digits, is not itself significant. Both fields are automatically aligned on their decimal points, and leading or trailing zeros are supplied, as needed, to make the lengths identical.

Numeric fields of more than 31 digits cannot be compared.

COMPARISON OF ALPHANUMERIC FIELDS. The comparison of alphanumeric fields begins with the leftmost character of each and proceeds, character-by-character, to the right until a pair of unequal characters is encountered. These characters are compared according to the EBCDIC collating sequence, and the field that contains the character found to be higher in the collating sequence is considered to be the HIGH field.

If the character-by-character comparison reaches the end of the fields, they are considered to be EQUAL.

Both fields are automatically aligned on their leftmost characters, and trailing spaces are supplied to the shorter field.

Alphanumeric fields of more than 511 characters cannot be compared.

TESTZ.
This operation tests the ZONE portion of the leftmost character of the RESULT FIELD, setting the RESULTING INDICATORS to the results of the test. The RESULT FIELD must be alphanumeric; FACTOR 1 and FACTOR 2 must be blank.

The character under test will set a specific result indicator, according to the following:

Character Under Test	Resulting Indicator Set
A-I, &	PLUS (columns 54-55)
J-R, -	MINUS (columns 56-57)
All others	ZERO (columns 58-59)

SETTING INDICATORS.
Up to three indicators may be set ON or OFF with one operation. These are entered, left to right, in the RESULTING INDICATORS field. FACTOR 1, FACTOR 2, and the RESULT FIELD must be left blank. The following rules must be observed when setting indicators:

a. The following indicators may not be set ON or OFF: 1P, MR, LO, or U1-U8.

b. Setting a control level indicator (L1-L9) ON or OFF does not affect any other control level indicator.

c. All control level and record identifying indicators are automatically turned OFF after detail output operations are completed, regardless of any previous SETON or SETOF operations.

d. If any halt indicators (H0-H9) are set ON and are not turned OFF before the detail output operations finish, the program will halt.

e. If the LR indicator is turned ON by a SETON operation which is conditioned by a control level indicator (columns 7-8 on Calculations Specifications form), the program will stop after all total output operations are completed. If the LR indicator is turned ON by a SETON operation which is not conditioned by a control level indicator, the program stops after the next total output operation is completed.

SETON.
This operation sets the indicators entered in the RESULTING INDICATORS field ON.

SETOF.
This operation sets the indicators entered in the RESULTING INDICATORS field OFF.

PROGRAM BRANCHING OPERATIONS.

Operations within the Calculation Specifications are normally performed in the order in which they are written. Branching operations allow variation of the order of operation; thus, conditional branching and repetitive operations are possible.

GOTO.
This operation causes the program to branch (GO TO) to some other instruction rather than "falling through" to the next sequential operation. Branching both forward and backward is allowed. However, branching to a TOTAL CALCULATION from a DETAIL CALCULATION is illegal in IBM's RPG II syntax.

FACTOR 2 must contain a label (TAG), which must follow the rules for formation of labels as described in section 2. FACTOR 1, RESULT FIELD, and the RESULTING INDICATORS must be left blank.

NOTE

Branching into or out of subroutines is not allowed.

TAG.
This operation is used to identify the point where a GOTO operation will branch. FACTOR 1 must contain a unique label (TAG), which must follow the rules for formation of labels as described in section 2. FACTOR 2, RESULT FIELD, and the RESULTING INDICATORS must be left blank. Conditioning by a control level indicator in columns 7-8 is permissible if the TAG is part of TOTAL CALCULATIONS. The INDICATOR field (columns 9-17) must be left blank.

LOOKUP OPERATIONS.

The lookup operation is used to search a table or array (vector) for a special element.

LOKUP.
This operation is used to search a vector for a particular data item. The vector name is entered in FACTOR 2. FACTOR 1 is used to name a data item called the SEARCH WORD. The LOKUP operation causes the des-

ignated vector to be searched in an attempt to find an element that matches the SEARCH WORD. The SEARCH WORD may be an alphanumeric or numeric constant, a field name, a vector element, or a table name. If the SEARCH WORD (FACTOR 1) is a table name, it refers only to the element of the table last selected by a LOKUP operation. The SEARCH WORD must be of the same length, type, and number of decimal positions (if numeric) as each element of the vector being searched.

The RESULT FIELD is used only if two vectors are to be specified in the LOKUP operation.

RESULTING INDICATORS must always be assigned to a LOKUP operation specification, and are used to indicate the type of vector search desired, as follows:

a. An indicator assigned to EQUAL (columns 58-59) specifies that the vector is to be searched for an item equal to the SEARCH WORD.

b. An indicator assigned to HIGH (columns 54-55) specifies that the vector is to be searched until the first item higher in sequence than the SEARCH WORD is found.

c. An indicator assigned to LOW (columns 56-57) specifies that the vector is to be searched until the first item lower in sequence than the SEARCH WORD is found.

The following rules must be observed when assigning RESULTING INDICATORS to a LOKUP operation:

a. At least one RESULTING INDICATOR must be assigned, but more than two is not allowed.

b. If two RESULTING INDICATORS are assigned, one of them must be assigned to EQUAL.

c. Any search operation for other than an EQUAL condition (LOW, HIGH, LOW and EQUAL, HIGH and EQUAL) is allowed only if the vector is in sequence.

d. If the search is successful, the RESULTING INDICATOR designating the type of search is turned ON. If two indicators are assigned, only the one indicating the comparison of the SEARCH WORD and the vector element will be turned ON.

e. If no element of the vector being searched satisfies the conditions, no RESULTING INDICATOR will be turned ON.

A LOKUP operation may begin searching a vector at some element other than the first by assigning a numeric literal or data item to the vector as an index. The search will start at the designated element and continue until the desired item is found or the end of the vector is reached. If the search is unsuccessful and the index is a data item rather than a literal, the index will be set to one. If the search is successful, the index will contain the number of the element which satisfied the designated conditions. If a literal is used as an index, it will not be changed to indicate the search result.

SINGLE-VECTOR LOKUP. When searching only one vector, FACTOR 1 (the SEARCH WORD), FACTOR 2 (the vector to be searched), and at least one (but not more than two) RESULTING INDICATORS must be specified. CONTROL LEVEL and conditioning INDICATORS may also be used. The RESULT FIELD must be left blank.

If the search is successful, the proper RESULTING INDICATOR is turned ON to indicate the comparison that caused the search to terminate. For a table, a copy of the element that satisfied the conditions is saved in a "hold area", replacing the previous contents of that area. If a data item is assigned to the vector as an index, it will contain the number of the element that satisfied the conditions. For an array, the element is not saved in a hold area; the index only points to the location of the element in the array.

If the search is not successful, no RESULTING INDICATORS are turned ON, the hold area is not affected, and any index assigned is set to one.

TWO-VECTOR LOKUP. When searching two vectors, FACTOR 1 (the SEARCH WORD), FACTOR 2 (the vector to be searched), the RESULT FIELD (a corresponding or related vector), and at least one (but not more than two) RESULTING INDICATORS must be specified. CONTROL LEVEL and conditioning INDICATORS may also be used.

When two vectors are specified in a LOKUP operation, only the vector named in FACTOR 2 is actually searched. If the search is successful, the proper RESULTING INDICATOR is turned ON to indicate the comparison that caused the search to terminate. For tables, corresponding elements from both vectors are saved in "hold areas", replacing the previous contents of each area. If a data item is assigned to either vector as an index, it will contain the number of the element that satisfied the conditions. For arrays, the elements are not saved in hold areas; the indices only point to the locations of the elements in each array.

The vectors specified in a LOKUP operation need not be of the same type (i.e., both tables or both arrays). Each will be handled properly, with updating of indices and hold areas performed as appropriate.

If the search is not successful, no RESULTING INDICATORS are turned ON, the hold areas are not affected, and any indices assigned are set to one.

NOTE

> A subscript error condition may occur during execution if the vector designated in the RESULT FIELD has fewer elements than the vector designated in FACTOR 2.

SUBROUTINES.

Subroutine operation codes are used only to delimit the beginning and end of a subroutine, or to call a subroutine for execution from some point in the program. Specification lines within a subroutine may contain the entries SR, OR, & AN (or be left blank) in the CONTROL LEVEL field (columns 7-8), and all subroutines must be the last operations specified in the Calculation Specifications.

BEGSR.
This operation code is used to indicate the beginning of a subroutine. FACTOR 1 must contain the name of the subroutine, which must follow the rules for formation of labels as described in section 1. FACTOR 2, the RESULT FIELD, and RESULTING INDICATORS must be left blank.

ENDSR.
This operation code is used to indicate the end of a subroutine. FACTOR 1 may contain a label. This label is used like a TAB as a point for a GOTO operation *within* the subroutine to branch, thus allowing exits from different points within the subroutine. FACTOR 2, the RESULT FIELD, and RESULTING INDICATORS must be left blank.

EXSR.
This operation causes execution of a subroutine, and may appear anywhere within the Calculation Specifications. When execution of the subroutine is completed, control returns to the next line following the EXSR operation.

The EXSR operation may be conditioned by an indicator, allowing the subroutine to be executed (called) only when all the conditions are satisfied. FACTOR 2 must contain the name of the subroutine being called, which must be the same name entered in FACTOR 1 of a BEGSR operation. FACTOR 1, the RESULT FIELD, and RESULTING INDICATORS must be left blank.

NOTE

Subroutines may not be nested in the Source Language.

PROGRAMMED CONTROL OF INPUT AND OUTPUT.

Under the normal B 1700 RPG program cycle, a record is read, calculations are performed (using the data from that input record), and an output record is written. The CHAIN, DSPLY, EXCPT, FORCE, and READ operations allow greater control over input and output, providing the capability to read and write records at times other than those normally available as part of the B 1700 RPG program cycle.

CHAIN.
This operation causes a record to be read from a CHAINED disk file during calculations or to position the actual key for the file so that a record will be written into the proper location during output. It is used for random processing of a direct or indexed file.

The RESULT FIELD, FIELD LENGTH, DECIMAL POSITIONS, and HALF ADJUST fields must be left blank. Indicators in columns 7-17 may be assigned. If the chained file is conditioned by an external indicator in the File Description Specifications, the CHAIN operation must be conditioned by the same indicator.

The HIGH subfield (columns 54-55) of the RESULTING INDICATORS field may contain an entry. If the designated record is not found on a read, the specified indicator will turn ON. The LOW and EQUAL subfields (columns 56-59) *should* be left blank. However, the LOW subfield *may* contain an entry; if so, it *must* be the same as the entry in the HIGH subfield.

CHAINED INPUT FILES. For CHAINED files declared as input, the CHAIN operation causes a record to be read during calculations. The desired record must be identified to the program, so that the proper input data will be made available from disk. For direct files, an actual key (relative record number) is used; for indexed files, a data key is used. The keys must be in ascending sequence and in key format (PACKED). The compiler will HALT with out-of-sequence keys and force an End-of-Job.

Direct Files.

FACTOR 1 must contain the actual key (relative record number), either as a numeric literal or as the name of a numeric field that contains the key. FACTOR 2 must contain the filename for the file from which the record will be read.

The actual key specified will be converted automatically to an absolute disk address, from which location the input record will be read. If the designated key points to a record outside the limits of the file, the indicator specified in columns 54-55 will turn ON and the record will not be read.

Indexed Files.

FACTOR 1 must contain the name of the data key, the field that will be compared against the index and the KEY FIELDS in the data file in order to locate the desired record. FACTOR 2 must contain the filename for the file from which the record will be read.

The data key specified will be compared to the entries in the index, which will point to a portion of the file on disk. That portion of the file pointed to by the index will be searched until the data key matches the KEY FIELD of a data record. If the desired record cannot be located in the file, the indicator specified in columns 54-55 will turn ON and no record will be read.

CHAINED OUTPUT FILES. For chained files declared as output, the CHAIN operation causes the proper disk address to be generated for writing the output record during normal output operations. The desired record location must be made available prior to output, so that the record will be written into the proper place in the file.

Direct output files are handled in the same manner as direct input files, except that no record is read or written. Only the disk address to be used for output is generated. If the designated key points to a record outside the limits of the file, the indicator specified in columns 54-55 will turn ON, and the output operation is inhibited.

NOTE
Output indexed files are not allowed.

CHAINED UPDATE FILES. For chained files declared as UPDATE, the CHAIN operation causes a record to be read during calculations, plus saving the disk address from where the record was read. This address is used during output, and causes the updated record to be written back into the same location from which it was read.

Direct update and indexed update files are handled in the same manner as their respective input files. Also, the disk address is saved for use by the output operation, to cause the record to be written back into the same location in the file.

DSPLY.
This operation causes data to be displayed on the console printer (SPO). Also, certain entries allow low-volume data to be entered by the systems operator through the SPO. The filename of the file assigned to the SPO must be entered in FACTOR 2. The FIELD LENGTH, DECIMAL POSITIONS, HALF ADJUST, and RESULTING INDICATORS fields must be left blank. Control level and conditioning indicators may be assigned. FACTOR 1 is used to name a data item, which may be a field name or a vector element, or to specify a literal (numeric or alphanumeric). The RESULT FIELD is optional and, if used, must specify the name of a data item (field name or vector element). The DSPLY operates in the following manner:

a. The data item or literal specified by FACTOR 1 is typed, and

b. If the RESULT FIELD is blank, the program proceeds to the next operation; otherwise:

    1) The data item specified by the RESULT FIELD is typed,

    2) The program waits for input from the system operator via the SPO (ACCEPT message).

The following rules must be observed when entering data through the SPO:

a. Numeric data must be entered with leading zeros; the number of characters typed in must be equal to the length of the field into which the data is being entered.

b. Signed numeric fields may not be entered through the SPO.

c. Alphanumeric data must be entered left-justified; all spaces typed in are considered part of the data.

d. Fields of more than 50 characters may not be displayed or accepted.

EXCPT.

This operation allows EXCEPTION records to be written during calculations. Every time the EXCPT operation is executed, all lines in the Output-Format Specifications with a TYPE entry (column 15) of E will be written (dependent, of course, on conditioning indicators).

The EXCPT operation may have CONTROL LEVEL and conditioning INDICATORS assigned; all other fields must be left blank.

FORCE.

