HONEYWELL EDP

SOFTWARE MANUAL

SERIES 200

REPORT GENERATOR A AND B

GENERAL SYSTEM: SERIES 200/BASIC PROGRAMMING SYSTEM

SUBJECT: Programming and operating procedures for Report Generator A (4K characters of main memory, two-character address mode) and Report Generator B (8K characters of main memory, three-character address mode).

SPECIAL INSTRUCTIONS: This publication supersedes the Honeywell Series 200 Report Generator Manual, DSI-325A.

DATE: December 30, 1965

FILE NO. 123.6605.000A.1-080*

8622
51265
Printed in U.S.A.

*When ordering this publication please specify Title and Underscored portion of File Number.

SECOND EDITION
First Printing, December, 1965

HONEYWELL EDP

SOFTWARE BULLETIN

SERIES 200

REPORT GENERATOR A AND B
ADDENDUM #1

GENERAL SYSTEM:	SERIES 200/BASIC PROGRAMMING SYSTEM
SUBJECT:	Additional information pertaining to the Report Generator B system. The information contained in this addendum reflects system changes and clarifications of the documentation and the operating procedures. Report Generator A is not affected.
SPECIAL INSTRUCTIONS:	Pages 1-1, 1-2, 1-5, 1-6, 2-13, 2-14, 2-23, 2-24, and all pages in Section VI shall be deleted from the Honeywell Series 200 Software Manual Report Generator A and B, Order Number 080, and replaced with the attached pages. The information contained herein will be incorporated into the manual at the time of the next revision.

DATE: March 25, 1966

FILE NO.: 123.6605.000A.1-080*

8807
5366
Printed in U.S.A.

*When ordering this publication please specify Title and Underscored portion of File Number.

FOREWORD

This publication presents the programming and operating information necessary for use of the Honeywell Series 200 Report Generator (RG), a program which provides users of the Series 200 Electronic Data Processing System with an efficient and easy means of preparing business data processing reports without concern for the detailed coding involved. The user simply describes his report on a set of specifications cards, from which the Series 200 RG generates a program to write the required report. Thus, reports can be prepared with a minimum of user effort and familiarity with the computer system.

Designed for Series 200 systems having 1/2-inch tape drives, the Report Generator program handles, with only minor modifications, input decks prepared for the 1401 RPG program. Consequently, personnel who are experienced in preparing 1401 RPG specifications cards will find that they can write Series 200 RG specifications by employing familiar procedures, with only minor modifications and additions. Moreover, a modified program will run on the Series 200 at faster speeds than were possible with the original RPG program.

The reader is assumed to be familiar with the operating procedures for the control panel and the various peripheral devices as presented in the Honeywell Series 200 Equipment Operators' Manual, Model 200 (DSI-294). In addition, the reader should be familiar with the Honeywell Series 200 Programmers' Reference Manual (Models 200/1200/2200, File No. 113.0005.0000.00.00; or Model 120, File No. 113.0006.0000.00.00) and the manual Easycoder 4K Operating Procedures (DSI-243C).

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

TABLE OF CONTENTS (cont)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I
INTRODUCTION

GENERAL DESCRIPTION

The primary purpose of the Honeywell Series 200 Report Generator is to free the programmer from the tedious detail required to develop and code numerous report programs when a variety of report outputs are required. Each individual report requires a separate program in order to present information in a desired arrangement of printed copy. If many different kinds of reports are required, or if report format must be changed frequently, the programming effort required is considerable.

The Report Generator permits the programmer to decrease substantially the amount of programming effort required to produce variable format reports. By using the Report Generator, the programmer no longer needs to write and encode a separate program for each report type required. Instead, he prepares the specifications, or descriptors, which outline the report format desired and merges these with the Report Generator master deck. The Report Generator then produces a program which writes the report in accordance with the specifications stated by the programmer.

The Honeywell Series 200 Report Generator (RG) is fully compatible with competitive Report Program Generator (RPG) systems. Thus, a programmer who has developed knowledge of 1401 RPG, for example, will find that 1401 RPG programs already written may be processed by the Series 200 RG with only minor modifications or additions.

There are two major steps in the process of operating the Series 200 Report Generator: (1) describing the report and establishing the format which the output will take, and (2) generating the symbolic program which is assembled and executed to produce the requisite report.

The report output can be in virtually any format; output limitations are imposed only by form size and the user's preference for arrangement of information on the printed page. Section II of this publication describes the detailed steps to be followed in order to obtain the desired report arrangement.

In the second step, executing the Series 200 RG, the specifications and the RG master deck are used to generate a symbolic program which is then assembled and executed to produce the desired report.

PROGRAM INPUT AND OUTPUT

The user is permitted wide latitude in his choice of input and output media. The simplest form of the Report Generator system uses punched card input of specifications, RG master deck, and data file (which may be card images on magnetic tape) to produce a printed report as output.

Generally speaking, two versions of the Series 200 Report Generator system exist: (1) Report Generator A, a 4K version in which card input to the generated program is single-buffered; and (2) Report Generator B, an 8K version in which card input to the generated program is double-buffered. Card output and printed output from a program generated by Report Generator A is single-buffered. In a program generated by Report Generator B, card output is double-buffered for all punched cards which do not use the stacker-select feature, and on-line printed output is double-buffered. The object program produced when using Report Generator B overlaps computing with printer and card reader operations. This saves approximately 30% of the number of memory cycles required by the non-overlapped programs produced by Report Generator A. In addition, the punched output of an object program produced with Report Generator B will be double-buffered except in the following cases:

1. The program has <u>both</u> card input and output and will be executed on a Model 120 equipped with an integrated control unit.
2. The time-saving effect of double-buffering is lost when punching cards that use the stacker-select feature.

Figure 1-1 shows all of the input/output possibilities from which the user may choose:

1. The specifications source deck of punched cards is loaded together with the RG master deck which may be either on punched cards or on magnetic tape.
2. The generated symbolic program may be produced directly on punched cards, or optionally on magnetic tape as an intermediate step. If the generated program is produced on magnetic tape, a symbolic card deck in the proper sequence is always produced from the tape; thus, hand sorting of the symbolic program is not necessary. The generated program is assembled by Easycoder Assembler A or B, which may be on either punched cards or magnetic tape, to produce a machine-language object program on punched cards or on magnetic tape, at the user's option. If the generated program is loaded from cards using a Type 214-2 or a Type 224-2 reader/punch, the End card of the condensed deck must be manually run out from the reader before the generated program does any punching. If this is not done, the first data section card punched by the generated program will be punched over the End card of the generated program condensed deck. Honeywell Report Generator B generates a symbolic program in the three-character address mode; Report Generator A uses two-character addressing.
3. The object program produced by Report Generator A or B can accept <u>either</u> card <u>or</u> tape files as input. Output of the object program may be:
   a. a printer report;
   b. a punched-card report;
   c. a print-image report (with or without spacing-control characters or lines) written on tape; or

d. any desired combination of the above three media (unless the tape contains spacing-control characters).

4. Tape input to the object program may consist of bannered or bannerless files, with or without header labels, and may contain blocked, fixed-length records or unblocked, fixed- or variable-length records. Tape output of the object program may be in either of the following two formats:

   a. A print-tape file of 133-character records with a line-spacing control character as the first character in each record. The file is identified by a 1HDRΔ header record having a banner character of 60_8 and is terminated by two 1ERI records.

   b. A tape file of 132-character records which begins with a 1HDRΔ record and ends with two 1ERI records.

Note that both Report Generator A and Report Generator B may accept card or tape input and both may produce printed, punched, or magnetic tape output. In Report Generator B, additional core storage may be utilized effectively since the generated program uses the three-character address mode.

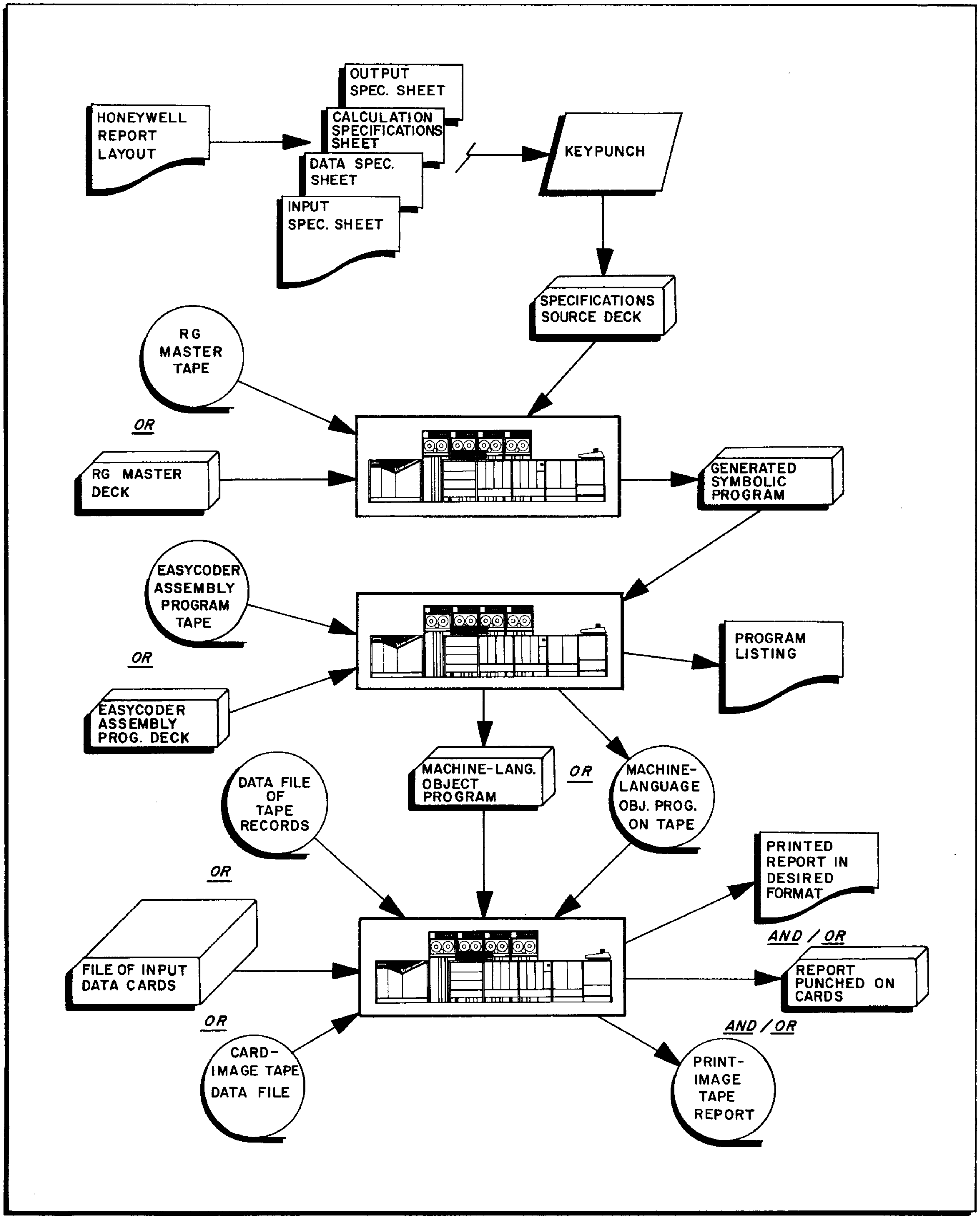

Figure 1-1. Systems Flow Chart of Series 200 Report Generator (Input/Output Possibilities)

movement, arithmetic manipulation of the data fields by logical alternatives, and finally, the movement and editing of these fields into the output area where the lines of the report are formed.

The five logical segments of the Series 200 Report Generator system are:

1. Control. This segment supplies the rest of the system with information about the hardware available for use and processes the generated input/output instructions.

2. Input. Instructions to search for and identify the source information for the report are constructed by this segment.

3. Data. The coding generated during this segment is such that once the source of information (i.e., the data record) has been identified, it is operated upon and placed in a field so that either the instructions produced by the output section may put the information into the report, or the instructions produced by the calculation section may further manipulate the information in the field.

4. Calculation. When such operations as comparisons and arithmetic manipulations are to be performed on the data fields and summation-type information is also desired, instructions to perform these operations are set up during the calculation segment.

5. Output. The instructions necessary to pick up the proper fields for the formation of each report line, when conditions for forming that line are met, are generated by this final segment, and the carriage-control tape-simulator card is also processed.

Creation of the object program is a two-part operation. The first part (generation) consists of feeding the Series 200 RG master deck (on cards or magnetic tape) and the specifications source deck into the machine and producing a "generated symbolic program" on punched cards, as shown in Figure 1-1. The second part of the operation (assembly) consists of feeding the generated symbolic program and the Easycoder Assembly Program A into the machine and producing an object program deck. In addition to the object program deck, the Assembly Program also produces a printed listing containing the symbolic source program and the corresponding object-program entries. Execution of the object program produces the report in the specified format and on the desired medium.

EQUIPMENT CONFIGURATION

The equipment requirements for the Honeywell Series 200 Report Generator are tabulated in Table 1-1. The tabulation shows the equipment configuration which is required both for Report Generator and for object program functions. In the Required Equipment column is shown that equipment which is required for a punched card version of the Series 200 RG. The Alternate Equipment column details the equipment which can be utilized, at the user's option, if magnetic tape input to the Series 200 RG is used.

Table 1-1. Equipment Configuration for Honeywell Series 200 Report Generator

REPORT GENERATOR FUNCTION	REQUIRED EQUIPMENT	ALTERNATE EQUIPMENT
Program Input	Card Reader (Type 214, 223, or 227)	Magnetic Tape Unit (Type 204B)
Control Information (Object Program Specification)	Card Reader (Type 214, 223, or 227)	----------------
Generated Output of RG (Easycoder Program Deck)	Card Punch (Type 214, 224, or 227)	Magnetic Tape Unit (Type 204B)[1]
Central Processor Requirements	4096 core storage locations (Report Generator A); 8192 core storage locations (Report Generator B); Editing instructions	----------------
OBJECT PROGRAM FUNCTION	**REQUIRED EQUIPMENT**	**ALTERNATE EQUIPMENT**
Program Input	Card Reader (Type 214, 223, or 227)	Magnetic Tape Unit (Type 204B)
Printed Output	Printer (Type 222 or 206)	Magnetic Tape Unit (Type 204B)
Data Input, Card Files or Card Images on Tape	Card Reader (Type 214, 223, or 227)	Magnetic Tape Unit (Type 204B)
Data Input, Tape Files	Magnetic Tape Unit (Type 204B)	----------------
Punched Output	Card Punch (Type 214, 224, or 227)	----------------
Magnetic Tape Output	Magnetic Tape Unit (Type 204B)	----------------
Central Processor Requirements	4096 core storage locations (Report Generator A); 8192 core storage locations (Report Generator B); Editing instructions	

[1] If Report Generator B is run on a computer equipped with a combination reader/punch, a work tape on logical drive 1 is required for the punched card images (see "Format of Control Card Specifications" in Section VI).

In addition to the required and alternate equipment specified, certain other equipment can be used by the Report Generator to increase operating efficiency and/or to decrease run time. The additional usable equipment is as follows:

1. Additional core storage locations;
2. Multiply/Divide feature;
3. Stacker-Select feature.

WARNING: The user must be very careful, when using the Series 200 Report Generator on a machine with only 4,096 core storage positions, to ascertain that there is sufficient memory for his program. If the program is a complex one involving many calculations and incorporating an intricate report format, it may not be successfully assembled. Since a complex report generates many tags, there is danger of overflowing the tag table, as indicated by the assembly listing. In this case, although assembly is completed, the object program cannot be executed.

SECTION II

THE SPECIFICATIONS - HONEYWELL SERIES 200 REPORT GENERATOR

DESCRIPTION OF THE REPORT

When the basic details of the type and desired format of the report are presented to the programmer, he then must state these specifications in a manner which can be interpreted by the Report Generator. The specifications describe the information which is to be printed in the report (or punched on cards or written on magnetic tape, at the programmer's option). The programmer must also describe the input data and how it is to be manipulated. He lays out the report exactly as it will appear in printed form, describes the data and the calculations to be performed with the data, and defines the report output by specifications which he writes on coding sheets designed for the purpose.

THE HONEYWELL REPORT LAYOUT CHART

Before he can specify the details of the report's data content, the programmer must specify the arrangement, or format, that the final report will take. The Honeywell Report Layout chart is used for this purpose; on it, the programmer can establish the positions at which various data will be printed, and he can also assign each line of printed output an identification code. This information will be used primarily on the Output Specifications sheet when it is time to prepare that document. A copy of the Honeywell Report Layout chart is shown in Figure 2-1.

The primary object of preparing the Honeywell Report Layout chart is to obtain a visual picture of the Honeywell high-speed printer output of this report prior to writing the report specifications. The printed report representation provided by the layout chart allows the programmer wide latitude in arranging the format of the printed output. On the chart he labels lines, and from the chart representation, he can identify the fields within each line. In addition, he can indicate the spacing between printed lines, the spacing between fields within each line, and the size of each field. Pre-printed headings on the report form can be written on the spacing chart for the user's convenience.

The Honeywell Report Layout chart is numbered across the top to correspond to the print positions of the Honeywell 206 and 222 high-speed printers. The chart provides a print-line layout diagram of up to 132 print positions per line. Forty-eight vertical lines are provided on the chart. Additional lines can be obtained by taping two layout charts together. By noting the position numbers at which he wants printing to occur, the programmer can specify report

REPORT LAYOUT

PROGRAM ________________ PROGRAMMER ________________ DATE __________ PAGE _____ OF _____ REPORT NAME ________________

COMMENTS ____

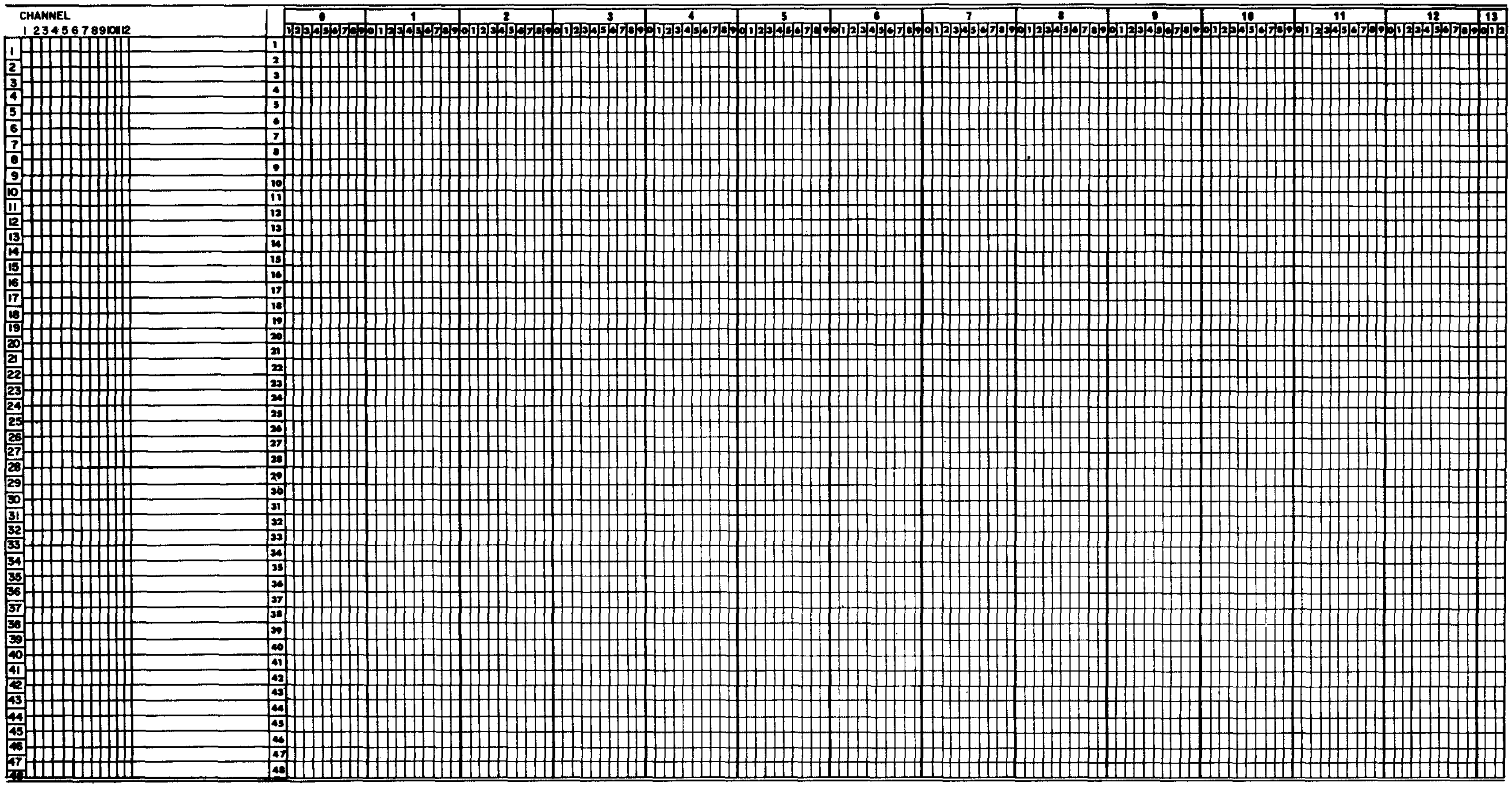

Figure 2-1. Honeywell Report Layout Chart

arrangement on the Output Specifications sheet. Each vertical line is also numbered (in the left-hand margin); vertical spacing is determined by noting these line numbers in the appropriate columns of the Output Specifications sheet.