This operation enables selection of the file from which the next record is to be taken for processing. This specification overrides the normal record selection process that occurs during input. The FORCE operation does not actually read any records during calculations, but only selects the file which will supply the next record for processing.

The FORCE operation may apply only to input, update, or combined files designated as PRIMARY or SECONDARY. FACTOR 2 must contain the name of the file to be forced; all other fields (except CONTROL LEVEL and conditioning INDICATORS) must be left blank.

If more than one FORCE operation is executed during one program cycle, all but the last are ignored. When End-of-File is encountered on a "forced" file, normal record selection processes will determine the next record to be provided for input.

READ.

This operation is used to cause a record to be read from a demand file during calculations. This differs from the FORCE operation because FORCE causes input during the *next* program cycle, whereas the READ operation causes input during the *current* program cycle. It also differs from the CHAIN operation because CHAIN is used to read records *randomly*, whereas the READ operation is used to read records *sequentially*.

FACTOR 2 must contain the name of the file to be read. The FACTOR 1, RESULT FIELD, FIELD LENGTH, DECIMAL POSITIONS, and HALF ADJUST fields must be left blank. The READ operation may be conditioned by CONTROL LEVEL or conditioning INDICATORS in columns 7-17.

An indicator should be specified in columns 58-59; if not, an End-of-File condition on the READ will cause the program to be discontinued. If End-of-File is reached and an indicator is specified, that indicator will be turned ON. Any subsequent attempts to read from the same file will cause the program to be discontinued.

The following rules must be observed when using the READ operation:

a. Only demand files designated as INPUT, UPDATE, or COMBINED may be read by the READ operation.

b. Sequence-checking in the Input Specifications is not allowed for demand files.

c. Control levels, matching fields, and look-ahead fields are not allowed for demand files.

d. If a demand file is conditioned by an external indicaor (U1-U8) which is not set, the READ operation will be ignored.

DEBUG OPERATION.
The DEBUG operation is a special-purpose function which simplifies the location and correction of errors in a RPG Program.

DEBUG.
This operation causes records to be written during calculations. These records contain specific information which may be helpful in locating errors in the program. DEBUG operations may appear in as many places as needed within the Calculation Specifications. Every time the DEBUG operation is executed, one or more fixed-format records will be written to an output device.

DEBUG operations will only be compiled into the program if the DEBUG field of the Control Card contains a 1. Otherwise, all DEBUG operations are ignored (treated as comments) during compilation.

FACTOR 1 is optional, and may contain a literal or field name, used to identify the particular DEBUG operation being executed. The literal or the value of the designated field is written as part of the output records. FACTOR 2 must contain the FILENAME of the output file to which the records are to be written. All DEBUG operations must specify the same output file. The RESULT FIELD is optional, and may contain the name of a field, table element, array element, or entire array. The contents of the designated data item will be written as part of the output records. The DEBUG operation may be conditioned by indicators in columns 7-17. Columns 49-59 must be left blank.

DEBUG OPERATION OUTPUT. One or more records will be written as output from every DEBUG operation. The first record is always written; the second and subsequent records will be written only if the RESULT FIELD contains an entry. See Figure C-1 for an example of the output produced by the DEBUG operation.

ZIP OPERATION. Zip may be used for programmatic scheduling of subordinate object programs contained in the Systems Program Library or MCP control functions.

The Operation Code is ZIP. Factor 2 contains an Alpha Variable of the program being Zipped to. The operation may use conditioning indicators. All other fields must be left blank.

SOURCE PROGRAM

```
0101 H        1

0102 FCARDIN  IPE     80  80               READER
0103 FPRINTOUTD      132 132               PRINTER

0203 E                      ARY          5  6 2

0301 ICARDIN  AA  01
0302 I                                               1   62ARDATA       020304
0303 I                                               9  100IX           05
0304 I                                              13   80 CMNT

0401 C    01 05               MOVE ARDATA    ARY,IX
0402 C             '1'        DEBUGPRINTOUT  ARY
0403 C                        SETON                         758085
0404 C             '2'        DEBUGPRINTOUT  ARDATA
0405 C                        SETOF                         80
0406 C                        DEBUGPRINTOUT
0407 C                        SETOF                         758505
0408 C                        DEBUGPRINTOUT

0501 OPRINTOUTD  2      01
0502 O                                    9 'ELEMENT #'
0503 O                          IX    Z  11
0504 O                                   33 ' OF ARRAY ''ARY'' LOADED'
0505 O                                   40 ' WITH: '
0506 O                          ARDATA   48 '  0 .  '
0507 O                          CMNT    120
```

OUTPUT LISTING

```
DEBUG- 1           INDICATORS ON-01 02 05
ARY    ,0001  000136
ARY    ,0002  000000
ARY    ,0003  000000
ARY    ,0004  000000
ARY    ,0005  000000
DEBUG- 2           INDICATORS ON-01 02 05 75 80 85
ARDATA        000136
DEBUG-             INDICATORS ON-01 02 05 75 85
DEBUG-             INDICATORS ON-01 02
ELEMENT # 1 OF ARRAY 'ARY' LOADED WITH:     1.36      LOAD ARY,1 WITH    1.36

DEBUG- 1           INDICATORS ON-01 04 05
ARY    ,0001  000136
ARY    ,0002  000000
ARY    ,0003  000000
ARY    ,0004  000000
ARY    ,0005  000000
DEBUG- 2           INDICATORS ON-01 04 05 75 80 85
ARDATA        000000
DEBUG-             INDICATORS ON-01 04 05 75 85
DEBUG-             INDICATORS ON-01 04
ELEMENT # 2 OF ARRAY 'ARY' LOADED WITH:     0.00      LOAD ARY,2 WITH    0.00

DEBUG- 1           INDICATORS UN-01 02 05
ARY    ,0001  000136
ARY    ,0002  000000
ARY    ,0003  123456
ARY    ,0004  000000
ARY    ,0005  000000
DEBUG- 2           INDICATORS ON-01 02 05 75 80 85
ARDATA        123456
DEBUG-             INDICATORS ON-01 02 05 75 85
DEBUG-             INDICATORS ON-01 02
ELEMENT # 3 OF ARRAY 'ARY' LOADED WITH: 1234.56      LOAD ARY,3 WITH 1234.56

DEBUG- 1           INDICATORS ON-01 02 05
ARY    ,0001  000136
ARY    ,0002  000000
ARY    ,0003  123456
ARY    ,0004  999999
ARY    ,0005  000000
DEBUG- 2           INDICATORS ON-01 02 05 75 80 85
ARDATA        999999
DEBUG-             INDICATORS ON-01 02 05 75 85
DEBUG-             INDICATORS ON-01 02
ELEMENT # 4 OF ARRAY 'ARY' LOADED WITH: 9999.99      LOAD ARY,4 WITH 9999.99

DEBUG- 1           INDICATORS ON-01 02 05
ARY    ,0001  000136
ARY    ,0002  000000
ARY    ,0003  123456
ARY    ,0004  999999
ARY    ,0005  654321
DEBUG- 2           INDICATORS ON-01 02 05 75 80 85
ARDATA        654321
DEBUG-             INDICATORS ON-01 02 05 75 85
DEBUG-             INDICATORS ON-01 02
ELEMENT # 5 OF ARRAY 'ARY' LOADED WITH: 6543.21      LOAD ARY,5 WITH 6543.21
```

Figure C-1. DEBUG Operation Output

APPENDIX D

COMPILE-TIME ERROR AND DIAGNOSTIC MESSAGES

GENERAL.

G085 - FILE NAME NEVER DEFINED
A filé name has been referenced that has never been defined in File Specification.

G095 - INVALID AND/OR LINE SPECIFICATION

G304 - U1 - U8 MAY BE SET EXTERNALLY ONLY
These external indicators may not be set programmatically and should only be used to indicate at execution time, whether or not the file is to be used by the program.

G306 - INDICATOR NEVER DEFINED
This message is caused from an indicator being tested that has never been set.

G315 - FIELD NAME NEVER DEFINED
This message is caused from a variable name being referenced that has never been defined.

G524 - CHAINING KEY LENGTH NOT EQUAL CHAINED FILE KEY LENGTH
This message indicates the key length specified on the File Specification does not agree with the variable used to initiate chaining.

G999 - DATA NOT NEEDED/EXPECTED IN THIS CARD

FILE DESCRIPTION SPECIFICATIONS.

Refer to Section 4 for further information.

F023 - INVALID FILE NAME
The message is caused from an invalid file name in columns 7-14. A file name must be unique in the first seven characters.

F024 - REDEFINED FILE NAME
The file name specified in columns 7-14 does not contain a unique name in the first seven characters.

F026 - INVALID FILE TYPE
This message is caused from an illegal entry in column 15 of the File Specification form. Valid entries are I, O, U, C and D.

F028 - INVALID FILE DESIGNATION
This message is caused from an illegal entry in column 16 of the File Specification form. Blank must be used for output files, otherwise one of the following entries are required: P, S, C, T or D.

F033 - NO SEQUENTIAL INPUT FILE FOUND
A primary or secondary file is required.

F036 - INVALID END OF FILE ENTRY
This message is caused from an invalid entry in column 17. Valid entries are Blank or E.

F039 - INVALID SEQUENCE ENTRY
This message is caused from an invalid entry in column 18. Valid entries are A, D or Blank.

F042 - INVALID BLOCK-RECORD LENGTH
When the Block and Record length contain blanks a default of 132 is assumed for print files, and 80 for all other devices. When the block length is blank it will assume the Record length. When the block and record length contains valid numbers, the block length must be an integral part of the record length.

F044 - MULTI BUFFERS AND STACKER SELECTION ON SAME FILE
Stacker selection may not be used when Multi buffers are specified, due to the processing of the records from the buffer areas.

F044 - INVALID ENTRY TYPE OF ORGANIZATION
This message is caused from an invalid entry in column 32. Valid entries must contain I, 1-9 or Blank.

F046 - INVALID ENTRY ADDITION-UNORDERED
This message is caused from an invalid entry in column 66. Valid entries are U, A or Blank.

F049 - INVALID/MISSING EXTENSION CODE
This message is caused from an invalid entry in column 39. Valid entries are Blank, E for table files, and L may be specified for print files.

F052 - INVALID DEVICE
This message is caused from an invalid device specified in columns 40-46. Valid entries are READER, MFCU1, PUNCH, PRINTER, PRINTR2, TAPE, DISK, SPO or CONSOLE.

F056 - INVALID FILE FORMAT/FILE CLOSE
This message is caused from an invalid entry in column 19. Valid entries are F, V, or Blank.

F057 - INVALID INDICATOR MUST BE U1-U8
This message is caused from an invalid entry in columns 71-72. Valid entries are U1-U8 or Blank.

F400 - INVALID ENTRY MODE
This mesage is caused from an invalid entry in column 28. This field must contain R or be left Blank.

F404 - INVALID ENTRY ADDRESS TYPE
This message is caused from an invalid entry in column 31. Valid entries are A, K, P, N or Blank. On indexed files this field specifies the format of the key; it must be left Blank if the file is not indexed.

F407 - INVALID ENTRY ADDITION-UNORDERED

F543 - INVALID KEY LENGTH ENTRY
The key length is required when using indexed processing. When used this field (columns 29-30) must contain a valid number from 1-99 and the key length plus the key start location must not exceed the record length.

F549 - INVALID KEY START LOCATION
The key start location field is required when using indexed processing. This field (columns 35-38) must be a valid number and the key start location plus the key length must not exceed the record length.

F553 - INVALID KEYS IN CORE ENTRY
The keys in core entry (columns 60-65) must contain a valid number not to exceed 9,999.

EXTENSION SPECIFICATIONS.

Refer to Section 5 for further information.

E061 - CHAINING FIELD FOR THIS LEVEL NOT DEFINED ON INPUT
This message indicates that automatic chaining has been specified, and the corresponding level has not been specified on input.

E062 - INVALID FROM FILE NAME ENTRY
this message is caused from an invalid entry in columns 11-18. It is used as the file name of every execution-time vector but must be left blank if the vector is to be loaded at compile time or via Input or Calculation Specifications.

E063 - INVALID TO FILE NAME ENTRY
This message is caused from an invalid entry in columns 19-26. It is used as the file name of a vector file that is to be written or punched. A valid file name must be used as well as being previously defined on the File Description Specification form.

E067 - INVALID VECTOR NAME
This message is caused from an invalid table or array name in column 27-32. Valid vector names cannot exceed six characters and must be unique. Table names must begin with the letters TAB.

E068 - INVALID ENTRY NUMBER OF ENTRIES PER RECORD
This message is caused from an invalid entry in columns 33-35. This field must contain a valid number. This entry times the length of Vector A plus the length of Vector B (if applicable)

must not be greater than the From File length or (if applicable) 96.

E070 - INVALID ENTRY NUMBER OF ENTRIES PER VECTOR
This message is caused from an invalid entry in columns 36-39. This entry must be a valid number limited only to the size of the field.

E072 - INVALID ENTRY LENGTH OF ENTRY
This message is caused from an invalid entry in columns 40-42/52-54. This entry must be a valid number specifying the length of the entry of the vector. Alpha entries must not exceed 511 and numeric entries 31.

E074 - INVALID ENTRY PACKED
This message is caused from an invalid entry in column 43/55. Valid entries are Blank and P. P is only allowed on numeric fields.

E076 - INVALID DECIMAL POSITIONS
This message is caused from an invalid entry in column 44/56. Valid entries are 0-9 or Blank.

E077 - INVALID SEQUENCE ENTRY
This message is caused from an invalid entry in column 45/57. Valid entries are A, D or Blank.

E079 - INVALID ALTERNATE VECTOR NAME
This message is caused from an invalid table or array name in columns 46-51. Valid vector names cannot exceed six characters and must be unique. Table names must begin with the letters TAB.

E122 - REDEFINED VECTOR NAME
This message is caused from a vector name that has been redefined. Vector names must be the same length, type and decimal positions as used on the first definition.

E228 - INVALID/OUT OF BOUNDS LITERAL -- VECTOR, LITERAL --
Literals must contain digits 0-9 and cannot exceed the size of the vector.

E315 - INVALID/UNDEFINED FIELD NAME -- VECTOR, FIELD NAME --
This message is caused from an illegal field name or a field name that has not been defined.

LINE COUNTER SPECIFICATIONS.

Refer to Section 6 for further information.