In laying out the format of the report as he desires to have it printed, the programmer must follow a few basic rules. Headings and constant information must be spelled out completely on the layout chart when the report layout is being designed. Variable information is represented by X's on the layout chart to show print positions, and includes symbols such as dollar signs, etc., where applicable. The last position to be zero-suppressed during the printout is represented by a zero rather than an X. For example, if a dollar amount is being specified which will never exceed $99,999.99, the representation on the layout chart should be $XX,XX0.XX. This representation provides for suppression of all leading zeros to the left of the zero shown in the example. The dollar sign can be floated to its correct position, by means of appropriate coding on the Output Specifications sheet, if the amount obtained during calculation contains fewer significant digits than are provided for by the representation (see page 2-40.)

Line Identification and Classification

Each line of print specified on the Report Layout chart must be assigned a three-character line-identification code. The purpose of this code is to furnish a reference for the print line when the Output Specifications sheet is prepared. The output specifications describe the type and content of each line in the printed report; hence, quick and easy identification of each line is mandatory at the time when the format of the output is specified. The three characters in the line-identification code designate line type, level, and number as follows:

1. Type - The first line-identification code character shows the type of the line that is being described. Every line of print must be identified as belonging to one of three types:

   H identifies a heading line;

   D identifies a detail line;

   T identifies a total line.

2. Level - The second character in the line-identification code is a number from 1 to 8, which identifies lines within a hierarchy and the numerical order of the line in the hierarchy. Independent lines are identified by an alphabetic character.

3. Number - The third character in the line-identification code is a number for lines which appear in a prescribed order in the output. If sequential ordering is not necessary, an alphabetic character is used.

There are several important points to be considered when identifying and classifying these lines which will become printed output. The line-identification codes described above are used to classify the output lines into the various categories which make up the printed output. It is a necessity to categorize the lines which will become output by their type, level, and number, as mentioned above.

Type classification of output lines into heading, detail, and total lines is a straightforward, logical process of categorization which usually does not require detailed explanation. The main point to be emphasized is the significant way in which heading and detail lines vary from total lines. Heading and detail lines can contain information from that input record which is in the input area when the lines are being generated for printing. Total lines cannot contain information from such a record, although the record in the input area can, as a result of control-field change, determine whether the lines will print. Heading lines, on the other hand, are usually composed mainly of constant information and are only partly made up of information from the input record. Detail lines are composed mainly or entirely of information from the input record.

The major concept to be fully considered during output line classification is that of hierarchies and hierarchical relationships within output line levels. A hierarchy of output lines will contain a number of lines, some of which have major importance, some may have intermediate weight, and the balance have minor importance in the relationship. The rank assigned to each level in the hierarchy of output lines determines the order in which any one line will be printed in relation to others in the hierarchy. Those lines which have major importance will force the intermediate and minor lines to print in the particular order desired. Total lines have a different hierarchical relationship from heading and detail lines. Total lines are in ascending order by level, while heading and detail line-levels are in descending order. Major lines in a hierarchy of total lines will therefore force minor lines to print before intermediate lines, and intermediate lines are forced to print before major lines. In a hierarchy of heading or detail lines, major lines print first (since line-levels are in descending order) and force intermediate lines to come after them, followed by minor lines.

Lines which are related in a hierarchy are assigned numerical-level designations on the report layout, which define their order of appearance (in descending order) as output. Heading lines would thus be designated H3x, H2x, and H1x to define major, intermediate, and minor levels of lines in a hierarchy. Total lines, being in ascending order by level, must be designated in the order: T3x for major lines, T2x for intermediate, and T1x for minor lines which exist within a hierarchy. Note that the first (major) heading line which appears on the report layout is designated by the highest as H3x, followed by lower levels. The reverse is true for total lines; the first to appear on the report layout is designated T1x, followed by T2x, etc., up to the major level.

Not all output lines are related in a hierarchy. Those lines which are independent (not related to lines of other levels) are assigned alphabetic-level designations to show their independence from any output-line hierarchy. Examples of lines which have independent status include heading lines printed as a result of page overflow and total lines printed as page totals or final totals at the end of the report.

It should be noted in connection with the above example that heading lines which are printed as a result of page overflow may also be conditioned to print because of control-field changes. A heading line conditioned both by overflow and control-field change must be given a second line-identification code. This second designation indicates that the header line is to be printed when the overflow condition occurs. It will also be necessary to define both of the line identification codes by assigning two names to the line to differentiate between the two states.

For example, a heading line designated H11 would be printed in the normal way (because it has no second designation). If this line were also to be printed as a result of a page overflow condition, it would also be designated as HA1. Thus, the comments column of the Report Layout chart would contain the entry "H11, HA1," to show that the two line identification codes are designated for the report line being considered. Both H11 and HA1 must be defined by line entries on the output specifications sheet. When both line identification designations have been so defined in the output specifications, the heading line will be printed under either of the stated conditions, thus permitting the heading line to be produced not only on the first page, but also on any and all subsequent pages.

The sequence in which a line within a level will be printed is designated by its line number. Any lines within a hierarchy must be designated by numerical sequence, even though only one line may be present in a particular level. For example, the designation H31 denotes a heading line of level 3 that is the first line in level 3 to be printed. Heading line H31 does not have to be followed by other numbered lines (e.g., H32, H33, etc.), but it must be followed by lines of levels 2 and 1 (H2x, H1x). In the case of independent lines, line numbers may be represented by alphabetic letters (e.g., HAA) if only one line is present. Numerical line-number designations for independent lines must be used when a particular sequence of the lines is desired in the output (e.g., HA1, HA2, etc.).

THE SERIES 200 REPORT GENERATOR SPECIFICATIONS

Before the object program can be generated, the programmer must supply certain information to the Report Generator program. The information is supplied by several different kinds of specifications, each of which defines one of the basic portions of the process of report generation. The report specifications are divided into four major types: Input, Data, Calculation, and Output specifications. The format of each type is described onthe following pages.

Format of Input Specifications

The input specifications for the report which is to be generated by the Series 200 Report Generator program define the input file used in preparing the report. The input records within each file must be described, as well as the record codes which identify each record. Report input specifications define the characteristics of the file from which the input data for the report are to be obtained. In order to extract the required information from the data file, the file must be described in detail. It is necessary to state the types of records contained in the file, the record codes identifying each record type, and the control fields associated with each record type.

The descriptive information is entered on the Input Specifications sheet (see Figure 2-2); this form, when completed, contains the requisite information in the correct arrangement for punching into 80-column cards. Table 2-1 presents the format for preparing input specifications and explains the entries, column by column.

Table 2-1. Format of Input Specifications

Parameter	Columns	Contents	Explanation
CARD ID	1	C, or S	C denotes card input for every line entry which specifies a record type. When a group of records must be in a fixed sequence, enter S on the line below the last entry for that program.
ORDER OF SEQUENTIAL RECORDS	2 - 4	See below	These columns indicate the number and sequence of records within control groups.
Sequence	2 - 3	Two-digit decimal number, two alphabetic characters, or CF	If a fixed sequence is required to ensure proper processing, the records are assigned numerical sequence in ascending order. Two alphabetic characters are used to indicate that no fixed sequence is required. If column 1 had an S, enter CF.
Sequential records per control group	4	1, N, Blank, or Control Field number	If only one sequential record per control group, enter 1. If more than one sequential record per control group, enter N. The blank indicates non-sequential records within the control group. If columns 1-3 entries are SCF, enter the number of the control field which is to be used.
OPTIONAL RECORDS	5	X, or blank	This column is used to specify the handling of optional records. Codes are: X - a sequential record-type whose presence is optional. Blank - a sequential record-type whose presence is required; or a non-sequential record.
RECORD CODE IDENTIFICATION OF RECORD TYPE	6 - 11	See below	These columns describe and show the location of the record code character which identifies this record type.
Record code character position	6 - 8	Three-digit decimal number	Position in the record of the record code character which identifies this record type.
Record type	9	N or blank	N, if absence of the record code character identifies this record type. Blank, if presence of the record code character identifies this record type.

Table 2-1 (cont). Format of Input Specifications

Parameter	Columns	Contents	Explanation
Code specified	10	Z, D, or C	Enter Z to indicate that the zone portion of the code character specified in column 11 identifies this record type.[1] Enter D to indicate that the digit portion of the code character specified in column 11 identifies this record type.[1] Enter C to indicate full character of the code specified in column 11 identifies this record type.[1]
Record code character	11	Any valid character	The record code character which is used to identify this record type. Any valid number, letter, or special character can be used.
ADDITIONAL RECORD CODES	12 - 41	(Other codes)	Room is provided for five additional record codes to identify this record type. Columns 12-17, 18-23, 24-29, 30-35, and 36-41 to be entered in the same manner as columns 6-11, above. (NOTE: All codes considered in an AND relation.)
RESULTING CONDITION NUMBER	42 - 43	Two-digit decimal number	A unique number used to identify a record of this record-type.
CONTROL FIELD IDENTIFICATION OF RECORD TYPE	44 - 48	See below	These columns specify the control field for this record type.
Control field location	44 - 46	Three-digit decimal number	Location in the record of the rightmost character position of the most minor control field (control field 1).
Control field length	47 - 48	Two-digit decimal number	Length of control field 1.
ADDITIONAL CONTROL FIELDS	49 - 73	(Other control fields)	Five other control fields may be specified for this record type. Columns 49-53, 54-58, 59-63, 64-68, and 69-73 to be entered in the same manner as columns 44-48, above.
-------------	74 - 75	Blank	Not used.
PAGE NUMBER	76 - 77	Two-digit decimal number	Page number of this specification sheet.
CARD NUMBER	78 - 80	Three-digit decimal number	Card number (preprinted on specification sheet).

[1] For example, the letter A is keypunched R, 1. R is the zone portion; 1 is the digit portion; and the R, 1 is the full character.

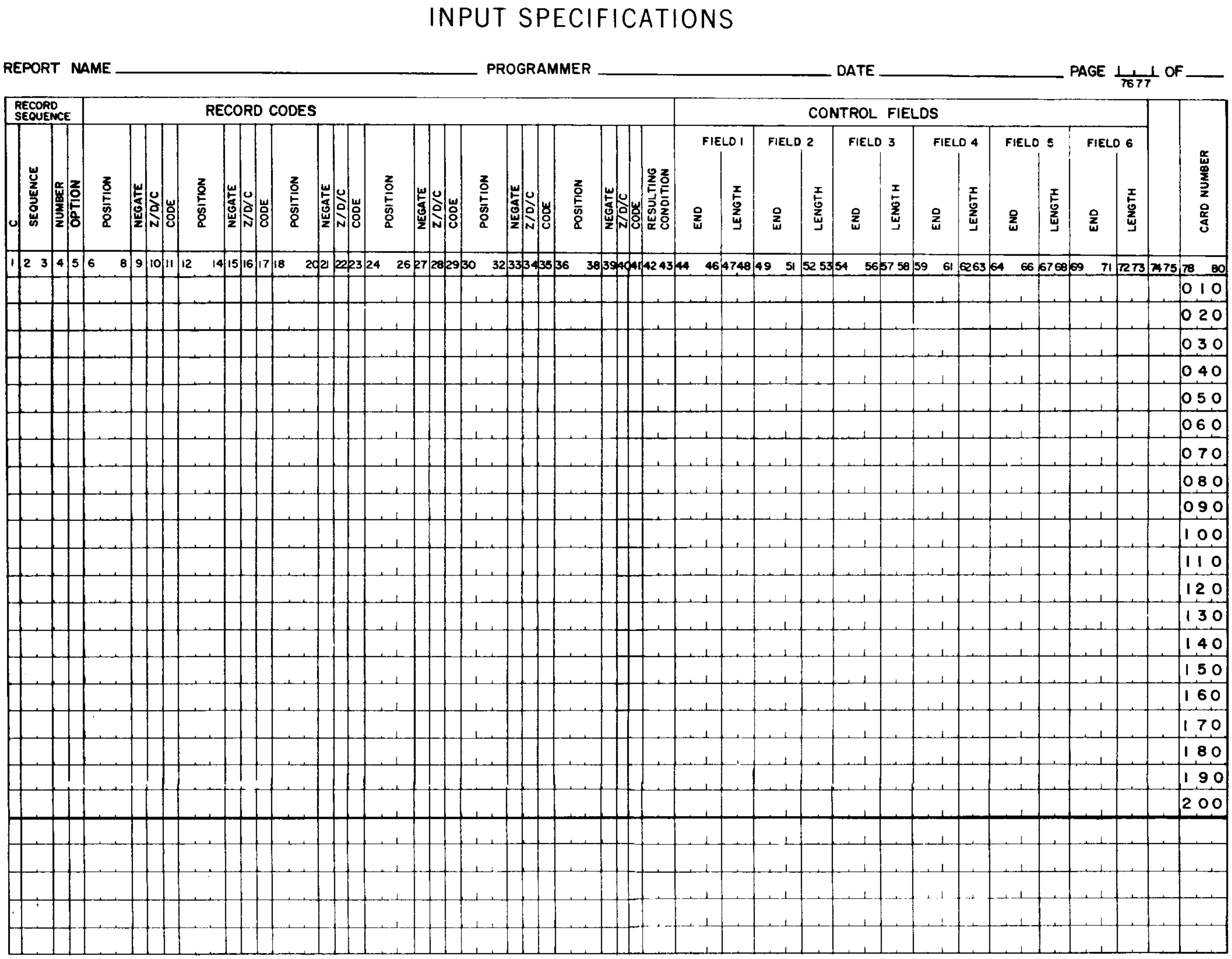

Figure 2-2. Series 200 RG Input Specifications Sheet

The Input Specifications sheet (Figure 2-2) is used to state a complete description of the input file which will be used as a data source in preparing the desired report. The input file is described in terms of each type of record it contains, the codes which identify each record, and the control fields associated with each record type.

In addition, the records which must be processed in an established sequence within the control groups (when control groups are present in the input data file) can be assigned their proper numerical order in the input specifications. The control groups in Series 200 RG are used in obtaining group-printed or group-indicated report formats. Grouping report output in this manner is a tabulating equipment concept (see the Honeywell Software Manual <u>Tabulating Simulators A and B</u>, File No. 123.6705.000A.00.00).

The input specifications have three major divisions, as described below:

1. Record Sequence. When records within a control group in the input file must be processed in a certain order, this status can be designated by numerical sequence entries on the Input Specifications sheet as shown in Figure 2-3, line entries 2-6. In the event that a particular sequence of records has not been established within an existing control group, or if it is desired to prevent a machine halt during processing when records are out of sequence, alphabetical rather than numerical sequence designations may be used. An example of a non-sequential record which has been given an alphabetic sequence designation is shown in line entry 1 of Figure 2-3. This example shows an input data record which has no relationship to the sequence in the control group specified below it. The non-sequential record is therefore given an alphabetic (AA) record sequence designation to distinguish it from the other type, i.e., sequential records.

   A position number entry is used to state the input card-column number in which the record-code character is located. The example in Figure 2-3 shows that all record-code characters are located in column 80 of the data specifications cards which will be punched to make up the Report Generator source deck for this application. The record-code character may be located in any punched card column which suits the user's convenience as long as the usage is consistent and is defined on the Input Specifications sheet.

INPUT SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE _ _ OF ____ (76 77)

RECORD SEQUENCE				RECORD CODES																									CONTROL FIELDS													
																													FIELD 1		FIELD 2		FIELD 3		FIELD 4		FIELD 5		FIELD 6			
C	SEQUENCE	NUMBER	OPTION	POSITION	NEGATE	Z/D/C	CODE	POSITION	NEGATE	Z/D/C	CODE	POSITION	NEGATE	Z/D/C	CODE	POSITION	NEGATE	Z/D/C	CODE	POSITION	NEGATE	Z/D/C	CODE	POSITION	NEGATE	Z/D/C	CODE	RESULTING CONDITION	END	LENGTH	END	LENGTH	END	LENGTH	END	LENGTH	END	LENGTH	END	LENGTH		CARD NUMBER
1	2 3	4	5	6 8	9	10	11	12 14	15	16	17	18 20	21	22	23	24 26	27	28	29	30 32	33	34	35	36 38	39	40	41	42 43	44 46	47 48	49 51	52 53	54 56	57 58	59 61	62 63	64 66	67 68	69 71	72 73	74 75	78 80
C	AA			Ø8Ø		C	%																					Ø1														010
C	Ø1	1		Ø8Ø		C	A																					Ø2			ØØ6	Ø3										020
C	Ø2	1	X	Ø8Ø		C	B																					Ø3			ØØ6	Ø3										030
C	Ø3	1		Ø8Ø		C	C																					Ø4			ØØ6	Ø3										040
C	Ø4	1		Ø8Ø		C	D																					Ø5			ØØ6	Ø3										050
C	Ø5	N		Ø8Ø		C	1																					Ø6	Ø12	Ø6	ØØ6	Ø3										060
C	Ø5	N		Ø8Ø		C	2																					Ø7	Ø12	Ø6	ØØ6	Ø3										070
C	Ø5	N		Ø8Ø		C	3																					Ø8	Ø12	Ø6	ØØ6	Ø3										080
S	CF	2																																								090
																																										100

Figure 2-3. Example of Input Specifications Sheet Entries

When the sequence of records is specified, the number of records of each type per control group must be defined. If only one record of the type being defined in the line entry is present in the control group, a 1 is entered in the Number column of the input specifications. If more than one record is present, N is entered. Both types of entries are shown in Figure 2-3.

When the presence of a particular sequential record in the input data file is optional, column 5 of the Input Specifications sheet must contain an X as shown in line entry 3 of Figure 2-3. An optional record can be defined as one whose presence in the file is not required in order for proper processing to take place, as opposed to a record type whose presence is mandatory for performing correct processing. An optional record, for example, might be encountered in a payroll application where items such as insurance deductions, savings bond deductions, etc., may or may not be present in each employee's pay record. In the above example, presence of the optional record items is not required for proper processing. If the optional records are present, but only if they are, the additional processing is performed. On the other hand, when presence of the particular record is mandatory for proper processing to be performed, column 5 must be left blank. All non-sequential records must be handled in the same manner as mandatory records.