L085 - INVALID FILE NAME OR FILE TYPE
This message specifies the file name specified in columns 7-14 is not valid or it has not been assigned to a print file. It must also have been previously defined on the File Specification.

L088 - WHEN DEFINED -FL- MUST BE IN COL 18-19

L090 - INVALID CHANNEL ENTRY
This message is caused from an illegal channel entry. Valid entries are 01-12, OL, or FL.

L091 - INVALID LINE ENTRY EXCEEDS FORM LENGTH OR 112
The line position entry must not exceed the form length or 112.

INPUT SPECIFICATIONS.

Refer to Section 7 for further information.

I084 - RECORD LINE ASSUMED 43-70 TREATED AS BLANK

I092 - INVALID FILE NAME
This message indicates the file name specified in columns 7-14 is not valid or it has not been assigned to an input file. It must also have been previously defined on the File Specifications.

I092 - FILE NOT DEFINED AS INPUT
The file type in column 15 of the File Specification form has not been Specified as I, U, C or D.

I093 - FIELD LINE ASSUMED 7-42 TREATED AS BLANK

I094 - RECORD LINE WITH FILE MUST BE FIRST INPUT SPEC
The first Input Specification must contain a file name.

I097 - NO FIELDS DEFINED IN LAST RECORD

I101 - SEQUENCE FIELD SPECIFICATION
This message is caused from an invalid entry in columns 15-16. It must contain a valid sequence entry, either 2 alpha characters other than ND or Rb, or a numeric entry from 01-99 in ascending order within the file.

I102 - INVALID ENTRY -NUMBER-
The number position entry in column 17 must contain 1 or N if the sequence entry is numeric, or Blank if alphabetic.

I103 - INVALID ENTRY -OPTION-
This message is caused from an invalid entry in column 18. Valid entries are O and Blank. The O may only be used on numerical entries and one numerical entry should be non-optional.

I107 - INVALID RECORD ID CODE
This message is caused from an invalid entry between 21-41. The position entry must fall within the record length.

I109 - INVALID ENTRY -STACKER-
This message is caused from an invalid entry in column 42. Valid entries are Blank and 1-6 on card files.

I111 - INVALID PACKED ENTRY
This message is caused from an invalid entry in column 43. Valid entries are P or Blank. P may only be specified on numeric field lines and if specified column 52 (decimal positions) must not be blank.

I113 - INVALID FROM TO ENTRY
The FROM and TO entries must be valid numbers and cannot exceed the record length. Numeric fields must not exceed 31. Alpha fields must not exceed 511. Vector load elements must be the same length of the element defined on the Extension Specifications and the entire array load must not be greater than the array length and modulo the element length.

I118 - INVALID FIELD NAME
This message is caused from an invalid field name in columns 53-58.

I119 - INVALID CONTROL LEVEL INDICATOR
This message is caused from an invalid entry in columns 59-60. Valid entries are Blank and level indicators L1-L9.

I120 - INVALID ENTRY MATCHING-CHAINING FIELD
This message is caused from an invalid entry in columns 61-62. Valid entries are Blank, M1-M9, and C1-C9.

I122 - REDEFINED FIELD NAME - LOOKAHEAD
A lookahead field name must be unique.

I122 - REDEFINED FIELD NAME
When used a duplicate field name must contain the same length and decimal positions entry.

I159 - INVALID RECORD INDICATOR
This message occurs when columns 19-20 do not contain a valid indicator or lookahead.

I166 - INVALID FIELD RESULTING INDICATOR
This message occurs when columns 65-70 do not contain Blanks or a valid indicator.

I170 - INVALID ENTRY FOR CHAIN OR DEMAND FILE
Chain or demand files may not contain control levels or matching fields.

I172 - INPUT FILE ORDER IS DIFFERENT FROM FILE SPEC
The input files must be in the same order as defined on the File Specifications.

I181 - REDEFINED LENGTH OF MATCHING-CHAINING FIELD
REDEFINED LENGTH OF CONTROL FIELD.
This message indicates the total length of a given level on the current record is not identical to the total length of the same level of a previous record.

I188 - INVALID RECORD RELATION INDICATOR
This message is caused from an invalid entry in columns 63-64. This field must contain a valid indicator or Blank.

I226 - INVALID ENTRY DECIMAL POSITIONS
The decimal positions entry in column 52 must contain a Blank or 0-9. On vector loads the decimal positions entry must be the same as the defined element.

I575 - UNREFERENCED OR MISSGROUPED REFERENCE OF FILE NAME
This message indicates an Input Specification has been referenced that has not been declared in the File Specification, or the file names appear more than once and not consecutively or it does not appear in the same order as the File Specification.

CALCULATION SPECIFICATIONS.

Refer to Section 8 for further information.

C122 - REDEFINED FIELD NAME
This message indicates the name defined in the result field has been previously defined with different length and decimal positions.

C122 - REDEFINED FIELD NAME - LOOK AHEAD
This message indicates the name defined in the result field has been previously defined as a look ahead field with different length and decimal positions.

C123 - INVALID CONTROL LEVEL INDICATOR
This message is caused from an invalid entry in columns 7-8. Valid entries are L0-L9, LR, AN, OR and when in a subroutine SR.

C125 - INVALID FACTOR 1
The field in columns 18-27 containing Factor 1 must be present if required, absent if not required as specified by the Operation Code. When present it must be a legal variable.

C128 - INVALID OPERATION CODE
The operation code specified in columns 28-32 is not legal.

C131 - INVALID FACTOR 2
The field in columns 33-42 containing Factor 2 must be present if required, absent if not required as specified by the operation code. When present it must be a legal variable.

C135 - INVALID RESULT FIELD
The field in columns 43-48 containing the result field must be present if required, absent if not required as specified by the operation code. When present it must be a legal variable but may not contain a literal.

C135 - INVALID RESULT FIELD NAME

C137 - INVALID RESULT FIELD LENGTH
This message is caused from an invalid entry in columns 49-51. The result field length must contain a valid number between 1-31 for numeric fields of between 1-511 for alpha fields.

C138 - INVALID DECIMAL POSITIONS
This message is caused from an invalid entry in column 52. Valid entries are Blank, 0-9.

C140 - INVALID ENTRY HALF ADJUST
This message is caused from an invalid entry in column 53. Valid entries are Blank and H. H may only be used with arithmetic operators.

C190 - INVALID BEGSR/ENDSR RELATIONSHIP
This message is used in conjunction with subroutines. Each subroutine must begin with a BEGSR and end with an ENDSR and may not contain another subroutine.

C200 - INVALID RESULTING INDICATOR
This message is caused from an invalid indicator or one that should not have been specified.

C207 - INVALID ALPHA LITERAL
Alpha literals must be contained in apostrophes and follow the rules for forming literals.

C207 - INVALID NUMERIC LITERAL
This message is caused from an invalid numeric literal. Numeric literals start with + - . , or 0-9 and may contain only 1 comma or 1 decimal point with no embedded blanks and only digits 0-9.

C214 - INVALID GOTO - TAG OR EXSR - BEGSR RELATIONSHIP
GOTO operations are only valid in conjunction with TAG and ENDSR labels. EXSR operators must be associated with a BEGSR.

C215 - FACTOR 1 AND FACTOR 2 BOTH LITERALS
It is illegal to have literals in both Factor 1 and Factor 2.

C221 - RESULT FIELD MAY NOT BE LARGE ENOUGH
This warning occurs when an overflow condition is likely to occur causing high order digits to be lost.

C232 - INVALID OR DUPLICATE TAG, BEGSR, OR ENDSR NAME

C304 - INVALID INDICATOR
A conditioning indicator must be specified on AND/OR lines and be a valid indicator.

C519 - INVALID AN - OR ENTRY
This message is caused from an illegal entry. The last of a series of AND/OR lines must contain a calculation operation.

OUTPUT-FORMAT SPECIFICATIONS.

Refer to Section 9 for further information.

0097 - NO FIELDS OR LITERALS DEFINED IN LAST RECORD

0142 - RECORD LINE ASSUMED 32-70 TREATED AS BLANK

0143 - INVALID OUTPUT RECORD TYPE
This message is caused from an invalid entry in column 15. Valid entries are H, D, T or E. This entry may also contain a N or R on AND/OR lines.

0146 - INVALID FILE NAME
This message specifies the file in columns 7-14 is not valid or it has not been previously defined on the File Description Specifications as an output file.

0148 - INVALID FIELD NAME
This message is caused from an invalid entry in columns 32-37. A field name previously defined on Input, Extension or Calculations Specification may be used, or one of the special reserved words.

0150 - INVALID BLANK AFTER ENTRY
The blank after entry in column 39 must contain a Blank or B.

0151 - INVALID ENDING POSITION
This message is caused from an invalid entry in columns 40-43. This entry must contain a valid number that must exceed the record length of the file and the length of the field must not underflow the record. This must also include the size of the editing symbols when used.

0152 - INVALID PACKED ENTRY
This message is caused from an invalid entry in column 44. Valid entries are Blank or P. When P is specified a numeric variable must be described in columns 32-37.

0154 - RECORD LINE WITH FILE MUST BE FIRST OUTPUT SPEC
The first Output Specifications must be a record line containing a file specification.

0154 - FIELD LINE ASSUMED 7-22 TREATED AS BLANK

0212 - EXCPT CALC WITHOUT EXCEPTION OUTPUT RECORD TYPE
The EXCPT operation code has been used when no Exception output records have been defined.

0256 - INVALID STACKER/FETCH ENTRY
This message is caused from an invalid entry in column 16. Valid entries on punch files are 1-6 and on Print files F or Blank.

0258 - INVALID SPACE/SKIP ENTRY
This message is caused from an invalid entry in columns 17-18 or 19-22. Valid entries for Spacing are Blank, 0 or 1-9 for Skipping 0-99, A0-A9 or B0-B2 and must only be used on Print files.

0273 - INVALID OUTPUT INDICATOR
The output indicators in columns 23-21 must contain a Blank or a valid indicator. Valid indicators are 01-99, L0-L9, LR, MR, H0-H9, U1-U8, OA-OG, OV or 1P.

0276 - INVALID EDIT CODE
Column 38 must contain a Blank if the field name field contains an alpha variable or literal; when numeric it may contain a Blank, 1-4, A-B-C-D, J-K-L-M, or X-Y-Z.

0277 - INVALID EDIT WORD
The number of replaceable characters in the edit word (columns 45-70) must be equal or greater in length than the length of the field to be edited.

0279 - INVALID CONSTANT SIZE

0283 - INVALID FILE TYPE FOR OUTPUT RECORD
The file type has not been defined as O, U, C or an input/add file.

0289 - *PLACE OR *PRINT PRECEDES ALL FIELDS AND CONSTANTS

0548 - INVALID FIE ADDITION
The file referenced has not been declared as an add file but ADD was specified. ADD will be assumed.

0554 - ADD NOT SPECIFIED ASSUME -ADD-
All files, except update files, using "A" in column 66 in the File Specifications should have "ADD" in columns 16-18 of each record in the output of the corresponding file.

0998 - EDIT TO BE PERFORMED ON ALPHANUMERIC FIELD
Only numeric items may be edited.

0999 - BLANK AFTER INVALID FOR OUTPUT LITERAL
Blank after may not be used after an output literal.

DOLLAR CARD SPECIFICATIONS.
Refer to Section 10 for further information.

DOL1 - INVALID ENTRY - NEGATE-
This message is caused from an invalid or illegal entry in column 8. Valid entries are Blank or N; however, on some Dollar Card Specifications it is illegal to use the Not option.

DOL2 - INVALID ENTRY - KEYWORD-
This mesage is caused from the keyword entry not containing a valid option.

DOL3 - INVALID ENTRY - VALUE-
This message is caused from an illegal entry in the value field of the $ option. Alpha entries must contain a character from A to Z in column 15 and numeric entries must contain blanks or zeroes in columns 15 and 16.

APPENDIX E

EXECUTION-TIME ERROR MESSAGES

GENERAL.

Certain conditions may arise during program execution which require operator notification and, in most cases, acknowledgment. Some program errors are recoverable, and some are not.

INPUT ERROR MESSAGES.

The following messages all denote error conditions arising during input, and all are recoverable. The program will request a reply from the operator; he may respond with "GO", in which case the erroneous record will be ignored and the next record from the same file will be read; or, he may respond with "STOP", in which case the erroneous record will be ignored, the LR indicator will be turned ON, and all final detail and total calculations and output will be performed.

The occurrence of the following message results from reading an unidentifiable input record (i.e., none of the designated record Identification Codes for the file could be found in the input record).

```
     program-name = mix index : IDENT
```

The occurrence of the following message results from reading an input record from a file with matching fields specified that is out of sequence. All records in matching files must be in sequence, either ascending or descending.

```
     program-name = mix index : MSEQ
```

The occurrence of the following message results from reading an input record which is out of sequence, as specified by the entries in columns 15-18 of the Input Specifications.

```
     program-name = mix index : SEQ
```

The occurrence of the following message results from reading an execution-time vector record which is out of sequence, as specified by the SEQUENCE fields in the Extension Specifications.

```
     program-name = mix index : VSEQ
```

PROGRAMMED HALTS.

The following message will be displayed at the end of a program cycle if any of the halt indicators (H0-H9) are set. Each n will either be blank (indicator not set) or a number 0-9 (indicator set).

```
     program-name = mix index : HALT nnnnnnnnnn
```

If the H0, H3, H4, H7, and H9 indicators were set, the message would appear as:

```
     program-name = mix index : HALT 0  34  7 9
```

The program will request a reply from the operator; he may respond with "GO", in which case all halt indicators will be turned OFF and the program will continue; or, he may respond with "STOP", in which case the LR indicator will be turned ON, and all final detail and total calculations and output will be performed.

FORMS_POSITIONING.

The following message will be displayed after all output lines conditioned by the 1P indicator have been printed, if FORMS POSITIONING has been specified in the Control Card (column 41). The program will request a reply from the operator; if he responds with "YES", all 1P lines will be printed again, and the above message will be repeated; if he responds with "NO", normal processing will begin.

```
     program-name = mix index : AGAIN?
```

Printing of the 1P lines may be requested as many time as necessary in order to align the forms properly.

ARITHMETIC_ERRORS.