Line entry 9 of Figure 2-3 illustrates sequence control. This is an entry which must be used whenever the input data file contains input data records which are in sequence within a control group. As shown in line entry 9, the entry SCF is made in the first three columns of the Input Specifications sheet and on the line below the last line entry which defines an input record. The entry in the Number column of this line entry is the number of the control field which governs the sequence of the numerically designated types of records preceding it. Thus, whenever a change occurs in control field 2, the sequence of the records specified will begin again with the first record of the sequence in the next control group. That next control group of input data records will be processed until a change in control field 2, and so on, until the end of the input file is reached.

2. Record Codes. A record code is an identifying character (any valid number, letter or special character including the blank) which is used to reference a particular input record. The presence or absence of the code(s) in a record uniquely identifies the record and distinguishes it from all other records which may be present in the input file. When several record codes are specified for an individual record type, they are considered in an AND relation, i.e., all specified codes must be present in the record in order for the required processing to be performed. An input record can be identified by more than one code; the number of codes is not limited to the six record-code-entry spaces provided on each line of the Input Specifications sheet since more than one line may be used if necessary. The Series 200 RG may use more than six codes, but they must be in an AND relation. The first line does not contain the resulting condition entry or the control fields. The succeeding lines have entries in column 1, the additional record-code entries with all control fields, and the resulting condition only on the last line entry which is used to describe the record.

The record code specified in column 11 of the Input Specifications sheet may be identified in three ways. An entry in column 10 of Z, D, or C specifies how the record-code comparison is to be made; whether on the zone portion, digit portion, or on the full character of the record code, respectively. For a D entry in column 10, the exact digit which will be used for comparison is entered in column 11 as the record code. A C entry in column 10 indicates that the full character (any valid alphabetic-numeric character, including a blank, in column 11) identifies this record. A Z entry in column 10 permits any character which contains the desired zone to be entered in column 11. It is worthwhile to note that a C entry uses less instructions than a Z or D entry, since both of these cause the zone or digit portions to be moved before testing. From the standpoint of program efficiency, a better approach when describing a record code specified by a zone which could never have a digit under it or when describing a digit which could never have a zone over it would be to describe the zone or digit as a character and thereby use less instructions in the record-code comparison.

The resulting condition entries on the Input Specifications sheet are two-digit numbers, each unique to one particular type of input data record. Resulting condition numbers are arbitrarily chosen by the programmer but must be consecutive starting with 01. A unique resulting condition number is used to represent each of the input record types which have been coded according to the programmer's specifications. When more than one kind of record of a particular type is present in the input data file, provisions must be made for distinguishing these records. If the several kinds of records present are to be treated as one type and processed in the same manner, they will all be assigned the same sequence numbers. (This rule applies to both numerical and alphabetic sequence designations.) Record codes which apply to these records are entered in the Code column, and each record defined by a line entry is assigned a unique resulting condition number to represent the record codes. The resulting condition numbers are different, so that the records can be distinguished for later usage. The resulting condition number must be present unless this is a continuation from the previous input specification card. There must be a resulting condition in the last input specification in the deck.

An illustration of three kinds of input records which are defined as one record-sequence type is shown in Figure 2-3, line entries 6, 7, and 8. These records are considered to have an OR relation, i.e., they are records which are coded differently but are processed alike in most respects. The same sequence designation is used for records which have an OR relation, but each is assigned a different resulting condition number.

3. Control Fields. A control field is a constant location where information for control purposes is placed. In the Series 200 RG input specifications, the control fields are record positions which contain information used to identify the record's classification. Each control field specified (up to a maximum of six) is described by its low-order (rightmost) position in the record and, additionally, by field length. Control fields are specified in an ascending order of control level importance. The most subordinate control level is specified in Field 1, followed by the increasingly important fields. Entries continue to the right as far as necessary or to the limits of Field 6, the most major level of control. Control field information need not be in the same position in different record types, but entries on the Input Specifications sheet which specify the same level of control must be consistent in their order. Control field entries are made only on the last line entry for a particular type of record when more than six record codes are present within the record type.

The example illustrated in Figure 2-3 shows control field entries which are explained as follows:

Line entry 1 defines a non-sequential record which is not governed by a control field. This input data record could be a date, an identifying code number which represents the file, or any other designation which is pertinent to the particular application. The source of this record is a header card which precedes the file and on which is punched the desired information.

Line entries 2, 3, 4, and 5 represent records which are governed by a second-order level of control; hence, entries are present only for control field 2. This type of entry would be used if the lowest level of control were not present in the input record being specified.

For example, assume that the levels of control in this report, from lowest to highest, are: (1) salesman's code number, (2) branch office code number, (3) department number, (4) division number, and (5) company or organization number. Provisions have been made in all the input records to specify the next-to-lowest level of control, i.e., branch office number. This is control field 2, and input-specifications line entries are made accordingly. Upon a change in branch office number, the sequence will start again with the highest level (i.e., company number) defined by line entry 2. Line entries 6, 7 and 8 contain the lowest level of control, as well as the second lowest, which is specified in all the input records. Therefore, both control fields are inserted in these line entries.

The entries for field end and field length designate the column positions on the punched cards occupied by the control fields. Control field 2, branch office number, is defined as a three-digit number (03 in columns 52-53 of the input specifications sheet) which ends in column 6 (006 entry) of the data card. By the same definition, control field 1, salesman's code number, is a six-digit number which occupies columns 7-12 of the input data card.

Format of Data Specifications

Report data specifications describe the data fields which are used in processing, as well as those which will appear in the output of the Series 200 Report Generator. The data specifications define the data fields by name, length, status, and source.

This descriptive information, when entered on the Data Specifications sheet (see Figure 2-4), is in the correct form and arrangement for punching into the 80-column cards which make up the Report Generator symbolic deck. Table 2-2 presents the format for preparing the data specifications, together with an explanation of the entries, column by column.

Table 2-2. Format of Data Specifications

Parameter	Columns	Contents	Explanation
CARD ID	1	D	Identifies a data specification. D must be inserted in column 1 for every line entry on the data specifications sheet.
DATA FIELD DESCRIPTION	2 - 19	See below	These columns are used to describe the length and status of the data field.
Field name	2 - 7	Alphabetic name	The name of this data field. Characters must be all alphabetic; no numerical designations are permitted.
Field length	8 - 10	Three-digit decimal number	Length of the data field without punctuation (unedited).

Table 2-2 (cont). Format of Data Specifications

Parameter	Columns	Contents	Explanation
Field status	11	B, Z, N, or P	Blank, zero, negative, or positive, respectively. Used to govern subsequent processing according to the status of this field. Negative and positive status include minus zero and plus zero, respectively.
Resulting condition	12 - 13	Two-digit decimal number	Used to identify the field status in column 11, so it may be referred to subsequently.
Second status condition	14	Same as column 11	Used to specify an additional, or second, status condition of this data field.
Second status identification	15 - 16	Same as columns 12 - 13	Used to identify the field status in column 14.
Third status condition	17	Same as column 11	Used to specify an additional, or third, status condition of this data field.
Third status identification	18 - 19	Same as columns 12 - 13	Used to identify the field status in column 17.
DEFINITION OF DATA FIELD SOURCE	20 - 36	See below	These columns are used to define the source of the data field, field length, the operation to be performed on the source field, and any conditions which govern operation on the source field.
Field source	20 - 22	Cxx, PAG, SER, or RCT	Used to define the source of the data field as follows: (Cxx) - card input records; (PAG) - page number; (SER) - serial numbers; (RCT) - record count numbers.
Field conversion	23	N, M, or blank	This column is used to specify numeric - or month-field conversion; otherwise it is left blank. When month-field conversion is specified, entries must be made in columns 27-29 to designate the specific single characters used in the source field to represent the desired months. See "Field length," columns 27-29, below.
Units position location	24 - 26	Three-digit decimal number or blank	Used to locate the units position of the field in the input record. If the source is PAG, RCT, or SER, these columns are blank.
Field length	27-29	Three-digit decimal number	Used to specify the field length on the input record only if different from that specified in columns 8-10. When month-field conversion is specified in column 23, above, the field length columns must contain the specific single characters used in the source field to represent the month(s) 10, 11, and 12 as follows:

Table 2-2 (cont). Format of Data Specifications

Parameter	Columns	Contents	Explanation
Field length (cont)			- (octal 40), October; J (octal 41), November K (octal 42), December. These characters are most generally used; however, any valid character may be used for designating months 10, 11, and 12.
Operation to be performed	30	Blank, A, S, $\overset{+}{0}$, $\bar{0}$, D, or Y	Used to describe the operation to be performed by the source field on the data field. The result will be in the data field. The codes are as follows: Blank - Move source field into data field; A - add; S - subtract; $\overset{+}{0}$ } - reset add; $\bar{0}$ } - reset subtract; D - move numeric portion; Y - move zone portion.
Negate condition	31	Blank or N	N is used when it is desired to negate the condition specified in columns 32-33. Blank - condition shall <u>not</u> be negated.
Condition to govern operation	32 - 33	Blank, S1*, S2, S3, S4, F1-F6, or two digit decimal number	Used to specify a condition which will govern when the operation on the source field is to be performed: Blank - no condition to govern; S1*, S2, S3, S4 - SENSE switch settings; F1 - F6 - control field changes; Two-digit numbers may be resulting conditions from the input specifications or from other entries of the data specifications.
Second negate condition	34	Same as column 31	Used to negate the condition specified in columns 35-36.
Second condition to govern operation on field	35 - 36	Same as columns 32-33	Used to specify a second condition governing the operation on the source field. The conditions are treated in an AND relationship.
DEFINITION OF SECOND SOURCE OF THE DATA FIELD	37 - 53	Same as columns 20-36	Used when the data field being specified has an additional, or second, source.

*NOTE: Use S1 with care; it is the load and halt switch. S1 and S2 are used by generated tape input routines.

Table 2-2 (cont). Format of Data Specifications

Parameter	Columns	Contents	Explanation
DEFINITION OF THIRD SOURCE OF THE DATA FIELD	54 - 70	Same as columns 20-36	Used when the data field being specified has an additional, or third, source.
----------	71 - 75	Blank	Not used.
PAGE NUMBER	76 - 77	Two-digit decimal number	Page number of this specification sheet.
CARD NUMBER	78 - 80	Three-digit decimal number	Card number (preprinted on specification sheet).

DATA SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE [76 77] OF ______

			FIELD STATUS						DATA FIELD SOURCES																						
ID	FIELD NAME	FIELD LENGTH UNEDITED	STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION		CARD NUMBER
1	2 7	8 10	11	12 13	14	15 16	17	18 19	20 22	23	24 26	27 29	30	31 33	34 36	37 39	40	41 43	44 46	47	48 50	51 53	54 56	57	58 60	61 63	64	65 67	68 70	71 75	78 80
																															010
																															020
																															030
																															040
																															050
																															060
																															070
																															080
																															090
																															100
																															110
																															120
																															130
																															140
																															150
																															160
																															170
																															180
																															190
																															200

Figure 2-4. Series 200 RG Data Specifications Sheet

A copy of the Series 200 RG Data Specifications sheet is shown in Figure 2-4. Data specifications are divided into four major categories which are used for the following purposes:

1. Field Name. Each data field necessary for report processing is identified by a distinctive name. The name assigned to each field is usually descriptive (e.g., SHIPTO, NETPAY, and the like) and must consist of all non-numeric characters. Six columnar spaces are allotted in each line entry for field name on the Data Specifications sheet; field names containing less than six characters must be left-justified. If this field is left blank, it is assumed that the card contains additional source fields for the previous card.

2. Field Length. The length of the data field, unedited, which will appear on the final report, must be specified. The unedited field length does not include the periods, commas, and other punctuation symbols which will be automatically generated by a Series 200 edit instruction when an edit control word is used, and which will appear in the printed report output.

   The concept of unedited field length can be illustrated with a date example. If the report is to be dated, a date header card can be punched and placed in position before the input data file. If the report date were August 27, 1964, the date header card might variously be punched as:

   (1) August 27, 1964;
   (2) Aug. 27, 64; or
   (3) Aug2764

   Example (1) contains 15 spaces; this form would be used only rarely due to its excessive length. However, if the date is punched in the card this way and an exact reproduction of this date form is desired on the printed report, the unedited field-length entry would be 015. It is more likely that the date coded in the card would be in the form shown by either example (2) or example (3). Example (2) contains 11 spaces. If the date is punched in the card this way and exactly this form is desired in the printed output, the unedited field-length entry would be 011. Contrast this with example (3), which contains but seven spaces. By specifying an unedited field-length entry of 007 and using the Series 200 edit instruction and a control word, the output format can be as shown in example (2) when input has the form of example (3).

3. Field Status. Status of a data field is used to control subsequent processing operations in which this field is used. In operations where field status is significant to processing, up to three resulting condition numbers may be assigned to represent field status possibilities. The first field status character, as shown in Table 2-2, has four possible meanings (blank, zero, negative, and positive). The field status character is identified by a two-digit decimal number which assigns a resulting condition. Any number may be used to specify this condition, provided that it is not also used to define a resulting condition on the Input Specifications sheet or Calculation Specifications sheet. The field status, when defined as above, is used for reference to the condition of the data field after the specified operation(s) has been performed by the source field by the object program. The resulting condition can then be used to govern other program operations, such as performing calculations or suppressing total lines.

   The use of field status entries is illustrated in Figure 2-5. The line entries are explained as follows:

DATA SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE ___ OF ___ (76 77)

			FIELD STATUS						DATA FIELD SOURCES																						
D	FIELD NAME	FIELD LENGTH UNEDITED	STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION		CARD NUMBER
1	2 7	8 10	11	12 13	14	15 16	17	18 19	20 22	23	24 26	27 29	30	31 33	34 36	37 39	40	41 43	44 46	47	48 50	51 53	54 56	57	58 60	61 63	64	65 67	68 70	71 75	78 80
D	FIELDA	ØØ7	B	05					CØ7		Ø21																				010
D	FIELDB	ØØ6	Z	03					CØ5		Ø67		A																		020
D	FIELDC	ØØ9	N	14					CØ6		Ø48	ØØ8	$\overset{+}{0}$			CØ7		Ø53	ØØ8	S											030

Figure 2-5. Example of Field Status Entries

Line entry 1 specifies a source field, C07, which is to be moved into the data field, FIELDA. This data field has been given a blank (B) status. If the data field is blank after the source field has been moved, the resulting condition 05 is fulfilled by the program.

Line entry 2 specifies addition (A) of the source field, C05, to the data field, FIELDB, which has a zero (Z) status. The resulting sum will be located in FIELDB. If the data field is zero after the addition is performed, condition 03 will be fulfilled by the program.

Line entry 3 specifies a reset-addition ($\overset{+}{0}$) operation in which the data field, FIELDC, that has a negative (N) status, is reset to zero before the source field, C06, is added to it. The source field, C07, is then subtracted from the sum of this addition, and the resulting remainder is placed in the data field, FIELDC. If the content of FIELDC after completion of these operations is negative, condition 14 is then fulfilled by the program. A data field with a positive status can be manipulated in a similar manner.

The resulting conditions described by each of these line entries can be used to govern other operations in the program. For example, the resulting condition, when fulfilled, can be used to govern whether a calculation specified on the Calculation Specifications sheet is to be performed or not, or whether output lines specified on the Output Specifications sheet are to be printed or suppressed. The designation and function of source fields are described fully in paragraph 4, below.

4. Field Sources. A report field source is defined by the name of the record it is found in (columns 1-3 of the Input Specification), the location of its rightmost position in the record, and its length. Various operations may be performed on the source fields, from simple field movements to the complex operations illustrated in Figure 2-6 and described below. Also, conditions which govern when an operation may be performed can be uniquely identified by the entries made in column 31 through 36 of the Data Specifications sheet.

The field-length columns may be left blank in the data-specifications line entry describing an input record if that length is not different from the unedited field length. An example is shown in line entry 1 of Figure 2-6. If the date is punched in the date header card as "Aug2764", the unedited field length is seven positions (007 entry). The desired output is "Aug. Δ 27, Δ 64", or eleven positions when edited by an editing instruction during program execution. (See paragraph 10 in the Output Specifications sheet discussion, page 2-54.)

Field sources may include such items as input records, page numbers, and record count numbers. Space is provided on each line entry for specifying up to three sources of the report field. Additional line entries may be used if more than three sources are specified (see Figure 2-6, line entries 10 and 11). Identifying names used to indicate the various sources are: PAG for page numbers; SER for serial numbers; RCT for record count numbers; and Cxx for input records. Input records, Cxx, are defined on the Input Specifications sheet; the field source entry on the Data Specifications sheet comes from columns 1-3 of that definition, when an input record is specified as a field source.

DATA SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE ___ (76 77) OF ______

D	FIELD NAME	FIELD LENGTH UNEDITED	FIELD STATUS: STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	DATA FIELD SOURCES: FIELD SOURSE	NUMERIC	FIELD END	FIELD LENGTH	OPERATON	CONDITION	AND CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION		CARD NUMBER
1	2 7	8 10	11	12 13	14	15 16	17	18 19	20 22	23	24 26	27 29	30	31 33	34 36	37 39	40	41 43	44 46	47	48 50	51 53	54 56	57	58 60	61 63	64	65 67	68 70	71 75	78 80
D	DATE	ØØ7							CAA		ØØ7																				010
D	DIVNAM	Ø22							CØ1		Ø26			F1																	020
D	PAGENO	ØØ2							PAG					F2																	030
D	AMOUNT	ØØ6							CØ5		Ø72	ØØ5	A																		040
D	DIVTOT	ØØ6							RCT																						050
D	DIVCNT	ØØ2							RCT					Ø4																	060
D	FIELDA	ØØ5							CØ5		Ø67					CØ5		Ø63		Y											070
D	FIELDB	ØØ5							CBB		Ø48					CBB		Ø65		Y											080
D	FIELDC	ØØ1							CØ3		Ø28		D																		090
D	FIELDN	ØØ8							CØ3		Ø13	ØØ6	÷			CØ4		Ø21	ØØ5	S			CØ5		Ø47	ØØ7	A				100
D									CØ6		Ø73	ØØ7	A																		110

Figure 2-6. Example of Data Specifications Sheet Entries

Field end and field length parameters are used only to define input records which are field sources. These columns are left blank when the field sources are page, serial and record count numbers (PAG, SER, or RCT entries). Field end specifies the units position of the field, and field length specifies the number of positions in the field. These parameters are derived from the input data cards, as described in paragraph 3, Control Fields, of the input specifications discussion.

Each report field source may be designated as a field conversion specification by an entry in the Numeric column of the Data Specifications sheet. An N entered in the Numeric column specifies conversion of a numerical field which has zoning in positions other than the units position. The N in the specification conditions the object program to strip the zone bits from all positions in the field except the units position. In the event that the character in the units position of the field contains no zone bits, the numeric conversion specification will generate an A bit (01xxxx) in the units position.

The Numeric column may also be used to specify a month-field conversion operation. An M in the Numeric column of the Data Specifications sheet identifies the entry as a month-field conversion specification. It is usually of value in any computer application to encode as much information as possible, thereby reducing the size of data files. The month conversion specification permits single-character representation of the months of the year in data files to be automatically converted to corresponding two-character representation when the object program is executed.

For example, January through September can be represented in data files by the single characters 1 through 9 which become 01 through 09, respectively, upon conversion. The months October, November, and December, represented by the single characters -, J, and K, respectively, become 10 through 12 upon conversion.

There are seven permissible operation characters which may be specified to designate the operation which shall be performed on a source field. These operation characters are stated in column 30 of the descriptors tabulated in Table 2-2. Use of the operation characters illustrated in Figure 2-6 is explained below:

> Line entry 4 describes a data field, AMOUNT, which has an unedited field-length of six positions (006 entry). To this field will be added (A) the contents of the source field, C05. The data field, AMOUNT, because it has a greater length than the source field, will be able to accept the largest total expected, since the source field has one less position.
>
> Line entries 7 and 8 specify move-zone operations (Y). In this kind of operation, the zone portion of the single-character source field is moved to the units (rightmost) position of the data field. The numerical portions of both fields remain unaffected by the move operation. Line entry 7 specifies that the five-character source field ending at position 067 in the input record named C05 is to be moved to the data field, FIELDA. In addition, the zone portion of the single-character source field in position 063 of the same input record shall be moved to the units position of FIELDA. The purpose of such an operation is to place the sign located in the high-order position of the source field in the units position of the data field.
>
> Line entry 8 specifies that the five-character source field ending at position 048 of the input record CBB will be moved into the data field, FIELDB, a five-position field. Also, the zone portion of the single-character source field contained in position 065 of the same input record shall be moved into the units position of FIELDB, the data field named in this line entry. This kind of entry permits any position in a record to contribute the sign which is placed in the data field units position. The programmer can remove an unwanted sign present in the units position of the data field simply by choosing a single-position field source which contains no zones to use in the move-zone operation.
>
> Line entry 9 illustrated a move-numerical (D) operation which permits movement of a one-character source field to the units position of the data field, without affecting the zone portion of either field.

> The example in line entry 9 specifies that the single-character source field located in position 028 of the input record named C03 will be moved into the data field, FIELDC, which is a one-position field. This kind of entry is used to move a one-position source field which has an unwanted sign attached to it into the data field, without moving the sign.

Two rules must be followed in specifying these operations to ensure that the object program will perform the operations correctly and in the proper sequence. Since move-numerical (D) and move-zone (Y) operations involve manipulation of single characters only, no source field length will be specified for either of them. Also, when these single-character move operations are specified in the same line-entry as another operation, the D and Y move operations are specified after the other operation. Note that these rules are followed in line entries 7 and 8 of Figure 2-6. The ordering of source entries on a line must also be observed in specifying reset-addition and reset-subtraction operations. In the operation of resetting a field before adding or subtracting more than one field from the same source record, the field which is to be reset should be specified first in the line entry. This principle can be illustrated by a modification of line entry 10 of Figure 2-6 (disregarding the fourth field source specified in line entry 11). If the three sources described in line entry 10 were all from the input record C03, the first field source would then be reset-added into FIELDN, the second source subtracted from it, the third source added to the remainder, and the resulting sum contained in the data field, FIELDN.

Line entries 10 and 11 of Figure 2-6 are used collectively to specify a field which has four sources. Since only three field sources may be specified in one line entry, it is necessary to use a second line entry to specify the remaining source. The rules which govern this kind of entry are that a D must be entered in column 1, columns 2-19 must be left blank, and the remaining field sources are then specified in their proper columns.

Two conditions may be specified which will govern the operation to be performed on the source field. If two conditions are specified, they are considered in an AND relation, and both conditions must be fulfilled in order for the specified operation to be performed. Conditions which may be specified are sense switch settings, control field changes, resulting conditions specified on the Input Specifications sheet or elsewhere on the Data Specifications sheet, or negation of any of these by entering an N before the resulting condition entry. The use of resulting conditions illustrated in Figure 2-6 is explained below:

> Line entry 2 may be used when a group-indicated report format is desired. DIVNAM is the data field which represents the most minor indicator, or label, to be printed in the report. It is desired to print the label only once, at the beginning of the lines which constitute the minor control group, although all input data cards in the group contain this indicator. Therefore, line entry 2 specifies that the label will be taken from the first card in the group only, following a change in control field 1 (F1), and moved into the print area. The resulting condition blocks the appearance of the label from any of the other cards in the group, until the minor control field changes again.
>
> Line entry 3 provides the condition for resetting the page-number count. This entry specifies that page numbers will begin again at 1, upon a change in control field 2. This application is useful in an invoicing or billing operation when the itemizing of

each account may require several pages, and it is desired to have the pages of each account numbered consecutively and to begin again at page 1 for the next account. Only one data field name can be used for page numbering in any individual report generated by the Series 200 Report Generator. The data field name must always be PAGENO; its field source is PAG. When a single PAG field source is specified, pages will automatically be numbered consecutively by the object program from page 1 to the end of the input record file, or to the end of the report (if more than one file is used for input). It is possible, however, to cause resetting of the page number under various conditions by specifying more than one PAG source to the data field named PAGENO. (Refer to "Initializing Numerical Sequences" on page 2-47.)

Line entries 5 and 6 of Figure 2-6 specify record-count operations. Serial count can be performed in the same manner as illustrated by these entries. Line entry 5 specifies counting of all input records. Line entry 6 specifies selective counting of records, governed by the resulting condition 04. This condition must be the resulting condition number of a record type previously defined on the Input Specifications sheet, in order to be used as a resulting condition for this entry. This entry makes it possible to count only the first card in a designated control group from beginning to end of the input file. The resulting record-count total is useful for maintaining a check on accuracy of the input file.

Format of Calculation Specifications

The calculation specifications for the Series 200 Report Generator are used to detail the calculations to be performed which are more extensive or more complex than those which can be described on the Data Specifications sheet. These are the calculations which encompass such operations as comparison, multiplication, division, and arithmetic manipulation of fields resulting from previous calculations. The fields used to perform the calculations must be those formed either by data specifications, a W-entry, or prior calculation-specifications entries. Figure 2-7 illustrates the Calculation Specifications sheet and Table 2-3 presents the format for preparing calculation specifications, with explanation of the entries, column by column.

Table 2-3. Format of Calculation Specifications

Parameter	Columns	Contents	Explanation
CARD ID	1	A	An A must be entered in column 1 of each line entry; identifies the entry as a calculation specification.
DESCRIPTION OF CALCULATION FIELD	2 - 10	See below	These columns are used to specify the field in which the calculation result will be placed, field length, status, and resulting condition(s).

Table 2-3 (cont). Format of Calculation Specifications

Parameter	Columns	Contents	Explanation
Field name	2 - 7	Field name or blank	Used to specify the name of the field which will contain the result of the calculation. Left blank if the operation is COMPARE (C in column 29), or if the result of the operation is contained in the same field as the preceding line entry.
Field length	8 - 10	Three-digit decimal number	Length of the field without punctuation (unedited).
FIELD STATUS	11 - 19	See below	Used to govern subsequent processing according to the status of this field.
First status condition	11	B, Z, N, P, E, U, H, or L	The use of the status codes B, Z, N, and P is the same as on the data specifications: B - blank; Z - ± zero; N - negative (includes minus zero); P - positive (includes plus zero). For comparison operations, the additional status codes are used as follows: E - Factor 2 is equal to factor 1; U - Factor 2 is unequal to factor 1; H - Factor 2 is higher in the collating sequence than factor 1; L - Factor 2 is lower in the collating sequence than factor 1.
Resulting condition	12 - 13	Two-digit decimal number	Used to identify the field status in column 11 so that it may be referred to subsequently.
Second status condition	14	Same as column 11	Used to specify an additional, or second, status condition of this field.
Second status identification	15 - 16	Same as columns 12-13	Used to identify the field status condition specified in column 14.
Third status condition	17	Same as column 11	Used to specify an additional, or third, status condition of this field.
Third status identification	18 - 19	Same as columns 12-13	Used to identify the field status condition specified in column 17.
FACTOR 1	20 - 28	See below	Used to specify the name and length of the field which is Factor 1 for the operation to be performed or the literal constant which is Factor 1.

Table 2-3 (cont). Format of Calculation Specifications

Parameter	Columns	Contents	Explanation
Name	20 - 25	Alphabetic name or literal constant	This name must have been specified on the Data Specifications sheet, previously on the Calculation Specifications sheet, or in a W entry on the Output Specifications sheet. Names of less than six characters are left-justified. A literal in the range from $00000\overset{\pm}{1}$ through $99999\overset{\pm}{9}$ may be specified, with the sign indicated over the units position in column 25. If the literal has no over-punched sign in the units position, it will be generated as unsigned. When a literal is used as a multiplier or a divisor, it must be signed. Literals of less than six digits are right-justified; no leading zeros need be entered.
Field length	26 - 28	Three-digit decimal number	Unedited length of the field which contains Factor 1.
OPERATION	29	+, -, X, /, or C	Used to identify the operation which is to be performed. Codes are as follows: + - plus, for addition operation; - - minus, for subtraction operation; X - multiplied by; / - divided by; C - compare Factor 2 to Factor 1.
FACTOR 2	30 - 38	See below	Used to specify the name and length of the field or the literal constant which is Factor 2 for the operation to be performed.
Name	30 - 35	Alphabetic name or literal constant	Same as columns 20-25 but describing the field which is Factor 2 for the operation or specifying the constant which is Factor 2.
Field length	36 - 38	Three-digit decimal number	Unedited length of the field which contains Factor 2. Note: A Factor 2 must be present on every Calculation Specification card. If Factor 2 is missing, RG uses the tag "ERR30" for the tag of the missing Factor 2. Assembly flags this tag as unassigned.
OPERATION RESULT	39	A, S, $\overset{+}{0}$, $\bar{0}$, or M	Used to indicate whether the result of the operation should be added to (A); subtracted from (S); reset-added to ($\overset{+}{0}$); reset-subtracted from ($\bar{0}$); or moved to (M) data field specified in columns 2-7.

Table 2-3 (cont). Format of Calculation Specifications

Parameter	Columns	Contents	Explanation
OPERATION RESULT (cont)			Notes: $\overset{+}{0}$ is represented by an R-, 0-punch; $\bar{0}$ is represented by an X-, 0-punch when the Calculation Specifications sheet entries are punched into cards. If the character in column 39 is not one of those listed above, RG generates the correct operands but uses as the op code "ERR". Assembly flags the latter as an illegal op code.
CONDITION GOVERNING CALCULATION	40-42	See below	These columns are used to define any condition(s) which govern performance of the calculation.
Negate condition	40	N or blank	N, if condition in columns 41-42 is to be negated; Blank, if condition in columns 41-42 is not to be negated.
Condition for performing the calculation	41 - 42	LC, F1-F6, S1*, S2, S3, S4, two-digit decimal number, or blank	Used to specify a condition which will govern whether or not the calculation will be performed. Codes are as follows: LC - last card; F1-F6 - control field changes; S1*, S2, S3, S4 - SENSE switch settings; Blank - performance of the calculation is unconditional; Two-digit number which was a previously defined resulting condition on the Input, Data, or Calculation Specification sheets.
Second negate condition	43	Same as column 40	Relates to condition in columns 44-45
Second condition for performing the calculation	44 - 45	Same as columns 41-42	Used to specify an additional, or second, condition which will govern whether or not the calculation is performed.
Third negate condition	46	Same as column 40	Relates to condition in columns 47-48.
Third condition for performing the calculation	47 - 48	Same as columns 41-42	Used to specify an additional, or third, condition which will govern whether or not the calculation will be performed. If more than one condition is specified, they are treated in an "and" relationship.

*NOTE: Use S1 with care; it is the load and halt switch
S1 and S2 are used by generated tape input routines.

Table 2-3 (cont). Format of Calculation Specifications

Parameter	Columns	Contents	Explanation
TIME WHEN CALCULATION IS TO BE PERFORMED	49	T or D	T, if this calculation is to be performed at total time; D, if this calculation is to be performed at detail time.
HALF-ADJUST	50 - 51	Two-digit decimal number	Indicates the position in the field of the digit which is to be half-adjusted (i.e., have five added to it).
POSITION-ADJUST	52 - 53	Two-digit decimal number	Indicates the position in the field of the leftmost truncated digit. This digit and all others to the right of it are dropped from the calculation result.
-------------	54 - 75	Blank	Not used.
PAGE NUMBER	76 - 77	Two-digit decimal number	Page number of this specification sheet.
CARD NUMBER	78-80	Three-digit decimal number	Card number (preprinted on the specification sheet).

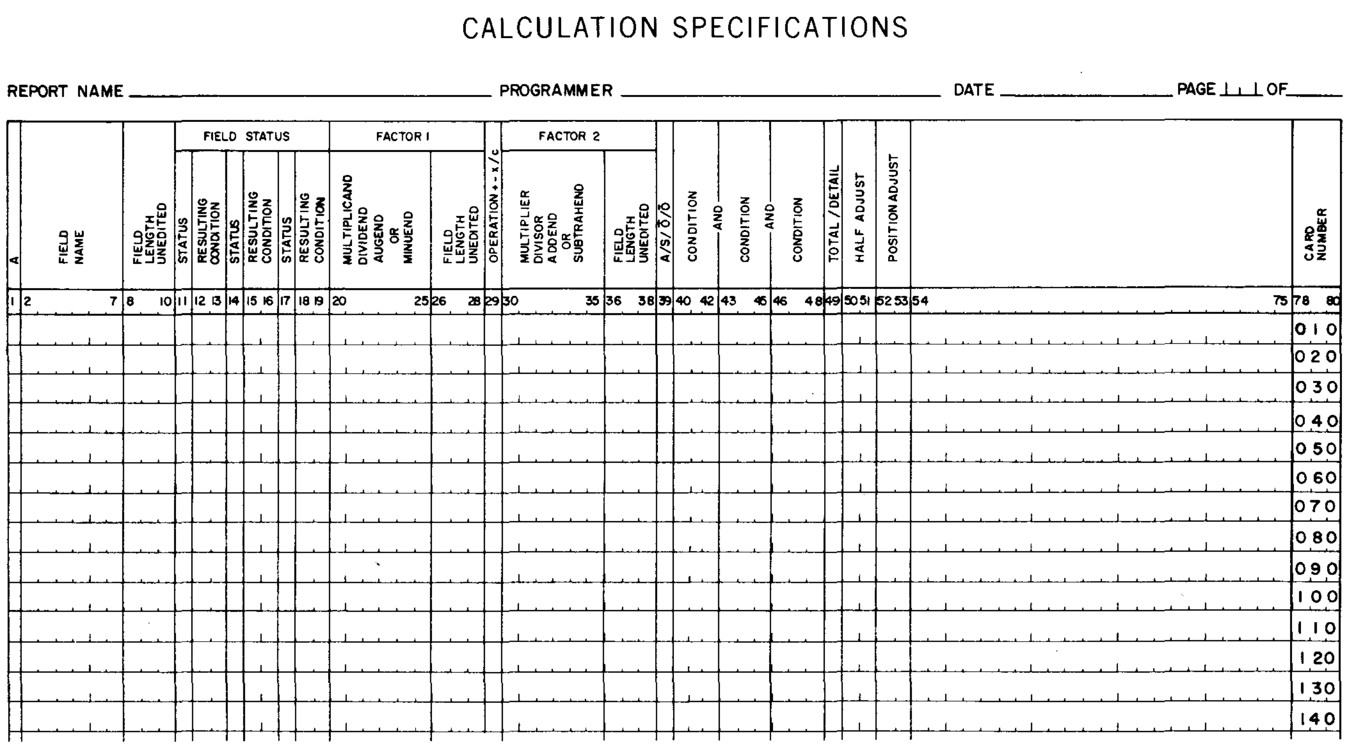

Figure 2-7. Series 200 RG Calculation Specifications Sheet

The various divisions of the Calculation Specifications sheet (Figure 2-7) describe the operations to be performed and the limits which govern these operations. The parameters include: field name, length, and status; the operation to be performed and the factors involved in performing the operation; up to three conditions which govern whether the calculation will be performed; line-entry type (i.e., total or detail); and the particulars of half-adjust and position-adjust of the calculation result. The line entries for the various parameters shown on the Calculation Specifications sheet are made in the manner described below and as further explained in Table 2-3.

1. Field Name. The field name specified on the Calculation Specifications sheet is that which will contain the result after the calculation is performed. The name must be composed of all alphabetic characters; if less than six characters are used, the name must be left-justified. The field name must be blank for a comparison operation and may also be left blank if the last data field specified as the result of a preceding calculation is affected by the present calculation.

2. Field Length. The field length required here is the unedited length of the calculation field. The unedited field length does not include the periods, commas, and other punctuation symbols which are generated automatically by an edit instruction and which appear in the printed report output. Normally, punctuation is not contained in the field. However, when an item card has punctuation punched in it and it is desired that the report output be exactly as punched, the unedited field length is the total spacing, <u>including</u> punctuation.

CALCULATION SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE __ OF ______

A	Field Name	Field Length Unedited	Field Status: Status	Resulting Condition	Status	Resulting Condition	Status	Resulting Condition	Factor 1: Multiplicand Dividend Augend or Minuend	Field Length Unedited	Operation + - x / c	Factor 2: Multiplier Divisor Addend or Subtrahend	Field Length Unedited	A/S/$\overset{+}{0}$/$\overset{-}{0}$	Condition	AND Condition	AND Condition	Total / Detail	Half Adjust	Position Adjust		Card Number
1	2 7	8 10	11	12 13	14	15 16	17	18 19	20 25	26 28	29	30 35	36 38	39	40 42	43 45	46 48	49	50 51	52 53	54 75	78 80
A	FIELDA	ØØ6							FIELDB	ØØ5	+	FIELDC	ØØ3	A				D				010
A	FIELDD	ØØ4							FIELDE	ØØ4	−	FIELDF	ØØ3	S				D				020
A	FIELDG	ØØ8							FIELDH	ØØ5	X	FIELDI	ØØ3	$\overset{+}{0}$				D				030
A	FIELDJ	ØØ6							FIELDK	ØØ6	/	FIELDL	ØØ2	$\overset{-}{0}$				D				040
A			E	18					FIELDA	ØØ6	C	1Ø9987	ØØ6					D				050
A	TOTTAX	ØØ6										WHTAX	ØØ5	A				T				060
A	NETAMT	ØØ7										TOTTAX	ØØ6	S				T				070

Figure 2-8. Example of Calculation Specifications Sheet Entries

3. Field Status. Status of the calculation field is used to control processing which is performed at a later time, according to the limits specified by format specifications or by subsequent calculation specifications. In addition to the four possible status meanings of the data specifications (blank, zero, negative, or positive), the calculation specifications permit comparison of the two fields described by Factors 1 and 2 to determine whether the second is equal to, unequal to, higher or lower in the collating sequence than the first. As in the case of the data-specifications field status, a two-digit decimal number which assigns a resulting condition is used to identify the field status character. Any number may be used to specify this condition, provided that it is not a number previously used to define a resulting condition on the Input or Data Specifications sheets. The resulting condition, when so defined, may then be used to govern other operations in the same manner as in the case of the data specifications.

4. Operation Factors. The calculation operation which is to be performed is defined in terms of the factors which are to be manipulated (multiplicand and multiplier, or dividend and divisor, etc.), the manner in which the calculation is to be performed (multiply, divide, etc.), and the way in which the calculation result is to be handled.

Handling of the calculation results may be accomplished in several ways. The result may be added to, subtracted from, or substituted for the contents of the field described in the columns titled "Field Name" of the particular line entry. Additionally, the operations involved in comparison of fields and in obtaining progressive totals can be specified at this time. These operations are illustrated in the example of Calculation-Specifications sheet entries shown in Figure 2-8.

The entries shown in Figure 2-8 are explained as follows:

Line entry 1: This line entry specifies an addition (+) operation.[1] The sum of FIELDB plus FIELDC is to be added (A) to the contents of FIELDA and the resulting sum placed in the location labeled FIELDA.

Line entry 2: A subtraction (-) operation is specified by this line entry; the difference of FIELDE minus FIELDF is to be subtracted (S) from the contents of FIELDD and the resulting difference placed at FIELDD.[1]

Line entry 3: A multiplication calculation (X) is specified in the operation described by this line-entry. FIELDH is to be multiplied by FIELDI and the product reset-added ($\overset{+}{0}$)[2] into FIELDG. In the reset-addition, FIELDG is reset to zero before the product is added to it.

Line entry 4: This line entry specifies a division (/) operation wherein FIELDK is divided by FIELDL and the quotient is reset-subtracted ($\bar{0}$)[2] from FIELDJ. In the reset-subtraction portion of the operation, FIELDJ is reset to zero before the quotient from the division operation is subtracted from FIELDJ.

Line entry 5: A comparison operation is specified by this line entry. The literal 109987 will be compared to FIELDA to determine whether the two factors are equal. FIELDA is defined by a previous entry on this Calculation Specification sheet. If it is not so defined, it must have been defined by a Data-Specification sheet entry in order to be used in this comparison. The comparison has been arbitrarily assigned resulting condition number 18; an equal result of the comparison can then be referenced in other operations by this resulting condition number.