The occurrence of the following message is a result of an attempt to perform a DIV (divide) operation using a divisor of zero. The program will be automatically discontinued (DS-ED) by the MCP.

```
     program-name = mix index : DIVIDE BY ZERO
```

The occurrence of the following message is a result of an attempt to perform a SQRT (square root) operation on a negative argument. The program will request a reply from the operator; he may respond with "GO", in which case the RESULT FIELD will be set to zero and the program will continue; or, he may respond with "STOP", in which case the LR indicator will be turned ON, and all final detail and total calculations and output will be performed.

```
     program-name = mix index : SQRT
```

SUBSCRIPT_ERRORS.

The occurrence of the following is a result of an attempt to reference an array element that is out-of-bounds (i.e., an index value less than or equal to zero, or greater than the maximum size of the array). The program will automatically be discontinued (DS-ED) by the MCP.

```
     program-name = mix index : INVALID SUBSCRIPT
```

SEQUENCE_ERRORS.

The occurrence of the following message indicates indexed data key out of sequence. Any response will force EOJ. Program data file will have to be re-sorted or program loading file or updating file will require re-working.

```
     program-name = mix index : KEYSEQ
```

APPENDIX F

RPG PROGRAM CYCLE

GENERAL.

During execution, the RPG Program goes through the same basic cycle of operations once for every input record that is processed. This is known as the RPG "program cycle", and is depicted in the flow chart in this appendix. A description of the program flow follows.

PROGRAM CYCLE DESCRIPTION.

The RPG program cycle is described in steps 1 through 33 below and can be traced on Figure F-1.

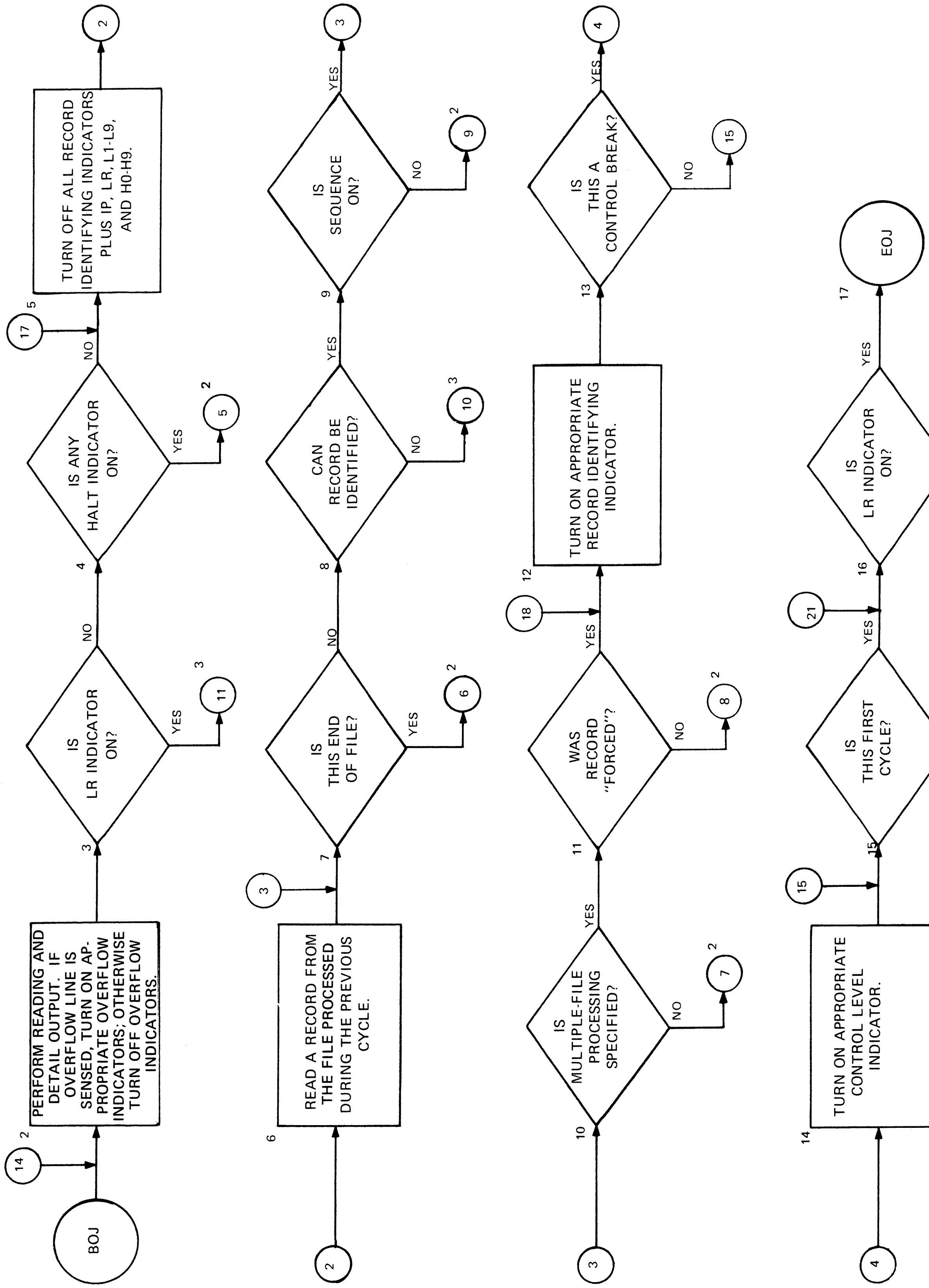

BOJ
14
2
PERFORM READING AND DETAIL OUTPUT. IF OVERFLOW LINE IS SENSED, TURN ON APPROPRIATE OVERFLOW INDICATORS; OTHERWISE TURN OFF OVERFLOW INDICATORS.
3
IS LR INDICATOR ON?
YES
11
3
NO
4
IS ANY HALT INDICATOR ON?
YES
5
2
NO
17
5
TURN OFF ALL RECORD IDENTIFYING INDICATORS PLUS IP, LR, L1-L9, AND H0-H9.
2
2
6
READ A RECORD FROM THE FILE PROCESSED DURING THE PREVIOUS CYCLE.
3
7
IS THIS END OF FILE?
YES
6
2
NO
8
CAN RECORD BE IDENTIFIED?
NO
10
3
YES
9
IS SEQUENCE ON?
NO
9
2
YES
3
3
10
IS MULTIPLE-FILE PROCESSING SPECIFIED?
NO
7
2
YES
11
WAS RECORD "FORCED"?
NO
8
2
YES
18
12
TURN ON APPROPRIATE RECORD IDENTIFYING INDICATOR.
13
IS THIS A CONTROL BREAK?
NO
15
YES
4
4
14
TURN ON APPROPRIATE CONTROL LEVEL INDICATOR.
15
15
IS THIS FIRST CYCLE?
NO
11
3
YES
21
16
IS LR INDICATOR ON?
NO
12
3
YES
17
EOJ

Figure F-1. RPG Program Cycle Flow Chart
(Sheet 1 of 3)

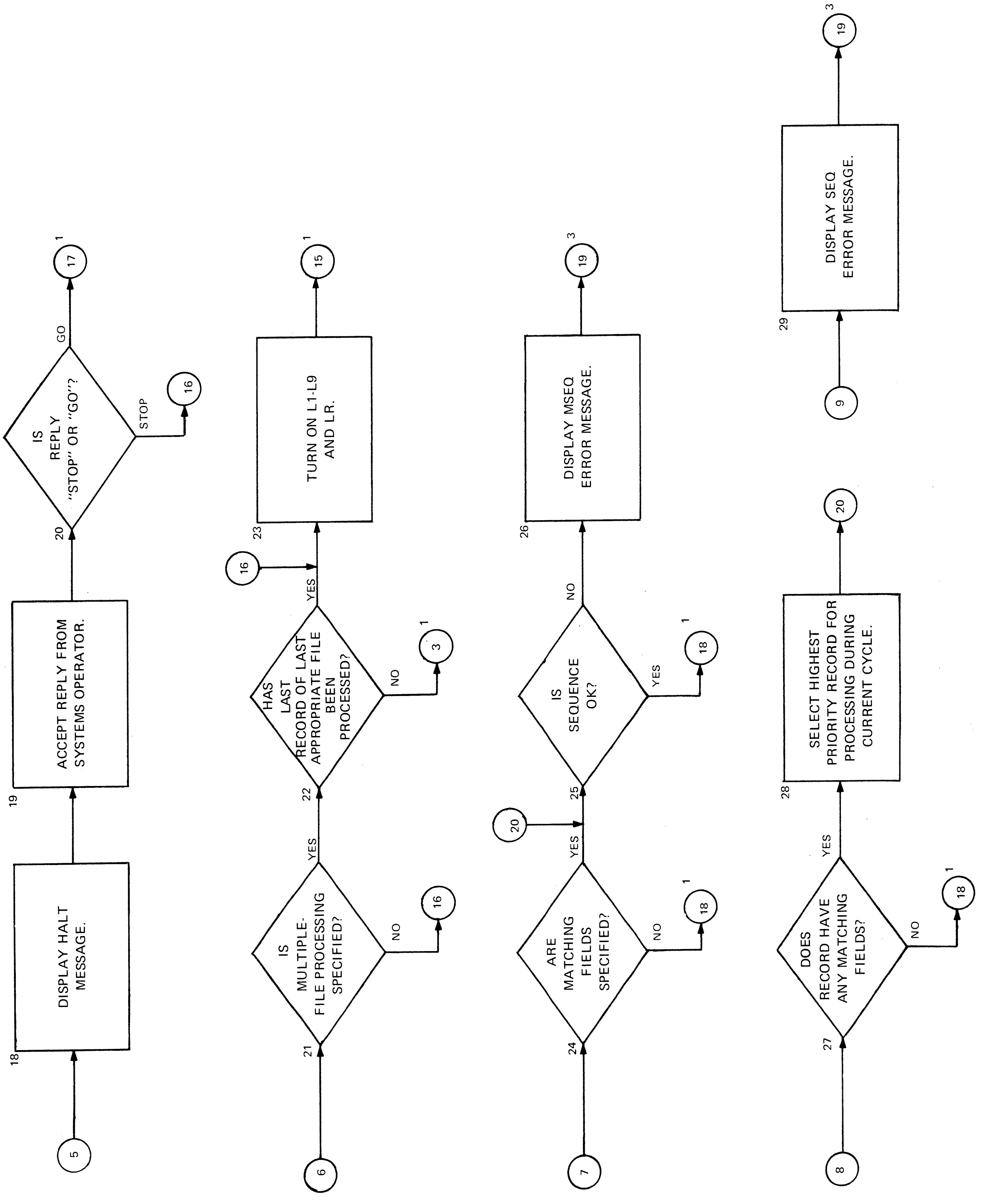

5
18
DISPLAY HALT MESSAGE.
19
ACCEPT REPLY FROM SYSTEMS OPERATOR.
20
IS REPLY "STOP" OR "GO"?
STOP
16
GO
17
1
6
21
IS MULTIPLE-FILE PROCESSING SPECIFIED?
NO
16
YES
22
HAS LAST RECORD OF LAST APPROPRIATE FILE BEEN PROCESSED?
NO
3
1
YES
16
23
TURN ON L1-L9 AND LR.
15
1
7
24
ARE MATCHING FIELDS SPECIFIED?
NO
18
1
YES
20
25
IS SEQUENCE OK?
YES
18
1
NO
26
DISPLAY MSEQ ERROR MESSAGE.
19
3
8
27
DOES RECORD HAVE ANY MATCHING FIELDS?
NO
18
1
YES
28
SELECT HIGHEST PRIORITY RECORD FOR PROCESSING DURING CURRENT CYCLE.
20
9
29
DISPLAY SEQ ERROR MESSAGE.
19
3

Figure F-1. RPG Program Cycle Flow Chart
(Sheet 2 of 3)