Line entries 6 and 7: These two lines are an example of a progressive total operation. In this example, line entry 6 specifies that the field WHTAX be added to the contents of the field TOTTAX and the sum be placed in the TOTTAX field. Line entry 7 specifies subtraction of the TOTTAX field from the NETAMT field

---

[1] Refer to Appendix A, Series 200 Character Codes.

[2] $\overset{+}{0}$ is represented by an R-, 0-punch; $\bar{0}$ is represented by an X-, 0-punch when the Calculation Specifications sheet entries are punched into cards.

with the resulting difference to be placed in the NETAMT field. In both entries, the OPERATION and FACTOR 1 columns are left blank; the calculation is controlled by positioning the entry in the FACTOR 2 columns and specifying placement of the result.

5. Conditions. The conditions which govern whether the calculation will be performed, up to a maximum of three conditions, can be specified in accordance with the following rules. These conditions are treated in an AND relation; if more than one condition is specified, all conditions must be present for the calculation to be performed. Any resulting conditions which have been previously defined on the Input, Data or Calculation Specifications sheets can be specified here. In addition, last card, control-field change, and sense-switch settings, as well as negations of any of these, may be specified as resulting conditions, or the columns can be left blank if the calculation is to be performed unconditionally. The Total or Detail designation described in paragraph 6 partially conditions the calculation. If, for example, every line entry on the Calculation Specification sheet defines a detail calculation which will be performed for every input file record, the calculation could be conditioned only by a D entry in the Total/Detail column. Since this is not generally the case in practice, many detail calculations and all total calculations must be further controlled by specifying the condition(s) governing whether the calculation will be performed. Preferably, all detail-time calculations should be specified in consecutive order, with all of these calculations that are governed by the same conditions placed together. The same order for total-time calculations should also be preserved.

6. Total or Detail. Each line entry of the Calculation Specifications sheet must contain either a T or a D entry in the appropriate column to indicate whether a total or a detail calculation is to be performed by the line entry. Either type of calculation will be performed after the test for a control-field change has been made, but the total calculation is made before information is removed from the input record, while the detail calculation is made only after information removal. This restriction is necessary to prevent information from the record just read being included in the calculation results when calculation performance is intended only at total time. Detail-time calculations will include in their results the information removed from the record just read. Control of the calculation is governed in part by the T or D entry, but additional control must be supplied by the condition(s) explained in the preceding paragraph.

7. Half-Adjust and Position-Adjust. Half-adjusting is the process of rounding the last decimal position of the desired result, while position-adjusting is the process of dropping unwanted digits so that the result will contain only the desired number of significant digits. In the case of either half-adjusting or position-adjusting, the low-order position of the result is designated by a 01 entry in the appropriate columns if it is intended to adjust that position. The next position to the left is designated 02, the next 03, etc.

   Thus, if the six-place result 123456 were to be carried as a four-place rounded result, 02 would be entered in both the half-adjust and position-adjust columns. The result carried would thus be 1235 (five added to the second position of the six-place result and the two low-order positions dropped), which is the desired four-place rounded result. This could represent a dollar amount of $12.35, an example in which it is desired to drop insignificant digits and round off the low-order decimal position (hundredths).

   Either adjust operation, or both, may be performed independently of all other calculations.

A table of values showing the number of positions to half-adjust or position-adjust in order to obtain a two-decimal place product which has been rounded to the nearest hundredth is presented in Table 2-4. The values in the table may be used to determine the number of positions which must be specified in order to obtain a product of any desired number of decimal places (providing that Factors 1 and 2 contain sufficient decimal positions to generate the required number of decimal places before adjusting is performed). To obtain a one-decimal product rounded to the nearest tenth, add 01 to the value determined from the table before entering the value in the half-adjust position-adjust columns of the Calculation Specifications sheet. To obtain a three-decimal place product rounded to the nearest thousandth, subtract 01 from the table value; for a four-decimal product rounded to the nearest ten-thousandth, subtract 02, etc.

Table 2-4. Half-Adjust and Position-Adjust Values

Number of Decimal Places in Factor 2	Number of Decimal Places in Factor 1: 0	1	2	3	4	5	6
0	00	00	00	01	02	03	04
1	00	00	01	02	03	04	05
2	00	01	02	03	04	05	06
3	01	02	03	04	05	06	07
4	02	03	04	05	06	07	08
5	03	04	05	06	07	08	09
6	04	05	06	07	08	09	10

To use the table:

1. Read down the column which is headed by the number of decimal places present in the Factor 1 being used.
2. Read to the right across the column which is headed by the number of decimal places present in the Factor 2 being used.
3. Select the value found at the intersection of the two columns specified in steps 1 and 2, above.

As an example:

Multiply the two numbers, 100.1001 (Factor 1) by 10.001 (Factor 2). The product, 1001.1011001, shall be rounded to the nearest thousandth.

Factor 1 has 4 decimal places, Factor 2 has 3 decimal places; the table value at the intersection of columns 4 and 3 is 05 (nearest hundredth) from which 01 is subtracted to obtain a value for a product of three-decimal places rounded to the nearest thousandth. Inserting the derived value 04, then, in the half-adjust and position-adjust columns of the Calculation Specifications sheet will yield the desired product, 1001.102.

When performing calculations, it is permissible, under certain circumstances, to leave both the field name and the field length blank. For example, consider a hypothetical situation concerning commissions paid to salesmen. It is desired to compute weekly commissions for each member of the sales staff and have this information printed out in report form for administrative purposes. The manner in which commissions are to be calculated is set up as follows:

$$\text{Salesman's commission} = \left[\text{gross sales revenue} - (\text{sales closed} \times 11.25)\right] \times 7.25\%$$

This formula states that the salesman's commission is 7 1/4% of the adjusted gross revenue he produced. The adjusted gross is obtained by deducting from the gross sales revenue an amount which is derived by multiplying the number of sales closed by a "profit factor," in this case, 11.25. The figure 11.25 represents an average, based on all sales for the period, which management has assigned as gross profit per item.

The Calculation Specifications sheet line entries which specify performance of the necessary calculation operations are shown in Figure 2-9. Note that all calculations will be performed at total-time (T in column 49) and upon a change in control field 1 (in this case, salesman number). A change in control field 1 initiates the sequence of calculations shown here, and is a condition for the performance of each calculation in the sequence.

CALCULATION SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE 1 OF ______

A (1)	FIELD NAME (2-7)	FIELD LENGTH UNEDITED (8-10)	FIELD STATUS: STATUS (11)	RESULTING CONDITION (12-13)	STATUS (14)	RESULTING CONDITION (15-16)	STATUS (17)	RESULTING CONDITION (18-19)	FACTOR 1: MULTIPLICAND DIVIDEND AUGEND OR MINUEND (20-25)	FIELD LENGTH UNEDITED (26-28)	OPERATION + - x/÷ (29)	FACTOR 2: MULTIPLIER DIVISOR ADDEND OR SUBTRAHEND (30-35)	FIELD LENGTH UNEDITED (36-38)	A/S/0̄/0 (39)	CONDITION (40-42)	AND CONDITION (43-45)	AND CONDITION (46-48)	TOTAL/DETAIL (49)	HALF ADJUST (50-51)	POSITION ADJUST (52-53)	(54-75)	CARD NUMBER (78-80)
A	COMMSN	ØØ7										GROSS	ØØ7	+0	F1			T				010
A			N	Ø7					SLSCL	ØØ3	X	1125	ØØ4	S	F1			T				020
A									COMMSN	ØØ7	X	725	ØØ3	+0	F1	NØ7		T	Ø4	Ø4		030

Figure 2-9. Calculations With Field Name and Length Blank

The entries shown in Figure 2-9 are explained as follows:

Line entry 1: This line entry specifies that the seven-position field, COMMSN, shall be set to zero and the seven-position amount, GROSS, shall then be added to the COMMSN field. The operation shall be performed at total-time, and upon a change in control field 1.

Line entry 2: The second line entry specifies multiplication of the three-position factor 1, SLSCL, by the literal 1125 and subtraction of the product obtained from the COMMSN field. Note here that neither the field name nor its length are specified. Both are left blank, since the last data field specified as the result of a preceding calculation is affected by the present calculation. After the subtraction operation has been performed, the sign of the COMMSN field is tested; if the field is negative (indicating that the amount by which gross revenue is to be adjusted is greater than the original amount of gross sales revenue), then the resulting condition 07 is fulfilled and the corresponding indicator will be set to ON.

Line entry 3: This line entry specifies the final computations in the sequence of calculations. The calculations specified in line entry 3 are performed as specified, providing that resulting condition 07 is not fulfilled. The contents of the COMMSN field are multiplied by the literal 725, producing a ten-position product. The last four positions of the product are dropped (04 position-adjust) after the half-adjust operation has been performed. The seven-position COMMSN field is set to zero and the adjusted, six-position product added to it. The desired answer, the salesman's commission computed to the nearest dollar, now is in its correct place in the COMMSN field, ready to be utilized as output in the printed report.

Format of Output Specifications

The output specifications detail the conditions under which the information defined by the other specifications sheets will become report output. The output specifications define two major categories of the report format, line and field.

Each line of the report output must be described in terms of its type, level, its order within the sequence of output, identification of the next line, the number of vertical spaces to space and skip before and after the output operations if printed output is desired, the stacker which will be used if punched-card output is desired, and the conditions which govern whether the line will become output.

For each field within a line which is used for output, a line entry must be made on the Output Specifications sheet to define the field name, the position it will occupy in the line of output, conditions which govern placing of the field in the line before printing, whether suppression of leading zeros is to be performed, the edited length of the field, and any additional information which controls editing of the output.

Figure 2-10 illustrates the Output Specifications sheet and Table 2-5 presents the format for preparing output specifications, with column-by-column explanations of the entries.

Table 2-5. Format of Output Specifications

Parameter	Columns	Contents	Explanation
CARD ID	1	L, F, B, K, or W	Used to identify an output specification. Codes are as follows: L - entry describing a line; F - a field entry which is not blanked after output; B - a field entry which is blanked after output; K - a field entry which uses a literal constant; W - a field entry that defines a constant or an edit control word.
LINE SPECIFICATIONS	2 - 28	See below	These columns are used for description of line specification entries.
Identification	2 - 4	Three characters	Line-identification code for this line (taken from the printer spacing chart) to show line type, level, and number.
Line type	2	H, D, or T	Column 2 must contain an H, D, or T to indicate whether the line entry defines a heading, detail, or total line.
Line level	3	One-digit decimal number or one alphabetic character	Heading and total lines which are in a hierarchy are assigned a number to designate the level to which they belong. Independent heading and total lines which are not related in a hierarchy are assigned alphabetic-level designation. The level of detail lines may be designated by either a numerical or alphabetic character.
Line number	4	One-digit decimal number or one alphabetic character	A numerical designation is used to govern the order of appearance of the line in the output. If the level is numerical, the line number must also be numerical. When the level is alphabetic, the line number may be either numerical or alphabetic as desired.
Printing directions	5*	X or blank	X, if line is to be printed; Blank, if line is not to be printed.
Punching directions	6*	X or blank	X, if line is to be punched; Blank, if line is not to be punched.
Tape writing directions	7*	X or blank	X, if line is to be written on magnetic tape; Blank, if line is not to be written on magnetic tape.
Identification of next line	8 - 10	Three characters	Line identification code of next line if it is to be output under the same conditions

*NOTE: Columns 5, 6, and 7 may not all be blank. An output medium must be specified.

Table 2-5 (cont). Format of Output Specifications

Parameter	Columns	Contents	Explanation
(cont)			as this one. The three-character identification code is derived in the same way as the entry for columns 2-4.
Spacing directions	11 - 14	See below	Used to specify the number of vertical spaces to be made both before and after printing this line.
Space before	11 - 12	01,02,03, or blank	The number of spaces to be made before printing this line.
Space after	13 - 14	Same as columns 11-12	Same as columns 11-12, except for directing spacing after printing.
Skipping directions	15 - 18	See below	Used to designate skipping to carriage tape channel 1-12.
Skip before	15 - 16	01-12 (channel number)	Causes skipping to channel 1-12 before printing this line.
Skip after	17 - 18	Same as columns 15-16	Same as columns 15-16, except for causing skipping after printing. Note: If columns 13, 14, 17, and 18 are all left blank, it is implied that the Report Generator is to space one line after printing. Therefore, if output is to be in the form of tape without spacing control characters, columns 13 and 14 must equal 00.
Stacker select	19	X or blank	Used to specify availability of stacker-select option as follows: X - if the card punch has the stacker-select option installed and punched output is desired in the middle (8/2) stacker. Blank - if no stacker-select option is installed, or is present but not to be used. Note: An X in column 19 of the Output Specifications for a program to be generated in three-character mode destroys the effect of double buffering the punched output since the punch must be stalled until the card has been punched and the eject order given.
Negate condition	20	N or blank	N, if condition specified in columns 21-22 is to be negated; Blank, if condition is not to be negated.

Table 2-5 (cont). Format of Output Specifications

Parameter	Columns	Contents	Explanation
Line output condition	21 - 22	1P, 0F, LC, F1-F6, S1,* S2, S3, S4, two-digit decimal number or blank	Used to specify a condition for which the line is to become output. Codes are as follows: 1P - first page OF - overflow; LC - last card; F1-F6 - control field change; S1,* S2, S3, S4 - SENSE switch settings; Blank output of this line has been conditioned by previous line entry (i.e., this line was previously specified as a next line). Two-digit numbers are resulting conditions defined by previous Input-, Data-, and Calculation-Specification sheet entries.
Second negate condition	23	Same as column 20	Relates to condition in columns 24-25.
Second output condition	24 - 25	Same as columns 21-22	Used to specify a second condition for which this line is to be output.
Third negate condition	26	Same as column 20	Relates to condition in columns 27-28.
Third output condition	27 - 28	Same as columns 21-22	Used to specify a third condition for which this line is to be output.
FIELD SPECIFICATIONS	29 - 47	See below	These columns are used for description of field specification entries.
Identification	29 - 34	Name, WORDxx, or blank	Used to identify the field name, or to name a constant (WORDxx). If this is a literal constant, columns 29-34 must be blank.
Print position	35 - 37	Three-digit decimal number	The print position for the rightmost character position of this field.
Negate condition	38	Same as column 20	Relates to condition in columns 39-40.
Field output condition	39 - 40	Same as columns 21-22	Used to specify a condition for the field to be output.

*NOTE: Use S1 with care; it is the load-and-halt switch.
S1 and S2 are used by generated tape input routines.

Table 2-5 (cont). Format of Output Specifications

Parameter	Columns	Contents	Explanation
Second negate condition	41	Same as column 20	Relates to condition in columns 42-43.
Second field output condition	42 - 43	Same as columns 21-22	Used to specify a second condition for which this field is to be output. When more than one condition is specified, all conditions are treated in an AND relationship.
Third negate condition	44	Same as colum 20	Relates to condition in columns 45-46.
Third field output condition	45 - 46	Same as columns 21-22	Used to specify a third condition for which this field is to be output. When more than one condition is specified, all conditions are treated in an AND relationship.
Zero suppression	47	Z or blank	Z, if field is to have leading zeros suppressed (plus edit control word); Blank, if leading zeros are to print.
FIELD DEFINITION, CONSTANT, OR EDIT CONTROL WORD	48 - 75	See below	Used to define a field which is a constant or to specify the edit word to be used when the field is output. These columns are left blank if neither applies.
Field length	48 - 50	Three-digit decimal number	Length of either an edit control word used with the field or a constant defined in columns 51-75.
Definition	51 - 75	Constant or edit control word	The definition is left-justified, and specifies the edit function to be performed or the constant to be used in the output operation.
PAGE NUMBER	76 - 77	Two-digit decimal number	Page number of this specification sheet.
CARD NUMBER	78 - 80	Three-digit decimal number	Card number (preprinted on the specification sheet).

OUTPUT SPECIFICATIONS

REPORT NAME ______________ PROGRAMMER ______________ DATE ______________ PAGE ___ OF ___ (76 77)

FORMAT	LINE: TYPE	LINE: LEVEL	LINE: NUMBER	OUTPUT: PRINT	OUTPUT: PUNCH	RESERVED	NEXT LINE: TYPE	NEXT LINE: LEVEL	NEXT LINE: NUMBER	SPACE: BEFORE	SPACE: AFTER	SKIP: BEFORE	SKIP: AFTER	STACKER	LINE OUTPUT CONDITIONS: CONDITION	—AND— CONDITION	—AND— CONDITION	FIELD NAME	FIELD END	FIELD OUTPUT CONDITIONS: CONDITION	—AND— CONDITION	—AND— CONDITION	ZERO SUPPRESS	FIELD LENGTH	CONSTANT OR EDIT CONTROL WORD	CARD NUMBER
1	2	3	4	5	6	7	8	9	10	11 12	13 14	15 16	17 18	19	20 22	23 25	26 28	29 34	35 37	38 40	41 43	44 46	47	48 50	51 75	78 80
																										010
																										020
																										030
																										040
																										050
																										060
																										070
																										080
																										090
																										100
																										110
																										120
																										130
																										140
																										150
																										160
																										170
																										180
																										190
																										200

Figure 2-10. Series 200 RG Output Specifications Sheet

The Output Specifications sheet (Figure 2-10) is divided in such a way that the operations required for producing report output will be fully defined in the appropriate order. The line entries for the various parameters of the Output Specifications sheet are made in the manner described below and as further explained in Table 2-5.

1. Identification. Each line entry is identified by a character which defines it as an output specification and further specifies the kind of output. The identifying character further distinguishes entries describing an output line from other entries which describe various types of fields that may be blanked after output.

2. Line Classification. An Output-Specifications-sheet line entry which describes an output line must be defined to show type, level, and number of that line. Line type is used to identify and differentiate between heading, total, and detail lines. The level of the line and its numerical order within the given level must also be specified. Line level defines the hierarchical relationship (numerical level) in which lines are to become output. Within

levels there may be a group of lines, each of which is also positioned in an established numerical order; these lines are identified by line number on the Output Specifications sheet. It is possible, then, to have a numerical hierarchy in the level of heading or total lines which will be printed, as well as a number of lines within each level. The line classification entries are used to specify this sequence. Line entries which specify these details must reflect an increasing numerical order for heading lines within the same level and a decreasing numerical order for total lines within the level. Entries on the Output Specifications sheet which define hierarchical output lines must be in the same order as the order in which these lines will be printed on the report.

3. Line Output. Each line entry on the Output Specifications sheet which specifies an output line (L in column 1) must also specify the output media. Columns 5, 6 and 7 are used to specify printed, punched and magnetic tape output, respectively. An alphabetic X is entered in the appropriate column to specify the output media desired; leaving an individual column blank results in no output for that media. By entering an alphabetic X in two or three of the columns, report output in more than one media at a time can be obtained.

4. Next Line. As the line entry is made for each output line, the next line of output which will follow unconditionally is referenced. If no lines follow, or if following lines will become output only on fulfillment of certain conditions, these columns are left blank for the Output-Specifications line entry being made. Any next line which is specified must be of the same type and level as the line defined in the present line entry. Also the next line so specified must have its output conditions (columns 20 through 28) blank.

5. Space and Skip. The number of vertical lines to be spaced before printing the line (if printed output has been specified), as well as the number of lines to be spaced after printing, are each designated by 01 for single spacing, 02 for double spacing, or 03 for triple spacing (no more than triple spacing may be performed). Line entries are also made to provide for skipping to carriage tape channels 1-12 both before and after printing the line being specified.[1] When headings appear at the top of each page of a report, provisions are generally made in the output specifications to control skipping from the end of one form to the head of the next when overflow conditions occur. In simpler types of reports, where page headings are not printed on each page, skipping can be controlled by the carriage control tape, obviating the necessity for specifications control.

   To illustrate: When printing is being performed and the object program senses a punch in carriage tape channel 12, automatic skipping to channel 1 occurs. In the event that a 12-punch in the carriage control tape is sensed while a total line is printing, overflow skipping will not take place until all total lines whose output conditions are met have been printed.

6. Line Output Conditions. A maximum of three conditions which will govern output of the line may be specified. Conditions which are specified on the same line entry are considered in an AND relation; the line will be printed only when all conditions are present and are fulfilled. The conditions which may be specified are any of the resulting conditions previously defined on any of the other specifications sheets plus those shown in Table 2-5. There

[1] If the printer being used does not require a carriage tape, refer to "Carriage Control Card Format."

can be no output conditions for a line entry which has been specified by a previous entry as a next line, because the line will print only under the same conditions as the previous line entry. Printing of any line which is specified on the output specifications sheet can take place only if one of the following situations exists when the line is called in by the object program:

a. The line entry which defines an output line must contain the condition(s) under which this line shall be printed, and the condition(s) specified must be met.

b. This line entry defines an output line which has been specified as a next line by a previous line entry on the Output Specifications sheet. Further, the output conditions of the previous line must have been fulfilled, in order for this next line to print. No conditions are specified for this present line entry; it is governed by the conditions specified in the previous line entry which referenced it as a next line.

c. The line entry defines an output line which is in a hierarchy and which is of lower level than another line of the same type whose output conditions have already been met.

There are certain conventions which must be observed in specifying line-output conditions, since the conditions are restrictive. Incorrect usage can prevent printing of certain lines when it is actually desired to have these lines printed. These conventions are explained below.

a. Overflow (OF) conditioning of a line restricts the connection between this and other lines, in that the line can neither specify a next line which is not also conditioned by OF, nor can it be specified itself as a next line by any line not similarly conditioned.

b. All output lines which are related in a hierarchy have to be of the same type. In addition, their relative positions in the hierarchy must be shown by numerical-level designations. Since the output of lower-level total lines is governed by lines of a higher level within a hierarchy, the object program will force the higher-level lines to be printed first, no matter what output conditions are specified for the lower-level lines. The opposite relationship applies to header lines.

c. When several lines appear within a particular level of a hierarchy, each line of output subsequent to the first must be designated by a next-line reference to ensure that all lines of the hierarchy are present in the output. All lines must, of course, have the same output condition(s). If line-output conditions vary within the same level, some lines may be deleted in the printing operation since they cannot be called in by a next-line reference when higher-level lines force lower-level lines to follow in the printed output. To preclude such an occurrence, the line levels can be considered in a non-hierarchical relationship by assigning alphabetic-level designations. Several levels of output lines would then be independent of each other and would print in accordance with the conditions specified for each. When several output-line levels are present whose conditions are fulfilled at the same time, the lines will be printed in the same order as that in which they appear on the Output Specifications sheet. Thus, with alphabetic designation of line levels, ascending or descending order of lines in the output can still be maintained. However, specifying output in this manner results in a sacrifice of speed in object-program performance. In certain cases, it may not be possible to disregard conditions which govern output of lower-level lines when the conditions for higher-level lines within the same type are met. In this event, alphabetic-level designations will have to be used so that the conditions of each line level will be checked, although run-time speed will be reduced as a direct result.

d. First page (1P) conditioning specified by the output specifications causes printing which appears on a cover sheet or as page heading lines on the first page of the report. The page heading lines can be produced on following pages by specifying an overflow (OF) condition. The first page condition is satisfied before any input records have been read into the input area. Therefore, only lines which contain constant information may be produced as output when a first page condition is specified. If information contained in the first input record is needed to complete the first page output, the line should be conditioned by the resulting-condition number which represents the required information. If, on the other hand, the line containing the information is conditioned by first page (1P), the information will not be produced in the first page of output. When specifying more than one line of output, only the first line entry need contain the 1P condition in columns 21 and 22. All following lines shall be indicated as next lines on the Output Specifications sheet.

e. In those instances where it is desired to have output produced in accordance with varying line-output conditions, the conditions must be specified in an OR relation. OR conditions are specified by making a separate line entry for each of the varying conditions which govern the output. The first line entry contains all the usual information required for definition of an output line and includes the first of the varying line-output conditions. When more than one output condition is specified on the first line entry, the conditions will have an AND relation. The additional OR conditions are specified, one by one, on subsequent line entries which contain an identifying L in column 1 and one of the varying conditions in columns 20-22. Line entries 1, 2, and 3 of Figure 2-11 on page 2-41, show a detail line called DBB which shall be printed as output upon fulfillment of any one of three different conditions. The line will be printed when: (1) condition 03 is met; or (2) when condition 05 and not condition 06 is met; or (3) when condition 07 is met. A limitation placed on the specification of varying conditions for line output is that if one line entry in the OR relation is governed by overflow (OF), all output-specifications line entries which are part of the OR relation must be conditioned by OF. Line entries 16, 17, and 18 of Figure 2-11 illustrate overflow (OF) as a line output condition when an OR relation also exists. In this instance, the line H34 will be printed as an overflow line whenever conditions 09, 11 or 12 are met.

7. Field Description. After the line which is to become output in the report and the conditions which govern its output are specified, any field or fields to be inserted in the line are defined on subsequent line entries. As shown in Table 2-5, field specifications may define data fields that may or may not be blanked after being placed in the output line. Field specifications may also identify entries which either use a constant or define a constant or an edit control word. Field names are left-justified and must be entered in accordance with the following restrictions:

a. An output-specifications line entry which contains an F in column 1 (field specification for a data field which is not to be blanked after output) must name either a data field from the Data or Calculation Specifications sheets or a WORDxx defined later by a W-entry on the Output Specifications sheet.

b. An output-specifications line entry which contains a B in column 1 (field specification for a data field which is blanked after output) must name a data field from the Data or Calculation Specifications sheet.

c. An output-specifications line entry which contains a K in column 1 (field specification that uses a constant) must not contain a field name; these columns are left blank. The constant being used is then stated in the area which is titled "Constant or Edit Control Word" (see paragraph 10 of this discussion).

d. An output-specifications line entry which contains a W in column 1 (field specification which defines a constant or an edit control word) must name a field, WORDxx, where xx is a number, unique to each W-entry, in the range from 00 to 99. (See paragraph 10 of this discussion.)

The end of the field being specified by the above entries is shown by a three-digit decimal entry immediately after the Field Name columns, except for a W-entry. This number shows the end (rightmost) position of the field as it will appear in the output, and is derived from the Report Layout chart. If the line entry is a field specification identified as a W-entry, the field end columns must be left blank.

8. Field Output Conditions. A maximum of three conditions which govern placing the specified field into the output line can be specified. When more than one condition is specified per line entry, the conditions have an AND relation, i.e., all specified conditions must be satisfied in order for the field to be placed in the output line. However, it is also possible to consider varying field-output conditions in an OR relation by specifying each OR condition as a separate line entry, in exactly the same manner as described under line-output conditions (paragraph 6). Each line entry which defines a field specification must contain all the information required to describe the field, plus one of the varying conditions which are part of the OR relation. As in the case of varying line-output conditions, when one line-entry for a field specification contains more than one condition, these are considered to have an AND relation, but subsequent line entries describing the field and a condition governing its output are considered in an OR relation.

9. Zero Suppression. Zero suppression may be specified for a line entry describing a field by entering an alphabetic Z (0- and 9-punch) in the zero suppression column (col. 47) of the Output Specifications sheet, plus an edit control word entry. Entry of the Z alone will cause zone suppression of zeros in all except the units position. The form of the edit control word must be in accord with the MCE (Move Characters and Edit) instruction. Editing will be performed as follows:

> Data is transferred from the field source specified, character by character, from right to left. The zero (0) symbol in the edit control word specifies zero suppression. Its location in the control word is interpreted as the rightmost limit of zero suppression. The zero-suppression symbol will be replaced by the character

from the corresponding position of the source field. If a zero-suppression symbol is not sensed in the edit control word, the edit operation terminates at the leftmost position of the source field. A zero-suppression symbol causes the edited result field to be scanned from left to right. During this scan, high-order zeros and commas are automatically replaced by blanks (unless an asterisk or a dollar sign appears immediately to the left of the zero-suppression symbol). Zero suppression is terminated by any of the following:

a) a decimal digit from one through nine,

b) a decimal point, or

c) the location that initially contained the zero-suppression symbol.

An asterisk immediately to the left of the zero-suppression symbol in the edit control word causes high-order zeros and commas to be replaced by asterisks instead of blanks in a zero-suppression operation. High-order blanks are also replaced by asterisks.

A dollar sign immediately to the left of the zero-suppression symbol in the edit control word is replaced with a source field character and causes the edit result to be rescanned following the zero-suppression operation. During this scan, the dollar sign is "floated" to the left of the high-order significant digit in the edited result.

OUTPUT SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE ___ OF ___ (76 77)

FORMAT (1)	LINE TYPE (2)	LINE LEVEL (3)	LINE NUMBER (4)	OUTPUT PRINT (5)	OUTPUT PUNCH (6)	RESERVED (7)	NEXT LINE TYPE (8)	NEXT LINE LEVEL (9)	NEXT LINE NUMBER (10)	SPACE BEFORE (11-12)	SPACE AFTER (13-14)	SKIP BEFORE (15-16)	SKIP AFTER (17-18)	STACKER (19)	LINE OUTPUT CONDITION (20-22)	AND CONDITION (23-25)	AND CONDITION (26-28)	FIELD NAME (29-34)	FIELD END (35-37)	FIELD OUTPUT CONDITION (38-40)	AND CONDITION (41-43)	AND CONDITION (44-46)	ZERO SUPPRESS (47)	FIELD LENGTH (48-50)	CONSTANT OR EDIT CONTROL WORD (51-75)	CARD NUMBER (78-80)
L	D	B	B	X											Ø3											010
L															Ø5	NØ6										020
L															Ø7											030
L	H	3	4	X							Ø3				OR	Ø9		Cols. 24 & 25								160
L															OR	11										170
L															OR	12										180

Figure 2-11. OR Conditions for Line Output

10. Constant or Edit Control Word. Both the length and a full definition of each constant and of each edit control word used on the Output Specifications sheet must be stated. When a particular constant is to be used only once in generating a report, the line entry format is: a K in column 1; entries only in the field-end and field-length columns; and the constant itself (written in the exact form in which it is to appear in the printed output), entered in the proper area (left-justified in columns 51-75). Any edit control word which is to be used only once in a particular

report is entered in the proper area (left-justified in columns 51-75) of the line entry which specifies the field to be edited. When a constant or an edit control word is used more than once in the Report Generator specifications, it should be defined by a W-entry on the Output Specifications sheet. This entry produces a more efficient object program and also permits a word which is so defined to be used as a calculation factor in specifying a literal larger than the six-character limit imposed by the calculation specifications. A W-entry must contain WORDxx as a field name, the field length, and either the word itself (exactly as it will appear in the output if a constant) or the edited format (if an edit control word). A WORDxx field name denotes that xx will be a number in the range from 01 to 99 which is unique to this W-entry. A W-entry to define each WORDxx used must be made on the Output Specifications sheet. Since a W-entry defines a word which will be used elsewhere on either the Output or the Calculation Specifications sheets, the defining line entry may appear anywhere on the sheet except for the first line-entry which must be used for an L-entry (an output specification which describes a line of output).

The format which can be used in specifying an edit control word must be in agreement with the Series 200 editing rules used in an MCE (Move Characters and Edit) instruction. The rules for zero suppression are stated in paragraph 9, above. Other edit instruction rules are stated in the Honeywell Series 200 Programmers' Reference Manuals. Since all cards are read in the Honeywell mode, if a "+" is to be used in the edit control word for the purpose of being replaced with a blank, it must be punched on the specifications cards as an R-, 0-punch, as required by the MCE instruction. Note also, that a decimal control feature similar to the 1401 series Expanded Print Option is not available for the Model 200 at present; therefore, decimal points will be printed on an output line even if the field is blank.

Handling Line Entries

The handling of various line entries on the Output Specifications sheet is illustrated in Figures 2-12 and 2-13, which show a hypothetical report example summarizing the preceding discussion. Several arbitrary assumptions have been made for the purpose of illustrating this example:

1. A 132-position printer will be used and the forms for the report are of standard 15-inch width.
2. The report heading will be centered at the top of each page of the report.
3. Certain exceptions, for the purpose of simplifying the illustrations, have been made in the line entries. These consist mainly of omissions which are pointed out in the explanation below pertaining to Figure 2-13.

The format desired for the hypothetical report example is depicted in Figure 2-12. Note that the Division Number and Department Number lines have no codes assigned to them. These will be considered to be level 2 and level 1 lines, respectively. To simplify this illustration, the output specifications for these and certain other lines have been deleted on both the Report Layout chart and in the output specifications.

REPORT LAYOUT

PROGRAM ______ PROGRAMMER ______ DATE ______ PAGE ______ OF ______ REPORT NAME ______

COMMENTS ______

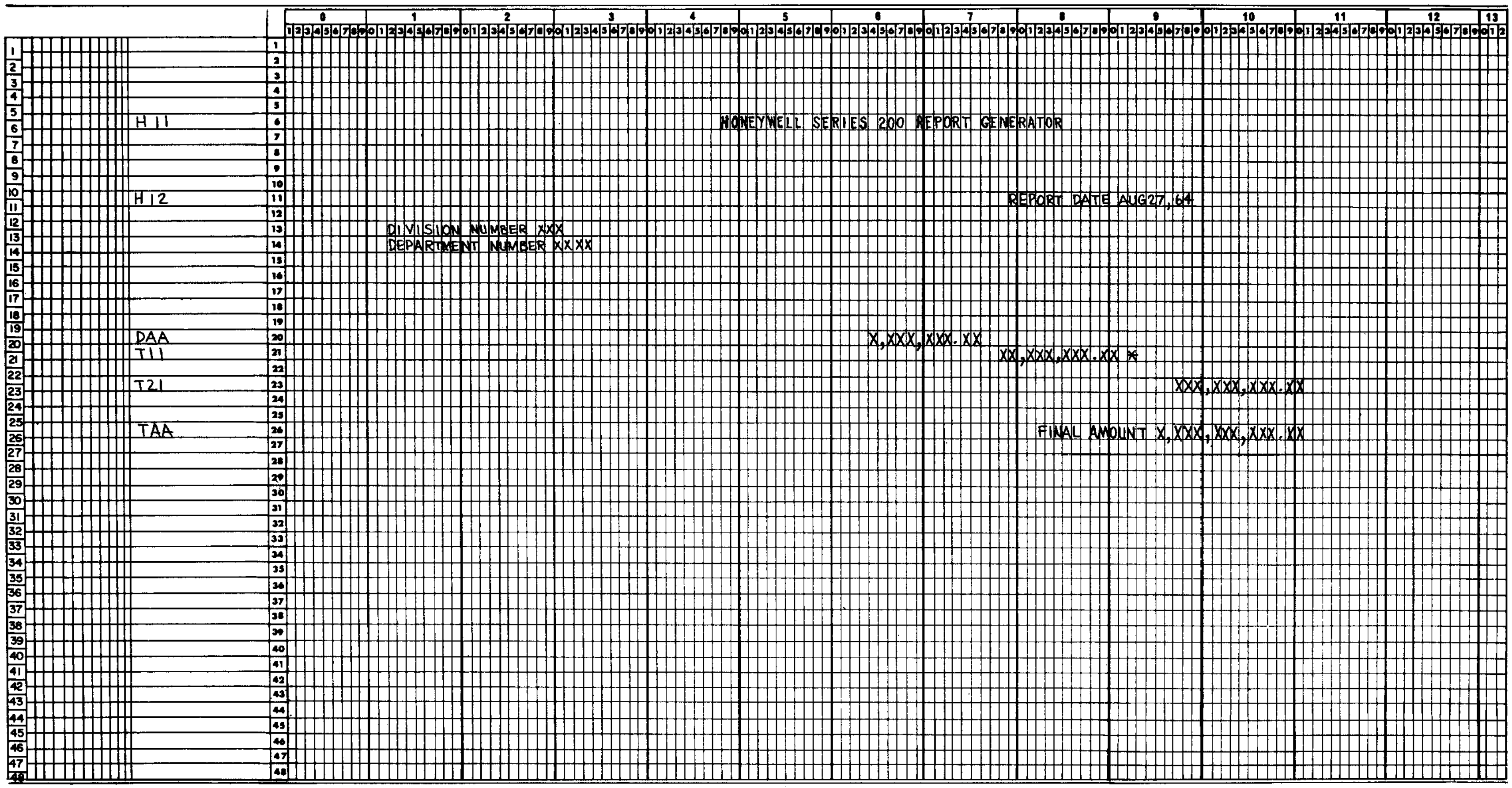

Figure 2-12. Example of a Series 200 RG Report Layout

OUTPUT SPECIFICATIONS

REPORT NAME ____________ PROGRAMMER ____________ DATE ____________ PAGE __ (76 77) OF ____

Format (1)	Line Type (2)	Line Level (3)	Line Number (4)	Output Print (5)	Output Punch (6)	Reserved (7)	Next Line Type (8)	Next Line Level (9)	Next Line Number (10)	Space Before (11-12)	Space After (13-14)	Skip Before (15-16)	Skip After (17-18)	Stacker (19)	Line Output Condition (20-22)	-And- Condition (23-25)	-And- Condition (26-28)	Field Name (29-34)	Field End (35-37)	Field Output Condition (38-40)	-And- Condition (41-43)	-And- Condition (44-46)	Zero Suppress (47)	Field Length (48-50)	Constant or Edit Control Word (51-75)	Card Number (78-80)
L	H	3	1	X			H	3	2			Ø1	Ø2		F2											010
K																			Ø67					Ø20	HONEYWELL SERIES 2ØØ	020
K																			Ø84					Ø16	REPORT GENERATOR	030
L	H	3	2	X									Ø3													040
F																		WORDØ1	Ø89							050
F																		DATE	Ø98							060
L	D	A	A	X							Ø1				Ø1											070
F																		AMOUNT	Ø75					Ø12	Δ,ΔΔΔ,ΔΔØ.	080
L	T	1	1	X							Ø2				F1	NØ2										090
B																		TOTAMT	Ø92					Ø15	ΔΔ,ΔΔΔ,ΔΔØ. b̸*	100
L	T	1	2	X							Ø1				F2	NØ2										110
B																		AMTDIV	11Ø					Ø14	ΔΔΔ,ΔΔΔ,ΔΔØ.	120
L	T	A	A	X						Ø3					LC											130
K																			Ø93					Ø12	FINAL AMOUNT	140
W																		WORDØ1						Ø11	REPORT DATE	150

Figure 2-13. Example of Output Specifications Sheet Entries

The line entries in Figure 2-13 are explained as follows:

Line entry 1 is identified as an output specification for a line by the L in column 1. The first entry on the Output Specification sheet (or first page of the sheets, if more than one is used) must be a line specification. Line entry 1 specifies a line which is defined as belonging to a hierarchy of heading lines; it will be printed upon a change in control field 2 and will be followed by the heading line H32. The control fields may be assumed to be employee, 1; department, 2; and division, 3. Thus, when department number changes, the heading will be printed at the head of the next form. Skipping to carriage-control-tape channel 1 before printing the line and to channel 2 after printing the line is also designated.

Line entries 2 and 3 specify the fields which constitute the heading line H31. These fields are constants which are specified in the Constant or Edit Control Word column. The two line entries will be printed on one line of the report, but since the total length of the heading exceeds the 25 spaces provided for defining the constant, two line entries must be made. The field-end entries show the printer position where the constant will end. "HONEYWELL SERIES 200" will be printed in the 20 spaces from position 48 to position 67, and "REPORT GENERATOR" will occupy the 16 positions from the 69 to 84th printer positions of the same line.

Line entry 4 specifies a second heading line, H32, which will unconditionally follow the heading line H31. Since both of these lines are level 3, there will be lines also of level 2 and level 1, since a hierarchical relationship exists. These lower-level heading lines are not shown in this example in order to simplify the illustration. In an actual report using this kind of output specifications, the descriptions of the lower-level heading lines (in descending order) would follow the line entries which describe level 3 heading lines. Line entry 4 is composed of two fields, one being REPORT DATE (identified by WORD01) and the second being the date itself. These two fields are defined by line entries 5 and 6. Note that WORD01 is itself defined by a W-entry at the end of the output specifications (line entry 15).