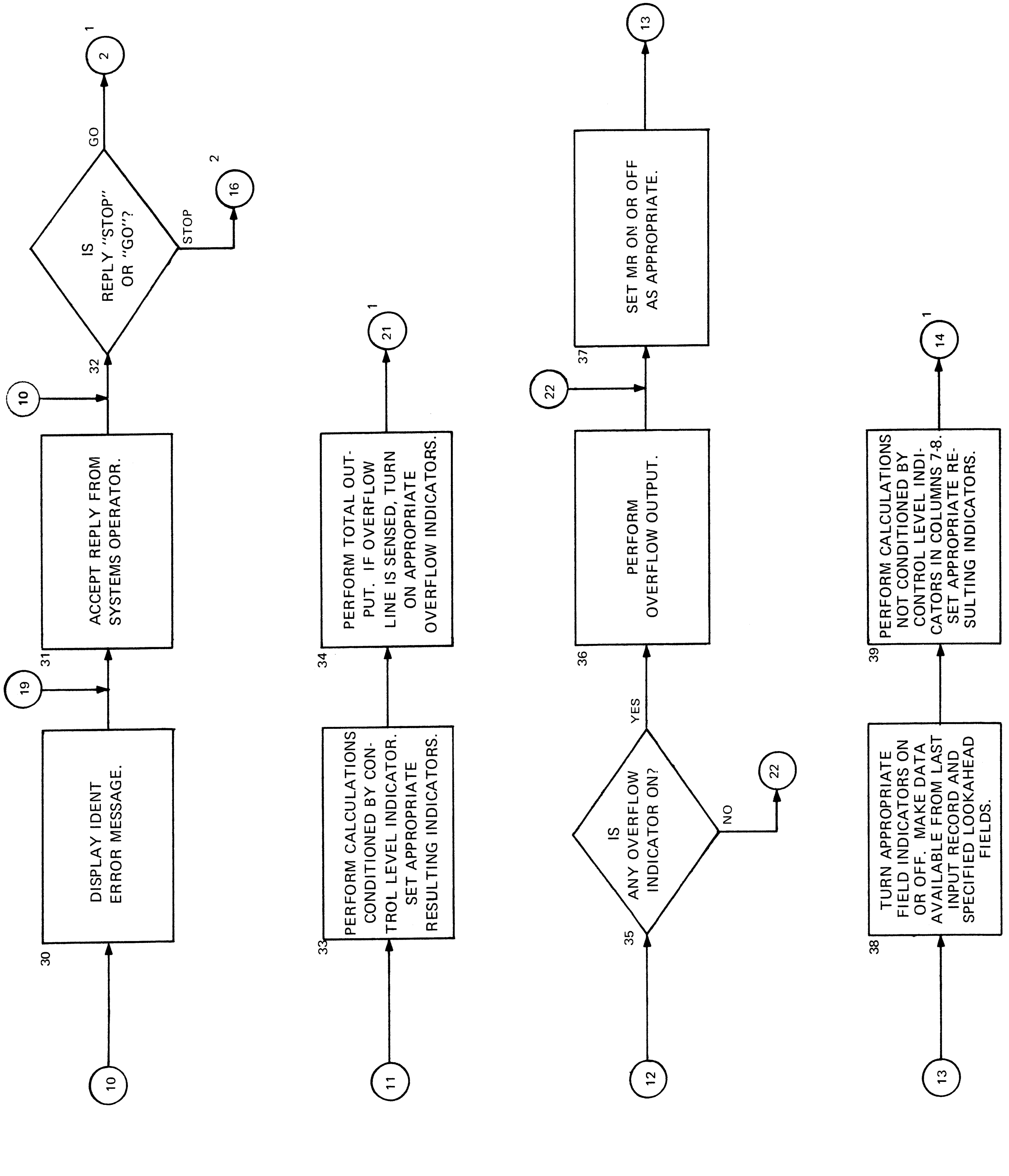

10
30
DISPLAY IDENT ERROR MESSAGE.
19
31
ACCEPT REPLY FROM SYSTEMS OPERATOR.
10
32
IS REPLY "STOP" OR "GO"?
GO
2
1
STOP
16
2
11
33
PERFORM CALCULATIONS CONDITIONED BY CONTROL LEVEL INDICATOR. SET APPROPRIATE RESULTING INDICATORS.
34
PERFORM TOTAL OUTPUT. IF OVERFLOW LINE IS SENSED, TURN ON APPROPRIATE OVERFLOW INDICATORS.
21
1
12
35
IS ANY OVERFLOW INDICATOR ON?
NO
22
YES
36
PERFORM OVERFLOW OUTPUT.
22
37
SET MR ON OR OFF AS APPROPRIATE.
13
13
38
TURN APPROPRIATE FIELD INDICATORS ON OR OFF. MAKE DATA AVAILABLE FROM LAST INPUT RECORD AND SPECIFIED LOOKAHEAD FIELDS.
39
PERFORM CALCULATIONS NOT CONDITIONED BY CONTROL LEVEL INDICATORS IN COLUMNS 7-8. SET APPROPRIATE RESULTING INDICATORS.
14
1

Figure F-1. RPG Program Cycle Flow Chart
(Sheet 3 of 3)

1. Beginning-of-Job.

2. All heading and detail output operations are performed which have their conditions satisfied, except for those conditioned by an overflow indicator. If the overflow line was reached on any printer file during these operations, the assigned overflow indicator will be turned ON; otherwise, it will be turned OFF.

3. If the LR indicator is ON, the program will branch to step 33 (IBM's RPG II syntax only).

4. If any of the halt indicators (H0-H9) are ON, the program will branch to step 18.

5. All record identifying indicators as well as the 1P, LR (other manufacturers RPG syntax only), L1-L9, and H0-H9 indicators are turned OFF.

6. The next input record is read from the file which was processed during the previous cycle. At BOJ, one record is read from each input file (if more than one is specified).

7. If the read in step 6 resulted in an End-of-File condition, the program will branch to step 21.

8. The input record just read is tested to see whether or not the record identification codes can be found. If not, the program will branch to step 32.

9. The input record just read is tested to see whether or not it is in the sequence specified (columns 15-18 of Input Specifications). If not, the program will branch to step 29.

10. If multiple input files are defined, the program will branch to to step 11. Otherwise, it will branch to step 24.

11. If the input record just read was "forced" (see FORCE operation code), the program will branch to step 12. Otherwise, it will branch to step 27.

12. The specified record identifying indicator is turned ON to identify the type of record just read.

13. If a control break (the contents of the CONTROL field in the current record is not equal to the contents of the stored CONTROL field) has occurred, the program will branch to step 14. Otherwise, it will branch to step 15.

14. The specified control level indicator (and all lower control level indicators) is turned ON to identify the control break that has occurred.

15. If this is the first time through the program cycle, the program will branch to step 16. Otherwise, it will branch to step 33.

16. If the LR indicator is ON, the program will branch to step 17. Otherwise, it will branch to step 35.

17. End-of-Job.

18. A message is displayed to inform the systems operator of the halt indicator settings.

19. The program accepts a message from the operator specifying whether to terminate the job or to continue program execution.

20. If the reply in step 19 is STOP, the program will branch to step 23. If the reply is GO, it will branch to step 5.

21. If multiple input files are defined, the program will branch to step 22. Otherwise, it will branch to step 23.

22. If processing is completed for all files specifying End-of-File checking, the program will branch to step 23. Otherwise, it will branch to step 7.

23. The LR and L1-L9 indicators are turned ON, and the program branches to step 15.

24. If matching fields are specified for the input record, the program will branch to step 25. Otherwise, it will branch to step 12.

25. If the sequence of the MATCH fields is correct the program will branch to step 12.

26. The MSEQ error message is displayed to inform the operator of a matching fields sequence error. The program then branches to step 31.

27. If the input record has no matching fields, the program will branch to step 12.

28. The file to be processed next is selected, and the program branches to step 25.

29. The SEQ error message is displayed to inform the operator of an input record sequence error. The program then branches to step 31.

30. The IDENT error message is displayed to inform the operator that the last input record cannot be identified.

31. The program accepts a message from the operator specifying whether to terminate the job or to continue program execution.

32. If the reply in step 31 is STOP, the program will branch to step 23. If the reply is GO, it will branch to step 6.

33. All calculations conditioned by a control level indicator in the CONTROL LEVEL field are performed (all specified conditions must be met, of course). Resulting indicators are set ON or OFF in each case, as appropriate.

34. All total output operations are performed which have their conditions satisfied, except for those conditioned by an overflow indicator. If the overflow line was reached on any printer file during these operations, the assigned overflow indicator will be turned ON. The program then branches to step 16.

35. If no overflow indicators are ON, the program will branch to step 37.

36. All specified overflow output is performed (total output, then detail output).

37. If the record selected for processing is a matching record and multiple file processing is specified, the MR indicator will be turned ON. Otherwise, it will be turned OFF.

38. Field indicators are turned ON or OFF as specified, and data from the last record selected for processing is made available. Also, specified data from look-ahead fields is made available.

39. All calculations not conditioned by a control level indicator in the CONTROL LEVEL field are performed (all specified conditions must be met, of course). Resulting indicators are set ON or OFF in each case, as appropriate. The program then branches to step 2.

APPENDIX G

IBM's RPG DIFFERENCES AND CONSIDERATIONS

GENERAL.

Since Burroughs RPG is intended to support both IBM's RPG and RPG II, some method of resolving the conflicts between the two dialects must be available. The SOURCE INPUT DIALECT field of the Control Card is used to notify the compiler of the type of input to expect for processing.

If IBM's RPG source input is specified, certain specifications are treated differently by the compiler. These differences are discussed in the paragraphs that follow.

The fact that IBM's RPG source input is specified does not limit the programmer to the strict regulations of the RPG language. It merely allows the compiler to choose the characteristics of one language over those of the other when faced with conflicting programming methods. Therefore, both languages can be incorporated in RPG without any problem.

FILE DESCRIPTION SPECIFICATIONS.

The File Description Specifications differences are described below.

DEVICE (COLUMNS 40-46).
The Burroughs RPG compiler will accept a number of device names in this field, regardless of the type of source input being processed. Each device will be equated to the corresponding Burroughs RPG device name by the compiler, as shown in Table G-1.

NOTE

If the OVERFLOW INDICATOR field is left blank, a PRINTR2 or PRINTUF entry in the DEVICE field will result in a default setting of OV for the overflow indicator; all other PRINTER file DEVICE entries will result in a default setting of OF.

Table G-1

Device Name Conversion

Device name accepted	Device name converted to
READER READ01 CCPRI CCSEC	READER

Table G-1 (cont)

Device Name Conversion

Device name accepted	Device name converted to
MFCM1 MFCM2 MFCU1 MFCU2 READ20 READ40 READ42 CRP CRP20	READER/PUNCH
PUNCH PUNCH20 PUNCH42 ROWPUNCH	PUNCH
PRINTER PRINTR2 PRINTUF PRINTLF PRINT16 PRINT48 PRINT63	PRINTER
TAPE	TAPE
DISK DISK11 DISK11F	DISK
SPO CONSOLE INQUIPT	SPO

CCPRI, CCSEC, and ROWPUNCH are accepted by the RPG/COBOL Translator only.

EXTENSION SPECIFICATIONS.

Differences in the Extension Specifications are described below.

CHAINING FIELD NUMBER (COLUMNS 9-10).
This field is used to assign a chaining field code number to associate a chained file with the chaining file that provides the key fields for accessing the chained file. This code number must be the same as the one entered in the MATCHING FIELDS OR CHAINING FIELDS field of the Input Specifications. Each CHAINING field to be specified, whether in one or more than one chaining file, must be designated on a separate line in the Extension Specifications. Valid entries for this field are:

Entry	Definition
Blank	Vector description.
C1-C9	Chaining field code.

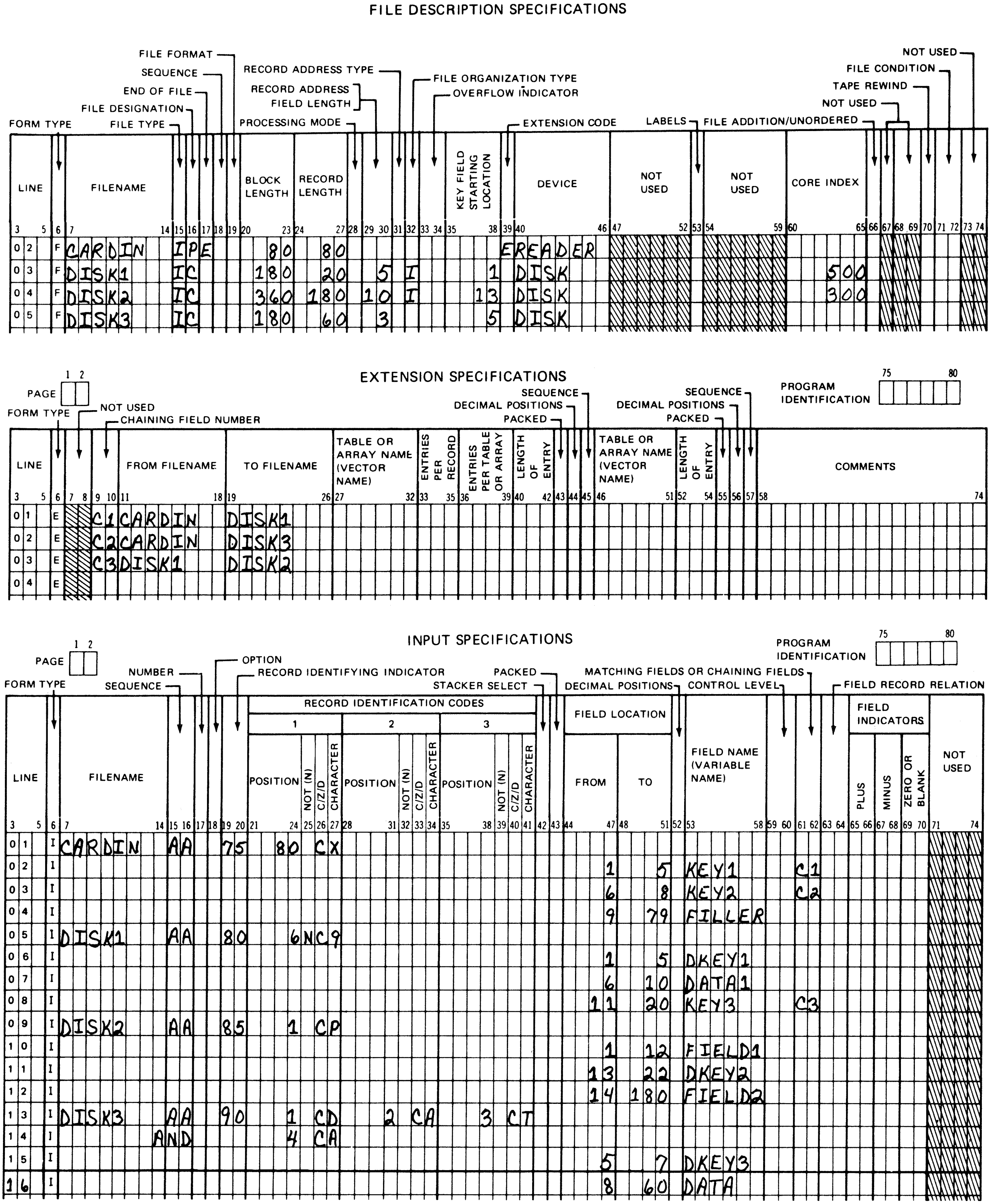
If an entry is made in this field, the FROM FILENAME field must contain the name of the chaining file, and the TO FILENAME field must contain the name of the chained file. See Figure G-1 for an example of chaining fields.

INPUT SPECIFICATIONS.

Differences in Input Specifications are described in the text that follows.

MATCHING FIELDS OR CHAINING FIELDS (COLUMNS 61-62).
This field is used to specify matching fields and sequence checking (refer to section 7) or CHAINING fields. If a CHAINING field code (C1-C9) is entered in this field, the same code must have been specified for the same file (the chaining file) in the Extension Specifications. Valid entries for this field are:

Entry	Definition
Blank	No chaining or matching.
M1-M9	MATCHING fields and sequence checking (refer to section 7).
C1-C9	CHAINING fields.