Line entry 7 specifies the detail line for the report, which will become output when condition 01 is fulfilled. Condition 01 must have been defined elsewhere, on one of the other specifications sheets. In this example, condition 01 is defined as the resulting condition number assigned to input data records on the Input Specifications sheet. This resulting condition number is assigned for the purpose of governing subsequent operations (in this instance, governing when the line will become output). The field which constitutes this detail line is an amount, defined in line entry 8, whose largest expected total is a 12-position figure in the form specified by the edit control word in this line entry.

Line entries 9 through 12 specify a hierarchical relationship of total lines. The fields which will be printed in the output will be blanked after being placed in the output lines. Line entry 10 defines a subtotal, which will be identified by an asterisk, as specified in the edit control word entry. Both lines specified by line entries 9 and 11 will become output upon fulfillment of two conditions: a control field change occurs, and not condition 02. Condition 02 must have been defined on the Input, Data, or Calculation Specifications sheets. In this instance, condition N02 is taken to mean that the field specified cannot be blank.

Line entries 13 and 14 specify the total line which is governed by the last card (LC) condition and is composed of the constant, FINAL AMOUNT, plus the amount itself, in numerical form. In the output specifications for an actual report, this entry would be followed by a field (F) specification specifying the edit control word for this final amount total, the length of the field, and its end position, in the same manner as shown in line entry 12 on this sheet.

SPECIAL CONSIDERATIONS

Certain considerations must be observed in preparing the Series 200 Report Generator specifications in order to ensure accuracy of the specifications and to make certain that the object program will operate as efficiently as possible. One consideration is that the arrangement of the specifications in certain ways not only results in less execution time being required for the object program, but will also save the amount of core storage needed for program execution. In describing the specifications which make up the Series 200 Report Generator, some of the column entries have not previously been discussed. Since these entries are common to the specifications, or serve similar purposes, they are discussed here as special considerations. Other considerations which are discussed here include those special applications which are either unique in nature or are related to the over-all concept of RG, rather than to a particular type of specification.

Column Entries Common to the Specifications

1. Identification of Entries

All Series 200 Report Generator specifications, whether input, data, calculation or output, contain an identifying letter in column 1 which uniquely identifies the type of entry.

Input Specifications - All input specifications must contain a C (R- and 3-punch) in column 1 for every line entry on the Input Specifications sheet.

Data Specifications - All data specifications must be identified by a D (R- and 4-punch) in column 1 of every line entry on the Data Specifications sheet.

Calculation Specifications - Every line entry on the Calculation Specifications sheet is identified by an A (R- and 1-punch) in column 1.

Note that the input, data, and calculation specifications are each identified by a single letter which must be the same for each line entry on the individual specifications sheet. Output-specifications line entries may be described by a variety of entries in column 1, depending on the function the line entry is to perform. Output-specifications line entries which define a line of output that will appear in the printed report must contain an L (X- and 3-punch) in column 1. If the line entry has the function of describing a field which is within a line or defines a constant, the column 1 entry and its meaning are as follows:

F - (R- and 6-punch) identifies a line entry which is a field specification. This specification defines a data field which will be placed in an output line. The data field will not be blanked after being placed in the output line of the printed report.

B - (R- and 2-punch) also identifies a line entry which is a field specification for a data field to be placed in an output line. Unlike the F-entry, this data field will be blanked after being placed in the output line. This operation is analogous to an accounting machine process of readout and reset total.

K - (X- and 2-punch) identifies a line entry which is a field specification. In this instance, the data field is a constant which will be used in the report output.

W - (0- and 6-punch) this line entry identifies a field specification for a data field which defines a constant. K- and W-entries frequently appear together; when a constant is used more than once in the output specifications, it is advantageous to define the constant with a

W-entry in order to produce a more efficient object program. It is not necessary for the W-entry to immediately follow the K-entry; the definition of the constant may appear anywhere on the Output Specifications sheet, except on the first line (which must be an L-entry).

2. Page Numbering

Each card in the Series 200 RG specifications source deck will contain the page number of the specifications sheet from which it was obtained and punched. The various specifications sheets each have in their headings a space for page entry which is identified as columns 76-77. (N.B.: Column numbers printed just above the body of each specifications sheet skip from column 75 to column 78.) Page number is required on each card, although the number is inserted only once on an individual specifications sheet from which a number of cards will be punched. The specifications sheets must be numbered consecutively and in the following order:

a. The Honeywell Report Layout chart always comes first in the page numbering sequence and is numbered 01 of ____ (______ will contain the total number of specifications pages prepared for the generation of this report).

b. Input Specifications, Data Specifications, Calculation Specifications, and Output Specifications sheets, in that order, follow the Report Layout chart. Page numbers are consecutive through the last Output Specifications sheet. If more than one sheet of any of the specifications is used, care should be taken to make certain that the sheets are correctly numbered in their intended order.

3. Card Numbering

Each line entry made on each specifications sheet will be punched into an 80-column card to make up the specifications source deck. For the purpose of providing faster and easier identification of line entries, as well as for establishing the order of entries, card numbers are preprinted in columns 78-80 on each of the specifications sheets. The three-character preprinted card numbers 010 to 200 permit a total of 20 line entries to be made, from which 20 cards will be punched. The five unnumbered lines at the bottom of each specifications sheet are for inserting additional line entries in the event that requisite statements may have been inadvertently left out. For example, if line entry 12 (preprinted number 120) should require further definition, a total of nine insertions could be referenced as card numbers 121 through 129. Five insertions (121 through 125) could be placed on this page; extension of the sheet by taping or glueing additional lines to the bottom of the page would be required to accommodate lines 126 through 129 and any statements referenced to other line entries.

Initializing Numerical Sequences

For any report which uses page, serial, or record count numbers, the object program will automatically number the pages, serials, or record counts consecutively. The object program initializes the first of these as 1 and maintains an increasing sequence as each succeeding page, serial, or record count number is brought into the report output. Since this processing is automatic, the programmer need not be concerned with maintaining the numerical sequences unless he wishes to interrupt the numbering sequence and resume or reinitialize it later (which is the same as starting the sequence with a number other than 1). In order to resume or reinitialize the numbering sequence, the programmer must prepare an initializing card which will be placed in front of the balance of the input records that are to be processed during the resumed or re-initialized machine run, and he must define the specifications contained in the card on both the Input and Data Specifications sheets.

An illustration of the specifications sheets entries is shown in Figure 2-14. The entries on the specifications sheets are explained as follows:

Line entry 1 of the Input Specifications sheet specifies a non-sequential record assigned two arbitrarily selected, alphabetic record-sequence characters. In accordance with the rules for preparing input specifications, a resulting condition number is specified for this line entry (also chosen arbitrarily), which indicates that the entry describes one input-record type.

Line entry 1 of the Data Specifications sheet specifies that page numbering will begin with that number specified in columns 5-7 of the CHH card. This is accomplished by re-setting to zero the data field, PAGENO., and adding the value from the CHH card source field. The reset-addition ($\overset{+}{0}$) operation is required only if the programmer is reinitializing the page number sequence at some point during a particular machine run. If the machine run is being resumed as a new run after an interruption (which may have lasted for a time of any duration), no operation character need be specified. The second source field specifies page number re-setting upon a change in control field 1. The source field specified, in this case ending in column 007, could be any convenient field in the initializing card. For this example, the card columns 5-7 have been designated. The page number value contained in these columns must be one less than the page number desired on the first page of output in the resumed or reinitialized run.

Line entry 2 of the Data Specifications sheet specifies the value for resuming or reinitializing the record-count number sequence. Serial numbering sequence can be specified in the same manner. The rules for resuming or reinitializing serial and record-count number sequences are the same as those for page numbers: The value specified must be one less than the serial or record-count number desired on the first page of output; any convenient field in the initializing card may be used to contain the value; and the reset-addition need be specified only for reinitializing numbering sequences within a particular run. This line entry is conditioned by F4, a change in control field 4. Record count numbers will increase when the control field representing the input records being counted changes.

INPUT SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE ⌊⌊ (76 77) OF ____

C (1)	RECORD SEQUENCE: SEQUENCE (2-3)	NUMBER (4-5)	RECORD CODES: POSITION (6-8)	NEGATE (9)	Z/D/C (10)	CODE (11)	POSITION (12-14)	NEGATE (15)	Z/D/C (16)	CODE (17)	POSITION (18-20)	NEGATE (21)	Z/D/C (22)	CODE (23)	POSITION (24-26)	NEGATE (27)	Z/D/C (28)	CODE (29)	POSITION (30-32)	NEGATE (33)	Z/D/C (34)	CODE (35)	POSITION (36-38)	NEGATE (39)	Z/D/C (40)	CODE (41)	RESULTING CONDITION (42-43)	CONTROL FIELDS: FIELD 1 END (44-46)	FIELD 1 LENGTH (47-48)	FIELD 2 END (49-51)	FIELD 2 LENGTH (52-53)	FIELD 3 END (54-56)	FIELD 3 LENGTH (57-58)	FIELD 4 END (59-61)	FIELD 4 LENGTH (62-63)	FIELD 5 END (64-66)	FIELD 5 LENGTH (67-68)	FIELD 6 END (69-71)	FIELD 6 LENGTH (72-73)	(74-75)	CARD NUMBER (78-80)
C	HH		Ø8Ø		C	1																					18														010
																																									020

DATA SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE ⌊⌊ (76 77) OF ____

D (1)	FIELD NAME (2-7)	FIELD LENGTH UNEDITED (8-10)	FIELD STATUS: STATUS (11)	RESULTING CONDITION (12-13)	STATUS (14)	RESULTING CONDITION (15-16)	STATUS (17)	RESULTING CONDITION (18-19)	DATA FIELD SOURCES: FIELD SOURCE (20-22)	NUMERIC (23)	FIELD END (24-26)	FIELD LENGTH (27-29)	OPERATION (30)	CONDITION (31-33)	AND CONDITION (34-36)	FIELD SOURCE (37-39)	NUMERIC (40)	FIELD END (41-43)	FIELD LENGTH (44-46)	OPERATION (47)	CONDITION (48-50)	AND CONDITION (51-53)	FIELD SOURCE (54-56)	NUMERIC (57)	FIELD END (58-60)	FIELD LENGTH (61-63)	OPERATION (64)	CONDITION (65-67)	AND CONDITION (68-70)	(71-75)	CARD NUMBER (78-80)
D	PAGENO	ØØ3							CHH		ØØ7		+0			PAG					F1										010
D	RECCNT	ØØ4							CHH		Ø12		+0			RCT					F4										020
																															030

Figure 2-14. Example of Entries for Initializing

One other stipulation for use of the initializing card is its identification. This can be provided, for example, by an I for "Initialize" (X-and 9 punch) in one column of the card. Identification can be provided, however, by any valid character in any column, provided that no other record types have been identified with the same code in the same column.

Efficiency of the Generated Program

When writing specifications for a report which will be generated by Series 200 RG, the programmer may select certain approaches to the problem which will result in a more efficient program than if he chooses other alternative approaches. Increased efficiency of the object program can be realized both in less program-execution time and decreased core-storage requirements for executing the program. Below are some suggestions.

Arrangement of the Specifications

When it is permissible to do so, the arrangement of the report specifications in a certain order will serve to increase program efficiency.

1. Input Specifications Arrangement

It is possible to arrange the input specifications in such a way that considerable savings in execution time and storage space result. Some of the means which may be employed are:

a. Those input-record types which occur most frequently in the input data file should be specified first on the Input Specifications sheet. For example, the line entry specifying the report date has only one source and, consequently, only one card is punched for this input record. This should be preceded by the line entry which specifies detail cards in an application where both types of input records are non-sequential.

b. Non-sequential input records should be arranged in such a way that those with the same control fields are specified by consecutive line entries. Arranging non-sequential records in this way will result in a considerable saving of space required for program storage.

c. The sequence-checking operation should be avoided, unless its contribution to the program is sufficient to outweigh the consideration that it uses a large amount of both execution time and core storage space.

2. Data Specifications Arrangement

No limitations are placed on the sequence of data-specifications line entries except in those entries specifying more than three field sources. However, the efficiency of the object program, both in time and space savings, can often be increased by rearranging the order of the line entries. The order which appears more logical is to enter the specifications in the same sequence on the Data Specifications sheets as they will appear in the printed copy output. It is better, though, for object program efficiency to arrange the data specifications such that those having the same field sources are placed together. Rearrangement of the data specifications to conform to this more desirable organization can be performed either before or after cards have been punched from the Data-Specifications sheet line entries.

If the line entries which define data fields with multiple sources are arranged in such a way that the first source specified on a line is the same as the last source specified on the preceding line, program efficiency is greatly increased. Figure 2-15 shows an example of data specifications which have multiple sources. The specifications are arranged in the order in which they will appear in the printed copy output.

Figure 2-16 shows rearrangement of the line entries to increase object-program efficiency. Line entries 1 through 4 have been arranged in a more efficient order, and transposition of the field sources of FIELDL performed. The result of this rearrangement will be a saving in both the amount of execution time and storage space required for the program.

3. Calculation Specifications Arrangement

Savings in both execution time and the amount of core storage space needed to store the object program can be effected by appropriate ordering of line entries on the Calculation Specifications sheet. Those calculations which are to be performed at detail time should be specified consecutively whenever it is possible to do so. Total-time calculations should also be specified consecutively whenever possible. Calculation

DATA SPECIFICATIONS

REPORT NAME ________ PROGRAMMER ________ DATE ________ PAGE ___ (76 77) OF ___

D	FIELD NAME	FIELD LENGTH UNEDITED	FIELD STATUS: STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	DATA FIELD SOURCES: FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION		CARD NUMBER
1	2 7	8 10	11	12 13	14	15 16	17	18 19	20 22	23	24 26	27 29	30	31 33	34 36	37 39	40	41 43	44 46	47	48 50	51 53	54 56	57	58 60	61 63	64	65 67	68 70	71 75	78 80
D	FIELDJ	ØØ3							CØ5		Ø42					CØ7		Ø39		S											010
D	FIELDK	ØØ5							CØ6		Ø17		$\overset{+}{o}$			CØ6		Ø31		A											020
D	FIELDL	ØØ4							CØ5		Ø53		S			CØ7		Ø67		A											030
D	FIELDM	ØØ3							CØ5		Ø72					CØ6		Ø26		S											040
D	FIELDX	ØØ6	Z	Ø7					CØ3		Ø24		A	F2		CØ4		Ø64		S			CØ3		Ø39		S	F2			050

Figure 2-15. Multiple Field Sources

DATA SPECIFICATIONS

REPORT NAME ________ PROGRAMMER ________ DATE ________ PAGE ___ (76 77) OF ___

D	FIELD NAME	FIELD LENGTH UNEDITED	FIELD STATUS: STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	DATA FIELD SOURCES: FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	CONDITION	AND CONDITION		CARD NUMBER
1	2 7	8 10	11	12 13	14	15 16	17	18 19	20 22	23	24 26	27 29	30	31 33	34 36	37 39	40	41 43	44 46	47	48 50	51 53	54 56	57	58 60	61 63	64	65 67	68 70	71 75	78 80
D	FIELDJ	ØØ3							CØ5		Ø42					CØ7		Ø39		S											010
D	FIELDL	ØØ4							CØ7		Ø67		A			CØ5		Ø53		S											020
D	FIELDM	ØØ3							CØ5		Ø72					CØ6		Ø26		S											030
D	FIELDK	ØØ5							CØ6		Ø17		$\overset{+}{o}$			CØ6		Ø31		A											040
D	FIELDX	ØØ6	Z	Ø7					CØ4		Ø64		S			CØ3		Ø24		S	F2		CØ3		Ø39		S	F2			050

Figure 2-16. Rearranged Data Specifications

specifications which are governed by the same conditions should be kept together, as was recommended in the case of data specifications for data fields having the same sources with the same conditions. It will not be possible to adhere to these ideal arrangements in every report application, but the more often it is possible to do so, the greater the object-program efficiency achieved.

4. Output Specifications Arrangement

The programmer can observe two conventions in writing output specifications which will materially reduce both the amount of execution time and the amount of storage space required for the object program. First, whenever the entry designating a "next line" is applicable, it should be used. Secondly, data fields should not be blanked after they are used, unless blanking of the field is absolutely necessary (because the operation wastes storage space). The blanking of a data field after it is placed in an output line is a process similar to that of an accounting machine when it performs a readout and reset total operation. Whenever possible, the programmer should write instructions that do not blank the field after it is placed in an output line (F instead of B in column 1 of the Output Specifications sheet).

Split Control Fields

An input record file is usually arranged so that the control fields are located in several adjacent character positions within each record. This may not always be the case, since it is frequently not possible to arrange the records in this way. Also, in some instances, files may have been originally set up in a particular manner to serve a certain purpose; rearranging all the input records involved into a different order would be too costly a process to be considered. Another instance might be encountered in the case of a particular report wherein it is desired to have subsequent operations governed by the information punched in non-contiguous card columns. Merely renumbering the control fields would not clarify the situation, because the characters making up the parts of such a control field might be located in widely separated columns. An example is shown in Figure 2-17 which includes a "split" control field (i.e., characters not located in adjacent positions).

The control fields which are specified in the line entry of Figure 2-17 show a minor control field located in columns 4-9 of the punched cards in the input record file, an intermediate field in columns 45-47 and columns 21-23, and a major control field in columns 30-34.

If subsequent processing is to be governed by a change in one of these control fields, special consideration must be given to handling of the split control field. When processing is to be governed by a change in the minor field, the condition entry F1 can be used. But if a processing is to be governed by a change in the intermediate field, a split control field, the condition entry should be F2. Condition F4 should be specified when processing is to be governed by a change in the major field.

INPUT SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE _ _ OF ____ (76 77)

RECORD SEQUENCE				RECORD CODES																									CONTROL FIELDS													
																													FIELD 1		FIELD 2		FIELD 3		FIELD 4		FIELD 5		FIELD 6			
C	SEQUENCE	NUMBER		POSITION	NEGATE	Z/D/C	CODE	POSITION	NEGATE	Z/D/C	CODE	POSITION	NEGATE	Z/D/C	CODE	POSITION	NEGATE	Z/D/C	CODE	POSITION	NEGATE	Z/D/C	CODE	POSITION	NEGATE	Z/D/C	CODE	RESULTING CONDITION	END	LENGTH	END	LENGTH	END	LENGTH	END	LENGTH	END	LENGTH	END	LENGTH		CARD NUMBER
1	2 3	4	5	6 8	9	10	11	12 14	15	16	17	18 20	21	22	23	24 26	27	28	29	30 32	33	34	35	36 38	39	40	41	42 43	44 46	47 48	49 51	52 53	54 56	57 58	59 61	62 63	64 66	67 68	69 71	72 73	74 75	78 80
C	Ø3			Ø8Ø		C																							ØØ9	Ø6	Ø47	Ø3	Ø23	Ø3	Ø34	Ø5						010
																																										020

Figure 2-17. Example of Split Control Field

Note that a change in control field 3 is not used as a condition entry. Explanation of this omission is derived by considering what takes place when a change in the data contained in the split control field occurs. If a change occurs in columns 45-47 of the split control field, conditions F1 and F2 are fulfilled. If a change occurs in columns 21-23, conditions F1, F2, and F3 are fulfilled. It is apparent, then, that any processing or operation conditioned by a change in control field 2 will be performed when a change occurs in either part of the split control field. On the other hand, if the operation is conditioned by F3, the condition will be fulfilled only upon a change in that part of the split control field specified in card columns 21-23; or alternatively when a change occurs in the major control field which incidentally fulfills conditions F1, F2, and F3, as well as condition F4. Therefore, if F3 instead of F2 is used to condition an operation which shall be performed upon a change in the intermediate control field, a change in columns 45-47 would not be recognized as fulfillment of the condition.