FILE DESCRIPTION SPECIFICATIONS

LINE	FORM TYPE	FILENAME	FILE TYPE	FILE DESIGNATION	END OF FILE	BLOCK LENGTH	RECORD LENGTH	RECORD ADDRESS FIELD LENGTH	FILE ORGANIZATION TYPE	KEY FIELD STARTING LOCATION	EXTENSION CODE	DEVICE	CORE INDEX
02	F	CARDIN	I	P	E	80	80				E	READER	
03	F	DISK1	I	C		180	20	5	I	1		DISK	500
04	F	DISK2	I	C		360	180	10	I	13		DISK	300
05	F	DISK3	I	C		180	60	3		5		DISK	

EXTENSION SPECIFICATIONS

LINE	FORM TYPE	CHAINING FIELD NUMBER	FROM FILENAME	TO FILENAME
01	E	C1	CARDIN	DISK1
02	E	C2	CARDIN	DISK3
03	E	C3	DISK1	DISK2
04	E			

INPUT SPECIFICATIONS

LINE	FORM TYPE	FILENAME	SEQUENCE	RECORD IDENTIFYING INDICATOR	1 POSITION	1 NOT (N)	1 C/Z/D	1 CHARACTER	2 POSITION	2 C/Z/D	2 CHARACTER	3 POSITION	3 C/Z/D	3 CHARACTER	FROM	TO	FIELD NAME	MATCHING FIELDS OR CHAINING FIELDS
01	I	CARDIN	AA	75	80		C	X										
02	I														1	5	KEY1	C1
03	I														6	8	KEY2	C2
04	I														9	79	FILLER	
05	I	DISK1	AA	80	6	N	C	9										
06	I														1	5	DKEY1	
07	I														6	10	DATA1	
08	I														11	20	KEY3	C3
09	I	DISK2	AA	85	1		C	P										
10	I														1	12	FIELD1	
11	I														13	22	DKEY2	
12	I														14	180	FIELD2	
13	I	DISK3	AA	90	1		C	D	2	C	A	3	C	T				
14	I		AND		4		C	A										
15	I														5	7	DKEY3	
16	I														8	60	DATA	

Figure G-1. Chaining Fields

As many as nine CHAINING FIELDS may be specified for a record in a chaining file, each of which is associated with a different chained file. Also, each CHAINING FIELD may chain to more than one chained file, in which case the field must be defined as many times as it is used (each with a different CHAINING FIELD code number assigned). Additionally, two or more chaining files may chain to the same chained file. In this case, they must all have the same CHAINING FIELD code number assigned.

Sequence-checking may also be specified for a chaining file by assigning an M1-M9 code to the proper field. If the field to be used for chaining is also to be sequence-checked, it must be defined a second time (with a different field name).

Split CHAINING FIELDs within one record are allowable, and each must have the same CHAINING FIELD code number assigned. They are combined in the order specified to form one CHAINING FIELD during execution.

See Figure G-1 for an example of CHAINING FIELDs.

OUTPUT SPECIFICATIONS.

Differences in Output Specifications are described in the text that follows.

SKIP (COLUMNS 19-22).
Forms skipping in IBM's RPG is related directly to the operation of the printer carriage control tape. The SKIP field is divided into two subfields such that skipping both BEFORE and AFTER printing may be specified. Valid entries for this field are 1 thru 11, which designate skip to channel 1 thru 11.

Skipping is done according to the format punched in the carriage control tape, and sensing a channel 12 punch will cause the proper overflow indicator to be turned ON.

Line Counter Specifications are not required. If, however, the printer output is going to be written to backup during execution, Line Counter Specifications must be included if overflow output is ever to occur. They are used to associate line numbers with the channel numbers referenced, so that the program will be able to determine when the overflow line has been reached.

MISCELLANEOUS DIFFERENCES.

Some other differences arise in the code generated by the compiler when IBM's RPG source input is specified on the Control Card. These are in the BLANK AFTER specification, the LR indicator, and EDIT WORDS.

BLANK AFTER
If an indicator is assigned to a field to test for ZERO or BLANK in the Input Specifications (FIELD INDICATORS) or Calculation Specifications (RESULTING INDICATORS) and the same field is used in the Output Specifications with a BLANK AFTER designation, that indicator will be turned ON after the field is blanked during the output operation.

LR INDICATOR.
IBM's RPG II tests the LR indicator after every detail and total output, whereas IBM's RPG only tests it after total output. Also, IBM's RPG II turns OFF the LR indicator once at the beginning of processing, whereas IBM's RPG turns it OFF after each detail output. Refer to appendix F for a complete description of the Burroughs RPG Program Cycle.

EDIT WORDS.
In IBM's RPG, suppression of leading zeros in at least one position (the high-order position) of the field to be edited cannot be prevented using edit words. In RPG II, an extra space can be left in the edit word if the first character of the edit word is a zero in order to force significance in the high-order position. Consider the following example:

Source Data	IBM's RPG		IBM's RPG II	
	Edit Word	Result	Edit Word	Result
00012345	'0bbbbb.bb'	b00123.45	'0bbbbbb.bb'	000123.45
000000	'0bbbbb'	b00000	'0bbbbbb'	000000

IBM's RPG maximum alpha field length is 256 characters whereas B 1700 RPG maximum length is 511 characters.

IBM uses standard right signed numeric fields, whereas B 1700 RPG uses standard left sign.

IBM's RPG extracts from the header record of existing files the correct blocking factors of that file and ignores the program entry of BLOCK LENGTH and RECORD LENGTH. Burroughs RPG uses this same technique of extracting the BLOCK LENGTH and RECORD LENGTH from the disk file header but provides the additional capability of accepting the BLOCK LENGTH and RECORD LENGTH as specified in the program file declarations by using the $REFORM option as explained in section 10.

IBM's SYSTEM 3 Standard plus sign is 'F' Right Sign, the 360/20 Standard plus sign is 'C' Right Sign and the Burroughs Standard plus sign is 'C' Left Sign.

Numeric Fields must be edited if the C Zone Punch is not wanted on output, otherwise a positive numeric field, unedited, will appear as
$\frac{C}{1}$ = A in the field C1F0F1F3 will appear A013 on printer output. Editing will assign a F to the leftmost character, thus making the output appear as 1013.

APPENDIX H

RPG-TO-COBOL TRANSLATION

To be specified at a later date.

APPENDIX I

VECTORS

GENERAL.

Tables and arrays are logical configurations of data elements having similar characteristics. Burroughs RPG, unlike other manufactures RPG and RPG II, makes very little distinction between tables and arrays and, hence, refers to them both as "vectors". Where differences do arise in their characteristics, they will be referred to individually as "tables" or "arrays".

Each element of a vector must be of the same length and data type (numeric or alphanumeric). If numeric, each element must have the same number of decimal positions.

TABLE AND ARRAY DIFFERENCES.

As was stated above, almost no difference exists in Burroughs RPG between tables and arrays. Thus, the more general concept of vectors has been adopted. Those differences that do exist are very minor, and are pointed out in the paragraphs that follow.

Every vector to be used by the program must be given a name. All table names must begin with the letters TAB, whereas array names may begin with any alphabetic character.

Both tables and arrays may use indices to access specific elements within them. If, however, a table name appears without an index assigned, it refers to the last item referenced in the table, as stored in the "hold area". The hold area is set by the LOKUP operation, as the result of a successful search (refer to LOOKUP OPERATIONS, appendix C). If an array name appears without an index assigned, it refers to the entire array. Such a reference specifies that the designated operation be performed repetitively on each element of the array.

The above differences are the only ones that arise between tables and arrays, and exist mainly for compatibility between Burroughs RPG and other manufacturers RPG and RPG II. For all practical purposes, tables and arrays can be handled in the same way in Burroughs RPG.

TYPES OF VECTORS.

Vectors can be distinguished according to the time they are loaded with data, which may occur at three distinct times: during compilation, at the beginning of program execution, or during program execution. Data elements in a vector may, however, be altered at any time during program execution, regardless of the time the vector was loaded.

COMPILE-TIME VECTORS.

For a compile-time vector load, the data to be loaded is read into the program storage area reserved by the entries in the Extension Specifications at the same time that the source program is compiled. Physically, the data is placed in a card file called "RPG/VECTOR", which immediately follows the source program deck.

For a compile-time vector load, short vectors (those which are only partially full) are not allowed. Each vector file must occur in the order in which it was specified, and must contain exactly the number of records necessary to fill it with data. No separators are used to delimit the end of one vector file and the beginning of the next; the compiler reads input records until one vector is full, and then proceeds to fill the next one in order.

> NOTE
> The last vector file entered may be short; the compiler will fill the unused portions of that vector with zeros or blanks (for numeric or alphanumeric vectors, respectively).

PRE-EXECUTION-TIME VECTORS.
For a pre-execution-time vector load, the data to be loaded is read into the program storage area reserved by the entries in the Extension Specifications at the beginning of object program execution, before the normal operations in the program cycle begin. The data for each vector is placed in a separate file, identified by the names assigned in the File Description Specifications.

Short vectors are permissible with pre-execution-time vector loads, because each vector file is delimited by MCP control cards. Any portions of the vector which are not loaded will be set to zeros or blanks (for numeric or alphanumeric vectors, respectively).

DYNAMIC/EXECUTION-TIME VECTORS.
Dynamic vectors are loaded during program execution through entries in the Input Specifications or Calculation Specifications. Certain fields of input records or the results of calculation operations may be used to load the elements of a dymamic vector. Such loading, unlike the automatic loading of compile-time and pre-execution-time vectors, is completely under programmatic control.

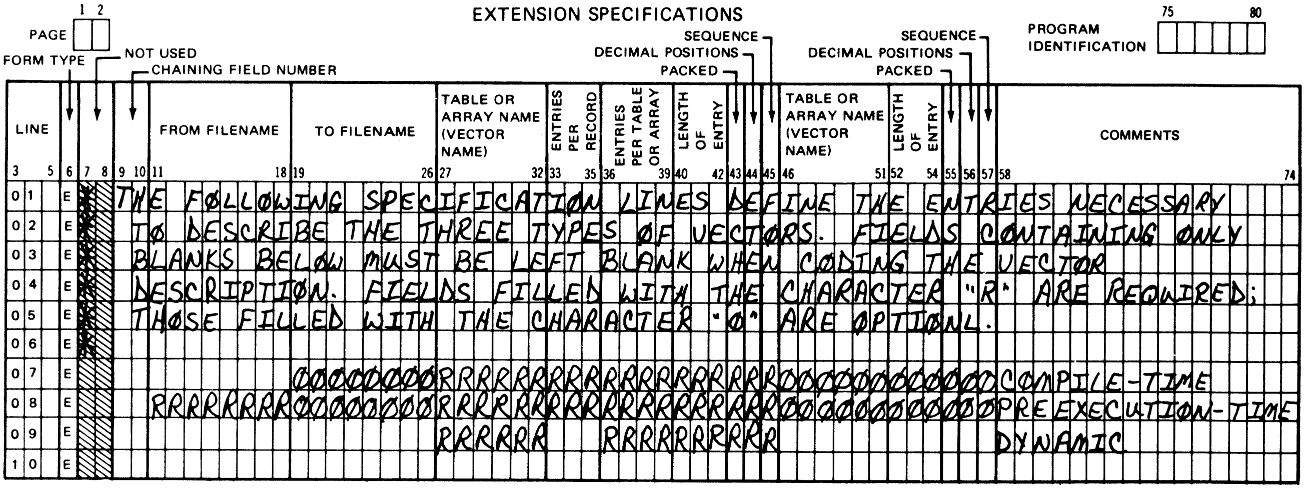
All vectors are able to be altered during program execution, regardless of when they were loaded initially. Because of this, all vectors may be considered to have "dynamic" characteristics.

VECTOR DECLARATIONS.
All vectors in a program must be described on the Extension Specifications sheet. Certain entries are required for all types of vectors, regardless of the time at which they will be loaded. Columns 27-45 must be completed for each vector (columns 46-57 if the vector is loaded in alternating format with another vector). These columns specify the name assigned to identify the vector (VECTOR NAME), the number of vector elements occurring on each input record (ENTRIES PER RECORD), the size of the vector (ENTRIES PER TABLE OR ARRAY and LENGTH OF ENTRY), whether the input data is in packed decimal format (PACKED), the number of decimal positions in each entry (DECIMAL POSITIONS), and the order in which the elements are sequenced (SEQUENCE).

Further entries are required if the vector is to be loaded at execution time (FROM FILENAME), or if the vector is to be written to an output file at End-of-Job (TO FILENAME). Any files named in these fields must also be described on the File Description Specifications

NOTE
For dynamic vectors, the ENTRIES PER RECORD, FROM FILENAME, and TO FILENAME fields must be left blank. Also, a second vector in alternating format (columns 46-57) cannot be declared.

See Figure I-1 for the entries necessary to describe a vector in the Extension Specifications.



EXTENSION SPECIFICATIONS

PAGE (1 2) □□ PROGRAM IDENTIFICATION (75–80) □□□□□□

LINE (3–5)	FORM TYPE (6)	NOT USED (7 8)	CHAINING FIELD NUMBER (9 10)	FROM FILENAME (11–18)	TO FILENAME (19–26)	TABLE OR ARRAY NAME (VECTOR NAME) (27–32)	ENTRIES PER RECORD (33–35)	ENTRIES PER TABLE OR ARRAY (36–39)	LENGTH OF ENTRY (40–42)	PACKED (43)	DECIMAL POSITIONS (44)	SEQUENCE (45)	TABLE OR ARRAY NAME (VECTOR NAME) (46–51)	LENGTH OF ENTRY (52–54)	PACKED (55)	DECIMAL POSITIONS (56)	SEQUENCE (57)	COMMENTS (58–74)
01	E	*	THE FØLLØWING SPECIFICATIØN LINES DEFINE THE ENTRIES NECESSARY															
02	E	*	TØ DESCRIBE THE THREE TYPES ØF VECTØRS. FIELDS CØNTAINING ØNLY															
03	E	*	BLANKS BELØW MUST BE LEFT BLANK WHEN CØDING THE VECTØR															
04	E	*	DESCRIPTIØN. FIELDS FILLED WITH THE CHARACTER "R" ARE REQUIRED;															
05	E	*	THØSE FILLED WITH THE CHARACTER "Ø" ARE ØPTIØNL.															
06	E	*																
07	E				ØØØØØØØØ	RRRRRR	RRR	RRRR	RRR	R	R	R	ØØØØØØ	ØØØ	Ø	Ø	Ø	COMPILE-TIME
08	E			RRRRRRRR	ØØØØØØØØ	RRRRRR	RRR	RRRR	RRR	R	R	R	ØØØØØØ	ØØØ	Ø	Ø	Ø	PRE EXECUTIØN-TIME
09	E					RRRRRR		RRRR	RRR	R	R	R						DYNAMIC
10	E																	

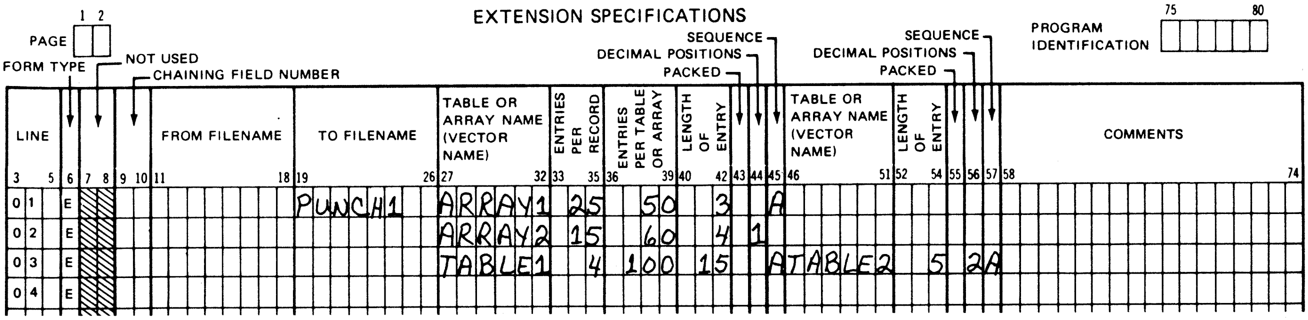
Figure I-1. Vector Declarations

VECTOR LOADING.

The following paragraphs describe loading for compile-time, pre-execution-time and dynamic vectors.

COMPILE-TIME VECTOR LOAD.
To load a compile-time vector, all that is necessary is to include the data cards for the vector in the proper order in the file "RPG/VECTOR". The compiler will automatically read this file after the source program has been read, and will load the data into the vectors declared.

If more than one compile-time vector has been declared, care must be exercised in setting up the data deck. The compiler expects data for the vectors to be entered in the same order in which they were declared in the Extension Specifications form. The comiler reads data cards and stores the elements in the first vector until it is full, then proceeds to do the same for the second vector, and so forth. If more or fewer records are present than are necessary to exactly fill a vector, data can be placed into the wrong vector. See Figure I-2 for an example of a compile-time vector load declaration.



EXTENSION SPECIFICATIONS

PAGE 1 2 □□ PROGRAM IDENTIFICATION 75 80 □□□□□□

LINE 3 5	FORM TYPE 6	NOT USED 7 8	CHAINING FIELD NUMBER 9 10	FROM FILENAME 11 18	TO FILENAME 19 26	TABLE OR ARRAY NAME (VECTOR NAME) 27 32	ENTRIES PER RECORD 33 35	ENTRIES PER TABLE OR ARRAY 36 39	LENGTH OF ENTRY 40 42	PACKED 43	DECIMAL POSITIONS 44	SEQUENCE 45	TABLE OR ARRAY NAME (VECTOR NAME) 46 51	LENGTH OF ENTRY 52 54	PACKED 55	DECIMAL POSITIONS 56	SEQUENCE 57	COMMENTS 58 74
01	E	▨			PUNCH1	ARRAY1	25	50	3			A						
02	E	▨				ARRAY2	15	60	4		1							
03	E	▨				TABLE1	4	100	15			A	TABLE2	5		2	A	
04	E	▨																

Figure I-2. Compile-Time Vector Load

PRE-EXECUTION-TIME VECTOR LOAD.
To load a pre-execution-time vector, the data to be loaded must be placed in the file described in the File Description Specifications to which the vector is assigned (FROM FILENAME). Data is read in at the beginning of program execution and placed in the vector until either End-of-File is reached or the vector is full. In the first case, the unused portion of the vector will be set to zero or blank (numeric or alphanumeric vector, respectively); in the second case, any remaining records in the file are ignored.

If more than one vector (not in alternating format) is assigned to a single file, some special considerations must be taken into account (see Figure I-3). Vectors are still loaded in the same order that they are specified in the Extension Specifications, so that files will be opened, read, and closed as necessary to load the designated vectors. If two vectors are assigned to the same table file, and no other pre-execution-time vector declaration comes between them, the data for both must be in the same card file. No separators are allowed between the data decks, so that the same restrictions as compile-time vector loads are imposed.

DYNAMIC/EXECUTION-TIME VECTOR LOAD.
To load a dynamic vector, the data elements may be obtained from fields within input records or from the result of operations in the Calculation Specifications.

INPUT SPECIFICATIONS LOAD. Fields within input records may contain data for vector loading. This is done by assigning a vector name with a numeric literal index or an array name without an index as a field name within an input record description (see Figure I-4). If the field name (VARIABLE NAME) designates a single element of the vector, the input field will be placed into the vector element when the record is selected. If the field name designates an entire array (no index assigned), the input field length must be an integral multiple of the element size (LENGTH OF ENTRY) and equal to or less than the total size of the array. If the input field is less than the size of the array, the elements not referenced will not be affected.

FILE DESCRIPTION SPECIFICATIONS

LINE 3-5	FORM TYPE 6	FILENAME 7-14	FILE TYPE 15	FILE DESIGNATION 16	END OF FILE 17	SEQUENCE 18	FILE FORMAT 19	BLOCK LENGTH 20-23	RECORD LENGTH 24-27	PROCESSING MODE 28	RECORD ADDRESS FIELD LENGTH 29-30	RECORD ADDRESS TYPE 31	FILE ORGANIZATION TYPE 32	OVERFLOW INDICATOR 33-34	KEY FIELD STARTING LOCATION 35-38	EXTENSION CODE 39	DEVICE 40-46	NOT USED 47-52	LABELS 53	NOT USED 54-59	CORE INDEX 60-65	FILE ADDITION/UNORDERED 66	NOT USED 67-69	TAPE REWIND 70	FILE CONDITION 71-72	NOT USED 73-74
02	F	FILE1	I	T			F	80	80							E	READER	▨		▨			▨			▨
03	F	FILE2	I	T			F	80	80							E	READER	▨		▨			▨			▨
04	F	FILE3	Ø				F	80	80								PUNCH	▨		▨			▨			▨

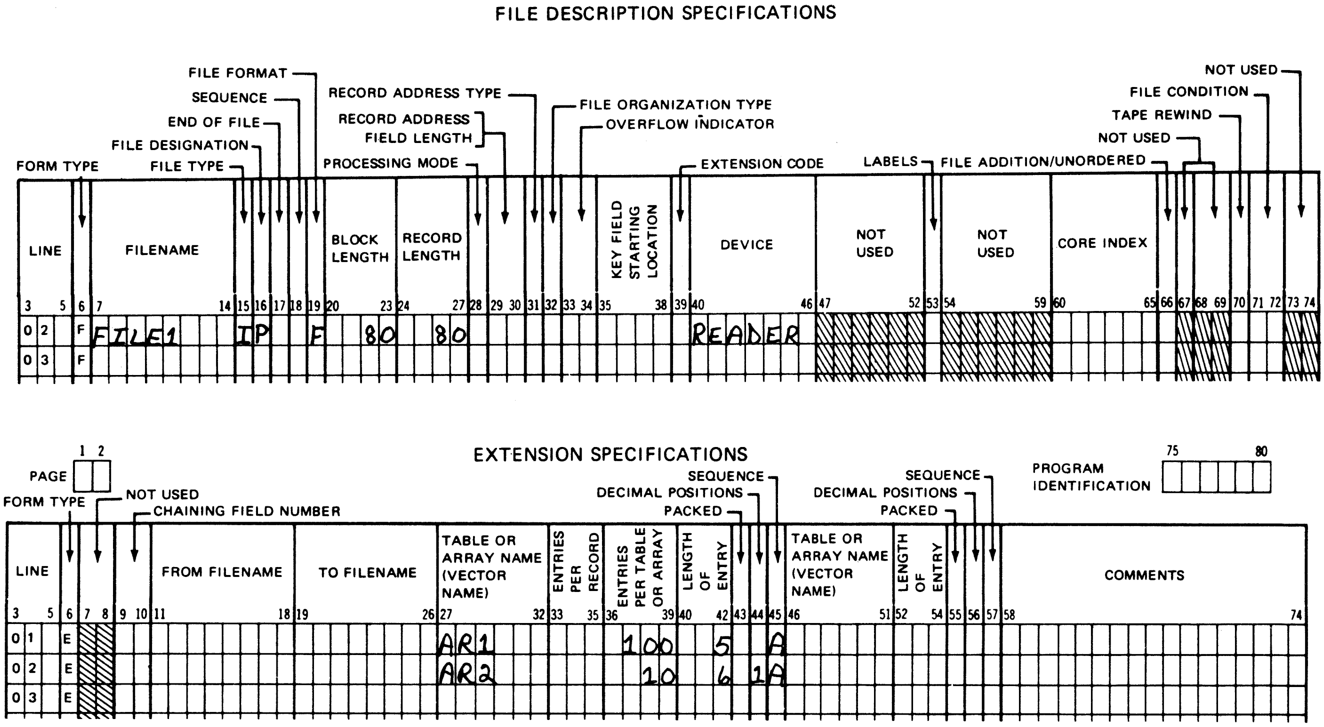
EXTENSION SPECIFICATIONS

PAGE □□ (1 2)

PROGRAM IDENTIFICATION □□□□□□ (75-80)

LINE 3-5	FORM TYPE 6	NOT USED 7-8	CHAINING FIELD NUMBER 9-10	FROM FILENAME 11-18	TO FILENAME 19-26	TABLE OR ARRAY NAME (VECTOR NAME) 27-32	ENTRIES PER RECORD 33-35	ENTRIES PER TABLE OR ARRAY 36-39	LENGTH OF ENTRY 40-42	PACKED 43	DECIMAL POSITIONS 44	SEQUENCE 45	TABLE OR ARRAY NAME (VECTOR NAME) 46-51	LENGTH OF ENTRY 52-54	PACKED 55	DECIMAL POSITIONS 56	SEQUENCE 57	COMMENTS 58-74
01	E	▨		FILE1		TABLE1	3	100	18									
02	E	▨		FILE2		ARRAY1	20	80	4		2	D						
03	E	▨		FILE1	FILE3	ARRAY2	5	15	10			A	ARRAY3	5		1	A	
04	E	▨		FILE1		TABLE2	10	100	6	P	1							

Figure I-3. Pre-Execution-time Vector Load



FILE DESCRIPTION SPECIFICATIONS

LINE (3-5)	FORM TYPE (6)	FILENAME (7-14)	FILE TYPE (15)	FILE DESIGNATION (16)	END OF FILE (17)	SEQUENCE (18)	FILE FORMAT (19)	BLOCK LENGTH (20-23)	RECORD LENGTH (24-27)	PROCESSING MODE (28)	RECORD ADDRESS FIELD LENGTH (29-30)	RECORD ADDRESS TYPE (31)	FILE ORGANIZATION TYPE (32)	OVERFLOW INDICATOR (33-34)	KEY FIELD STARTING LOCATION (35-38)	EXTENSION CODE (39)	DEVICE (40-46)	NOT USED (47-52)	LABELS (53)	NOT USED (54-59)	CORE INDEX (60-65)	FILE ADDITION/UNORDERED (66)	NOT USED (67-69)	TAPE REWIND (70)	FILE CONDITION (71-72)	NOT USED (73-74)
02	F	FILE1	I	P			F	80	80								READER									
03	F																									

PAGE (1-2) □□

EXTENSION SPECIFICATIONS

PROGRAM IDENTIFICATION (75-80) □□□□□□

LINE (3-5)	FORM TYPE (6)	NOT USED (7-8)	CHAINING FIELD NUMBER (9-10)	FROM FILENAME (11-18)	TO FILENAME (19-26)	TABLE OR ARRAY NAME (VECTOR NAME) (27-32)	ENTRIES PER RECORD (33-35)	ENTRIES PER TABLE OR ARRAY (36-39)	LENGTH OF ENTRY (40-42)	PACKED (43)	DECIMAL POSITIONS (44)	SEQUENCE (45)	TABLE OR ARRAY NAME (VECTOR NAME) (46-51)	LENGTH OF ENTRY (52-54)	PACKED (55)	DECIMAL POSITIONS (56)	SEQUENCE (57)	COMMENTS (58-74)
01	E					AR1		100	5			A						
02	E					AR2		10	6		1	A						
03	E																	

Figure I-4. Dynamic Vector Load
(Sheet 1 of 2)

INPUT SPECIFICATIONS

PAGE 1 2 · PROGRAM IDENTIFICATION 75–80

FORM TYPE · FILENAME · SEQUENCE · NUMBER · OPTION · RECORD IDENTIFYING INDICATOR · RECORD IDENTIFICATION CODES (1, 2, 3: POSITION, NOT (N), C/Z/D, CHARACTER) · STACKER SELECT · PACKED · FIELD LOCATION (FROM, TO) · DECIMAL POSITIONS · FIELD NAME (VARIABLE NAME) · CONTROL LEVEL · MATCHING FIELDS OR CHAINING FIELDS · FIELD RECORD RELATION · FIELD INDICATORS (PLUS, MINUS, ZERO OR BLANK) · NOT USED

```
LINE
3   5 6 7       14 15 16 17 18 19 20 21  24 25 26 27 ...  44  47 48  51 52 53    58
0 1   I FILE1      AA          01        1     C  T
0 2   I                                                         2      6     AR1,1
0 3   I                                                         8     67     AR2
0 4   I*
0 5   I* WHEN A RECØRD FRØM FILE1 WITH A "T" IN PØSITIØN 1 IS READ,
0 6   I*   THE 01 INDICATØR WILL BE TURNED ØN. THE 5 CHARACTERS
0 7   I*   BEGINNING IN PØSITIØN 2 WILL BE LØADED INTØ THE FIRST ELEMENT
0 8   I*   ØF THE ARRAY AR1. THE 60 CHARACTERS BEGINNING IN PØSITIØN 8
0 9   I*   WILL BE LØADED INTØ THE ENTIRE ARRAY AR2.
1 0   I
```

CALCULATION SPECIFICATIONS

PAGE 1 2 · PROGRAM IDENTIFICATION 75–80

FORM TYPE · CONTROL LEVEL · INDICATORS (NOT, AND NOT, AND NOT) · FACTOR 1 · OPERATION · FACTOR 2 · RESULT FIELD · FIELD LENGTH · DECIMAL POSITIONS · HALF ADJUST · Resulting Indicators: Arithmetic (Plus, Minus, Zero); Compare (High 1>2, Low 1<2, Equal 1 = 2); Lookup Table (Factor 2) is (High, Low, Equal) · COMMENTS

```
LINE
3   5 6 7 8 9     17 18        27 28     32 33        42 43       48 49 51 52 53 54 55 56 57 58 59 60
0 1   C     01        AR1,1         ADD       AR2          AR2
0 2   C* ADD THE FIRST ELEMENT ØF AR1 TØ EVERY ELEMENT ØF AR2
0 3   C     01                      MØVE      0            IX         30
0 4   C* DEFINE THE FIELD "IX" (3 LØNG, 0 DECIMALS) & SET IT TØ ZERØ
0 5   C     01        LØØP          TAG
0 6   C* DEFINE THE LABEL "LØØP"
0 7   C     01        1             ADD       IX           IX
0 8   C* INCREMENT IX BY 1
0 9   C     01        10            COMP      IX                                  02
1 0   C* IF IX > 10 TURN INDICATØR 02
1 1   C     01N02     AR2,IX        DIV       3.1415       AR2,IX
1 2   C* IF THE 02 INDICATØR IS ØFF, DIVIDE 3.1415 INTØ THE ELEMENT ØF
1 3   C* AR2 PØINTED TØ BY IX.
1 4   C     01N02                   GØTØ      LØØP
1 5   C* IF THE 02 INDICATØR IS ØFF, BRANCH TØ "LØØP"
16    C               'EØJ'         DSPLY
17    C* DISPLAY "EØJ" MESSAGE
      C
```

Figure I-4. Dynamic Vector Load
(Sheet 2 of 2)

CALCULATION SPECIFICATIONS LOAD. Any operation which specifies a vector (with or without an index) as the RESULT FIELD will cause the designated vector element (or entire array, if no index is specified) to be loaded with the result of the operation. See Figure I-4 for an example of a load via Calculation Specifications. Refer to appendix C for the operations which may be specified for vectors.

VECTOR OUTPUT.
Compile-time and pre-execution-time vectors may be written to an output device at End-of-Job by specifying an output file in the TO FILENAME field of the Extension Specifications. This vector output is performed automatically after all processing has been completed. Vector records will be in the format specified by the Extension Specifications.

NOTE

> Dynamic vectors cannot specify a TO FILENAME, and thus cannot be automatically written out at End-of-Job.

Also, vector elements (or entire arrays) may be written during output time by specifying vector names (with or without indices) as field names (VARIABLE NAMES) in the Output-Format Specifications. The same basic rules apply as for specifying vector names in the Input Specifications. The individual elements of the vector may be edited on output, as any other field.

VECTORS IN ALTERNATING FORMAT.
Vectors specified as occurring in alternating format have related elements contained in alternating format on each input record (see figure I-5). The two vectors need not be of the same size (LENGTH OF ENTRY), type (numeric or alphanumeric), or sequence order, but they must have the same number of elements contained on each input record, and each vector must contain the same number of elements (NUMBER OF ENTRIES PER TABLE OR ARRAY). Each pair of elements on the input record is considered one entry. The first element of each pair belongs to the vector described in columns 27-45; the second element belongs to the vector described in columns 46-57.

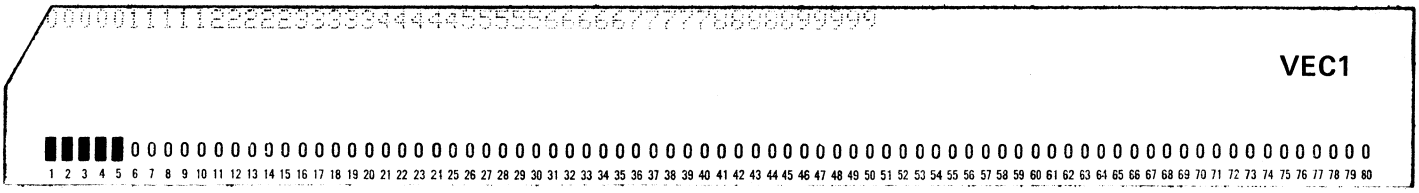
EXTENSION SPECIFICATIONS

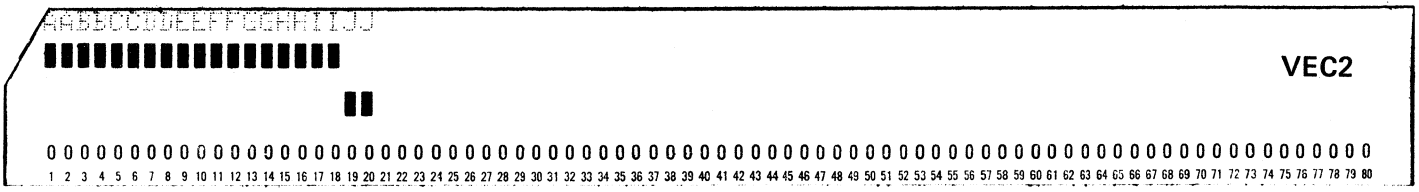
PAGE □□ (1 2)

PROGRAM IDENTIFICATION □□□□□□ (75 80)

LINE (3 5)	FORM TYPE (6)	NOT USED (7 8)	CHAINING FIELD NUMBER (9 10)	FROM FILENAME (11 18)	TO FILENAME (19 26)	TABLE OR ARRAY NAME (VECTOR NAME) (27 32)	ENTRIES PER RECORD (33 35)	ENTRIES PER TABLE OR ARRAY (36 39)	LENGTH OF ENTRY (40 42)	PACKED (43)	DECIMAL POSITIONS (44)	SEQUENCE (45)	TABLE OR ARRAY NAME (VECTOR NAME) (46 51)	LENGTH OF ENTRY (52 54)	PACKED (55)	DECIMAL POSITIONS (56)	SEQUENCE (57)	COMMENTS (58 74)
01	E	▨		VECTOR1		VEC1	10	10	5		2							
02	E	▨		VECTOR2		VEC2	10	10	2									
03	E	▨				VEC3	10	10	3		2		VEC4	2				

INPUT RECORDS





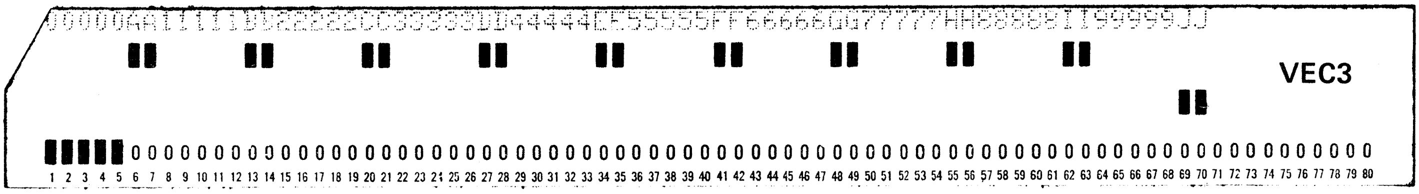
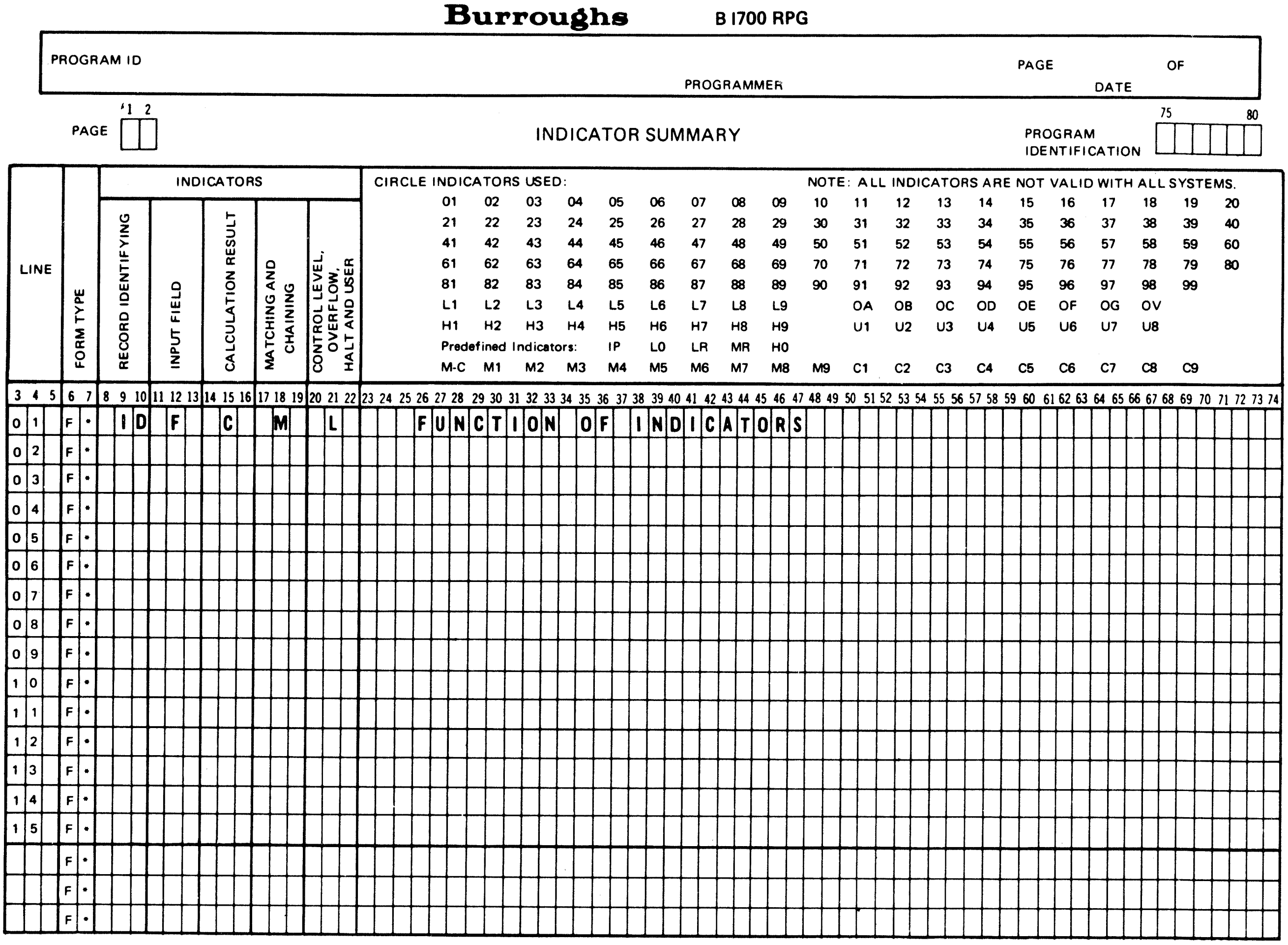


Figure I-5. Vectors in Alternating Format

APPENDIX J

BURROUGHS INDICATOR SUMMARY FORM

The RPG Indicator Summary Form illustrated below (see figure J-1) is used strictly for documentational purposes. Its function is to provide an accurate record of which indicators are used and the function of those indicators in that RPG Program.



Burroughs B 1700 RPG

PROGRAM ID

PROGRAMMER

PAGE OF

DATE

PAGE □□ (1 2)

INDICATOR SUMMARY

PROGRAM IDENTIFICATION □□□□□□ (75 80)

CIRCLE INDICATORS USED:

NOTE: ALL INDICATORS ARE NOT VALID WITH ALL SYSTEMS.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	
L1	L2	L3	L4	L5	L6	L7	L8	L9		OA	OB	OC	OD	OE	OF	OG	OV		
H1	H2	H3	H4	H5	H6	H7	H8	H9		U1	U2	U3	U4	U5	U6	U7	U8		
Predefined Indicators:				IP	L0	LR	MR	H0											
M-C	M1	M2	M3	M4	M5	M6	M7	M8	M9	C1	C2	C3	C4	C5	C6	C7	C8	C9	

LINE	FORM TYPE		INDICATORS					
			RECORD IDENTIFYING	INPUT FIELD	CALCULATION RESULT	MATCHING AND CHAINING	CONTROL LEVEL, OVERFLOW, HALT AND USER	
3 4 5	6	7	8 9 10	11 12 13	14 15 16	17 18 19	20 21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	F	*	ID	F	C	M	L	FUNCTION OF INDICATORS
0 2	F	*						
0 3	F	*						
0 4	F	*						
0 5	F	*						
0 6	F	*						
0 7	F	*						
0 8	F	*						
0 9	F	*						
1 0	F	*						
1 1	F	*						
1 2	F	*						
1 3	F	*						
1 4	F	*						
1 5	F	*						
	F	*						
	F	*						
	F	*						

Figure J-1. Burroughs Indicator Summary Form

The Indicator Summary Form fields and their content are:

Column	Content
6	Form type (optional predefined F).
7	Asterisk required - causes complete card to be commented.
8-10	Record identifying indicator.

Column	Content
11-13	Input field indicators.
14-16	Calculation resulting indicators.
17-19	Matching and chaining indicators.
20-22	Control level, Overflow, Halt, and User indicators.
23-74	The function of the indicators described in columns 8-22.
75-80	Program ID.

INDEX

cut along dotted line

BURROUGHS CORPORATION
DATA PROCESSING PUBLICATIONS
REMARKS FORM

TITLE: B 1700 REPORT PROGRAM GENERATOR
REFERENCE MANUAL

FORM: 1057189
DATE: 1-73

CHECK TYPE OF SUGGESTION:

□ ADDITION □ DELETION □ REVISION □ ERROR

GENERAL COMMENTS AND/OR SUGGESTIONS FOR IMPROVEMENT OF PUBLICATION:

FROM: NAME ______________________ DATE ______________

TITLE ______________________

COMPANY ______________________

ADDRESS ______________________

____

STAPLE

FOLD DOWN SECOND FOLD DOWN

Postage Will Be Paid by Addressee

No Postage Stamp Necessary If Mailed in the United States

BUSINESS REPLY MAIL
First Class Permit No. 817, Detroit, Mich. 48232

Burroughs Corporation
6071 Second Avenue
Detroit, Michigan 48232

attn: Sales Technical Services
Systems Documentation

FOLD UP FIRST FOLD UP

Wherever There's
Business There's Burroughs

1057189 1-73 Printed in U. S. America