There are, then, two rules for specifying split control fields and their condition entries:

1. As shown above, the lowest numbered field of those which comprise the split control field must be designated as the condition entry when a change in split split field is being specified.
2. The parts of the split control field may be specified in any order, provided that rule 1, above, is observed. In Figure 2-17, card columns 45-47 may be specified as either field 2 or field 3, but condition F2 must be used in order to govern performance of an operation when a change in the split intermediate control field occurs.

Suppression of Processing

Control fields have been defined previously as those character positions in input data records which contain information that is used to govern subsequent operations. (The input data records in this instance will be punched cards in a file of records.) Whenever the information contained in the input records indicate to the Report Generator that changes in control fields have occurred, the heading lines, total lines, and total calculations which are conditioned by those control field changes will be processed. Conditioning these operations by control-field changes normally will result in the proper processing taking place, but instances can occur in which control-field change conditioning is not sufficient to govern calculations or output.

Such an instance would be encountered in the case of an input record at the beginning of a run having its control fields compared with blanks. This would result in a change in control; if the input record produced a total line, this line would be processed and become printed output. At the beginning of a run, heading lines are usually processed, but generally it is not desirable to have total lines become output. Additional conditioning is required to suppress the processing of total calculations and total lines due to control-field change when the cards are being read.

The additional or double conditioning device which is usually used to govern suppression of these total operations is to specify that the field is not blank, as well as to specify conditioning by a control-field change. A field which has blank status will contain usable information only after the first input record has been processed, suppression of total operations can be performed when the cards are being read if the field is blank.

SECTION III

FUNCTIONS AND LOGIC OF THE SERIES 200 REPORT GENERATOR SYSTEM

In the introduction to this publication, the Series 200 Report Generator system was described as a series of five logical segments linked together to form the Report Generator Master Deck. Section II of this manual describes the specifications of the Series 200 Report Generator in detail. Section VI details the process of converting the specifications into printed output (alternatively, punched card or tape output can be obtained in the Series 200 Report Generator system). Section III details the functions of the Series 200 Report Generator and the logic path followed to perform the functions.

OPERATING FUNCTIONS OF THE SERIES 200 REPORT GENERATOR

The Control Section

The first segment of the Report Generator system is the Control section. Input to the Control section is the one card formulated by the user which contains specific information about the available hardware which can be used by the generated program. This card is identified by the characters CNTL contained in columns 1-4. Information about optional hardware is also specified on the control card. The Control section turns on switches in the common storage area of upper memory in bank 0 so that the other sections of the Series 200 Report Generator system may check to see if the following options are available for their use:

1. The select-stacker option switch is interrogated by all the sections of the system and is used to separate the symbol definition cards from the executable instructions.
2. The multiply-divide hardware switch can be turned on and interrogated by the Calculation section, but the hardware is not presently available to the Series 200 RG system. The system will be modified at a later date in order to use the multiply-divide hardware option.
3. A carriage control tape hardware switch can also be turned on by the Control section; however, the basic version of the Series 200 RG has no routine to utilize the hardware. At present, a subroutine simulates the carriage control tape. The user is thereby enabled to designate 12 channels that will be available with the hardware.

In order to utilize core storage as economically as possible, the generated program has its origin at location 134 (decimal). Information concerning the size of the printer is stored in upper memory of bank 0 for use by the Output section. The control card also contains the input/output trunk number designations, information which is used by the Control section. The trunk number information is also stored in upper memory of bank 0 so that the Output section may check to determine whether instructions have been set up to permit equipment usage. The Control section also stores in upper memory of bank 0 a six-character identification label (from columns 75-80 of the Control Card) which the user may specify to mark every card in the generated deck. All sections of the system will then mark their cards with this identification in columns 75 through 80.

After the Control section has interpreted the CNTL card, it then reads in one by one a deck of cards which contain peripheral instructions and subroutines used in common by all report generating programs. All cards in this deck contain a code in column 60. The Control section interprets the code and accordingly causes the cards to be punched as they are, or modified and punched, or skipped, depending upon the peripheral equipment configuration specified on the Control Card. The instructions contained by the cards are written in Series 200 Easycoder programming language and include:

1. A PROG card and an ORG card.
2. CEQU cards which define the trunk numbers for the peripheral devices used.
3. Instructions which interrogate the console switches and consequently set internal switches which are interrogated by other sections of the generated program.
4. Instructions to initialize the peripheral equipment and set word marks for the input/output areas.
5. Instructions for reading in the input data file(s).
6. Instructions for writing and/or punching the output file.

The Input Section

The specification cards read into memory by the Input section of the Series 200 Report Generator system are separated into four basic types of information about the input file:

1. The input record's name designates whether or not the input record can be part of a sequential order. The record names are saved in a table in upper memory of bank 0 to be used later by the Data section.
2. The record is identified by one or more unique characters (record codes) contained in designated positions. The Input section generates instructions that will compare the contents of the position designated with the possible identification character names. Instead of full-character identification, the zone bits or the numeric bits may be used. The Input section will continue setting up record-code interrogation instructions until a resulting condition is specified on the card.
3. When a resulting condition is encountered, it is picked up and becomes a part of a symbol which indicates to other sections of the generated program that this is the record that has been read in for processing.
4. The Input section provides instructions which will set switches when certain information, identified by location and length, changes for that record type. For example, when the date contained in a certain field of the record changes, the user wishes to start another page in the report.

After the four types of information listed above have been handled, the Input section sets up an instruction, or connector label, which will insert the appropriate entrance to the field processing instruction. The connector label consists of two parts: The record name, and the initialized sequence, 00.

The Input section is capable of handling optional records, i.e., those records that do not necessarily have to be present for a report line to be processed. In order for an optional record to be handled properly, the Data section must also generate a routine for handling the record.

The Data Section

The Data section of the Series 200 Report Generator interprets the information contained on its specifications cards in one of three ways:

1. The name of the field;
2. The field source (where in the input record the contents of this field may be found); and
3. Routines to be generated according to the field status after it has been operated upon.

A symbol definition statement is set up for the field name. The space reserved is the length of the field.

The possible data field sources are identified by the record name in which the field is found, the length of the field in the record, and the location within the record of the rightmost position of the field. The source may be moved into, added to, or subtracted from the field. The field may be zeroed and the information added to or subtracted from it. Either the numeric portion or the zone portion of a single character may be transferred to the units position of the field. Finally, it is possible to define two conditions, in an "and" relationship, that can govern whether or not any of the above operations will be performed.

After the field has been set up, the Data section will generate instructions which will interrogate the state of the field (if a field status was specified) and set switches to appropriately reflect the field state. The types of conditions (i.e., field states) can then be used to govern subsequent processing.

The labeling used to process all of the information in a specific record is the sequenced record name, appropriately incremented. After all of the specification cards for this section have been processed, the Data section then defines the dangling sequence labels. All of these paths converge at the entrance of the instructions set up by the Calculation section to perform detailed calculations.

The Calculation Section

The Calculation section of the Series 200 Report Generator system generates the instructions which perform final arithmetic operations upon the fields that are to appear in a printed line. There are two main labeling paths which may be developed, depending on whether the information is for a detail line or a total line. In addition to the type of operations which are permitted in the Data section (except the "move" operation), the operations of multiplication, division, and comparison of one factor to another are permitted in the Calculation section. These factors may be defined in either the Data or Calculation sections. The specialized operations which can be performed by the Calculation section are:

1. Intermediate fields may be defined by the result of one symbolic or literal factor operating upon another.
2. Three conditions, in an "and" relationship, may be defined to govern whether or not the operation is to be performed.
3. A field may be half-adjusted by a specification which designates the position of the digit which shall have five added to it to round off the result.
4. The resulting field of an operation may be shifted in order to drop insignificant digits.

After the areas of the Calculation Specifications card listed above have been interrogated, the Calculation section, like the Data section, can check the resulting condition of the field and set appropriate switches on or off. The Calculation section will also insert the multiplication and division subroutines in the generated program if they are needed.

When all of the Calculation Specifications cards have been read and processed, the exits for the two paths are set up. These exits lead to appropriate entrances to the instructions set up by the Output section. Which exit is taken depends on whether the processing dealt with a total line or a detail line.

The Output Section

The final section of the Series 200 Report Generator system, the Output section, is itself two linked parts. The first part analyzes the specifications cards for the different kinds of output lines and generates routines to produce the output. The second part of the Output section is read in after all of the Output Specifications cards have been read. The latter part of Output completes path linkage between the various types of routines generated, supplies additional routines if necessary, and reads and processes the CARRIAGET card.

The organization of the first part of the Output section is based on the unit of cards which work together to describe a single output line. The unit is composed of a line specification card (more than one if "or" conditions are present) and the various field specification cards associated with the line. Processing of each unit of specification cards varies according to the type of line specified, whether Heading, Detail, or Total. The processing is similar in each case; the differences exist in labels produced (e.g., HN0000, DD0000, and T11T00) and in sequenced hierarchy logic.

Each line specification card, when it is read, is moved to a save area where the information remains available for use in setting up entry and exit points for the routines generated to handle the line. All the field cards for the output line are read into a read area where they are analyzed. Instructions are then generated (1) to test for the specified conditions; (2) for checking conditions associated with each field; and (3) for moving and editing fields. When the last field card has been processed, instructions for controlling output and page alignment are generated. The program supplies overflow routines for the line, if necessary, and puts out links to other line routines based on "next line" and sequenced hierarchy logic. The program processes each unit of cards in the same manner until all specifications have been processed. Sensing of a comment card results in the reading of the second part of the Output section.

The second part of the Output section generates branches between the various types of line routines and links between the routines produced by this section and the routines produced by previous sections. If necessary, an overflow subroutine is generated; and a carriage tape control simulator routine is read and punched out by the Control section. The CARRIAGET card is read by this part of the Output section; constant field definitions for the number of lines per page, the overflow equivalent line number, and the current line count are generated; and a table of specified channel and line equivalents is punched.

PROCESSING BY THE CONTROL SECTION

The Control section processing logic is organized around the following items:

1. Information saved from the Control Card;
2. Program and Origin Cards;
3. Processing of symbolic routines; and
4. End of symbolic routines.

Information Saved From the Control Card

When the Control Card is read (reading is performed in a subroutine), the information to be used by other sections is stored in the communication area. The information saved includes the following:

1. The name of the generated program;
2. The memory size of the object computer;
3. Whether multiply-divide hardware is available on the object computer;

4. Whether the output of the generated program is to be punched cards;
5. The printer size and number of print positions to be used by the object program;
6. The use of tape for input and/or output;
7. Data tape description if any input tape use is specified;
8. Whether the stacker select option is available to the Report Generator; and
9. Description of the format of the output data tape if there is one.

The specified trunk numbers of the reader, punch, printer and magnetic tape units of the object computer are saved for use by Control. If no trunks are specified, Report Generator will use the Series 200 recommended trunk numbers.

Program and Origin Cards

The program (PROG) card is punched using the program in the operand field. An origin (ORG) card is punched, setting the origin for the assembled instructions of the generated program at location 134.

Processing Symbolic Routines

After Control has interpreted the control specifications card, it reads the comment card. Then it reads (reading in both cases is performed in a subroutine) and processes, one by one, records containing routines which will be used by the generated program. The routines are written in the Honeywell 200 Easycoder programming language.

Codes in columns 60 through 62 of each record and the saved information from the control specifications card govern the output of the instruction cards. Records may be punched, modified and punched, or skipped, depending on the coded information.

The contents of column 60 may be: blank, C, D, A, M, N, P, or J.

If column 60 contains an "M," column 61 may be: blank, C, D, N, M, B, A, or T.

If columns 60 and 61 contain "MN", column 62 may be: blank, or N.

The permissible code combinations and their meanings are shown in Table 3-1.

Processing starts by testing column 60. If a blank or "C" is present in column 60, the card is punched.

Table 3-1. Codes for Handling Dummy Instructions

Contents of Card Column			Meaning of Code
60	61	62	
blank	blank	blank	No significance; always punched.
A	blank	blank	Card Reader input.
C	blank	blank	No significance; always punched.
D	blank	blank	CEQU cards for 10 devices control number inserted.
J	blank	blank	Punch output.
M	blank	blank	Magnetic tape input.
M	B	blank	Tape read PCB card: drive number inserted.
M	C	blank	Tape input, unblocked variable-length record, with banner character.
M	D	blank	Tape input, fixed-length record, with banner character.
M	M	blank	FILCNT card; number of tape input files inserted.
M	N	blank	Tape input, blocked records.
M	N	N	Tape input, blocked, RECSIZ card: record size inserted.
M	T	blank	Tape read PDT card: parity and drive number inserted.
N	A, or blank	blank	Tape output with or without spacing control characters and lines.
P	blank	blank	Printer output.

If a "D" is present in column 60, the reader trunk number is inserted and the card is punched.

If an "A" is present in column 60, a test is made to see if the input was specified as cards. Records with an "A" in column 60 are punched for card input. If input is tape, then transfer is made directly to read another card and the "A" card is skipped.

If an "N" is present in column 60, a test is made to see if output was specified as tape. "N" records are punched only if there is tape output. Otherwise, "N" records are skipped.

If a "P" is present in column 60, a test is made to see if printer size was specified. "P" records are punched when there will be printer output. Otherwise, "P" records are skipped.

If a "J" is present in column 60, a test is made to see if punched output was specified. "J" records are punched only if output is cards. Otherwise, "J" records are skipped.

If an "M" is present in column 60, a test is made to see if the input was specified as tape. Records with an "M" in column 60 are not punched for card input. If tape input is specified, the "M" record is moved to the output area.

Tests are next made on columns 61 and 62, with column 61 being tested first. If column 61 is blank, the card is punched.

If column 61 is a "C", a test is made to see if variable record length was specified. "MC" records are not punched for fixed length input records. If the records are specified as variable length, a test is made to see if banner characters were specified. If so, the record is punched; if not, the record is skipped.

If column 61 is a "D", a test is made to see if variable length records were specified. "MD" records are not punched for variable length records. If the records are specified as fixed length, Control tests to see if banner characters were specified. If so, the record is punched; if not, the record is skipped.

If column 61 is an "M", the file count specified is inserted and the card is punched.

If column 61 is a "B", the drive number is inserted and the card is punched.

If column 61 is a "T", the drive number and the parity number are inserted and the card is punched.

If column 61 is an "N", a test is made to see if blocking was specified in the control card. If blocking was not specified, "MN" records are not punched. If blocking was specified, Control tests column 62. If column 62 is blank, the record is punched. If column 62 is an "N", the record size is inserted and then the card is punched.

If column 61 is an "A" and column 20 of the Control Card was blank, the card is skipped.

End of Symbolic Routines

When a record is read with an "E" in column 60, all records have been read and processed. Control ends by reading the Input bootstrap into memory and branching to it, thus loading the Input section.

PROCESSING BY INPUT SECTION

The Input section processing logic is organized around the sequence of the specifications:

alphabetic (non-sequenced) record specifications, and numeric (sequenced) record specifications. Processing of non-sequenced and sequenced record specifications is similar. Preparation of the linkage for sequenced specifications, however, is more involved, as the linkage must include sequence checking logic.

The Input section does preliminary processing of each specification card, producing instructions for checking record codes and control fields if needed. In order to supply "or" logic and sequence logic, the Input section reads the next card. When processing of the first card is completed, processing of the card just read begins.

When a card is read which specifies end of a fixed sequence (SCF card), the processing of the previous card continues. Then Input processes the SCF card. After processing the SCF card, Input branches to read another card.

Processing of the Input Specification cards is as follows:

1. Determine the Type of Specification. The first step in the processing of each card is a check of the specification type. If the card is a C-type, transfer is made by Input to processing of input record specifications - first, to prepare Tag and Exit for Routines Based on the Sequence of the Record. If the card is an S-type, Input goes to Process Fixed-Sequence Specifications. If the card contains an asterisk, Input transfers to End of Cards routines.

2. Process Fixed-Sequence Specification. When an S-type card is found, Input generates the instructions at which all routines for sequenced records terminate. The generated instructions use the control field specified by the fixed-sequence card for a test which will check for an error in the sequence field and will transfer if there is no error to ENTRY2. The first instruction has the tag SEQR1. An error halt, ERS2, is generated. This halt is reached when the generated program finds that a necessary sequenced record is missing or that there is an incorrect change in the specified control field. After generating these instructions, Input transfers to read another card and Determine the Type of Specification.

3. Tag and Exit for Routines Based on the Sequence of the Record. A C-type specification card describes an input record. Input tests to see if an alphabetic or numeric sequence is described. The previously used level of the appropriate tag for the sequence specified is incremented and moved to a save area. This tag will be used on the first generated instruction of the routine for checking the record. If the sequence is alphabetic, the tag is DA(XX)00. If the sequence is numeric, the tag is DN(XX)00. (This is true except for the second of two "or" records. Then the tag is RN(XXXX+2) where RN(XXXX) is the RN tag for the first of the "or" records. See OUT(XX) Instructions for Numerically Sequenced Records and RN(XXXX) Instructions for Numerically Sequenced Records.)

4. SAVE Columns 1-5. The first five columns of the specification card are saved. These will be used after the next record specification card is read for completing the processing of this card.

5. Instructions for Checking Record Codes. Six fields per card can be used for specifying record codes. For each specified code the appropriate compare and branch instructions are generated. If the sequence is numeric, the generated branch instruction will go to the current level of the OUT(XX) tags when the code is not as specified. When the sequence is alphabetic, the branch is to the next higher level of the DA(XX) 00 tag. When the six record code parameters have been checked, Input tests for a resulting condition. If none is specified, Input reads another card and checks for six more record codes (i.e., it is assumed that the next card is a continuation of the present one.)

6. Resulting Condition. When a resulting condition is sensed, Input generates instructions to turn on the condition switch. The resulting condition code is stored in a table. Input will use the table for generating data formatting statements for condition switches after all cards have been processed.

7. Instruction for Modifying SW4. An instruction is generated to modify SW4. The instruction will place the address of the routines for this source record in SW4. (SW4 directs processing to the appropriate data manipulating routines produced by the Data section.)

8. Routine for Checking Control Fields. Up to six control fields may be specified. Where possible, Input generates one routine to check control fields for changes. Input generates links to this routine from other record checking routines.

   Input generates a tag: either TN(XXX) for numeric sequenced specifications, or TA(XX)00 for alpha specifications. Input then checks to see if the previously used T tag was for a numeric or an alphabetic sequence. If sequencing has changed from alpha to numeric, a new set of instructions is required for control field checking. Also, if there is a change in the control fields specified, a new set of instructions is required. If neither of these changes has occurred, Input generates a branch to the routine previously generated for control field checking. Input processes control field specifications from right to left, that is, from field six to field one. If a field is specified, the appropriate compare and branch instructions are generated to test for a field change.

   The branch on a field change is to STF(XXX). Each time a new routine is generated for checking control fields, the last digit of the STF(XXX) tag is incremented. The second X digit is 1 to 6, referring to one of the six control fields. If any control fields are specified, the digits are moved to a save area. When all control field specifications have been processed, Input puts out the exit instruction. (See Tag and Exit Routines Based on the Sequence of the Record.)

   If there were no control fields specified, the exit instruction has the T_A^N(XXXX) tag. If control fields were specified, the first compare instruction has the T_A^N(XXXX) tag. If control fields were specified, Input generates the instructions to set a switch to signal a change in control field and to save the new value of the field. There are two instructions for each control field. The first instruction has the STF(XXX) tag.

   Input keeps a table of the control field specifications for data formatting statements. These will be generated when all input specifications have been processed. After all STF(XXX) instructions have been generated, the exit instruction is put out.

9. Read the Next Card. After Input has put out the required instructions to handle changes in the control field, another card is read. If the specifications

of the new card do not imply an "or" specification, Input moves the record name and optional information saved previously (see SAVE Columns 1-5) into the table of source names. If the previous card specified an alphabetic sequence, no more processing is required. Input branches back to Determine the Type of Specification for the new card.

10. OUT(XX) Instructions for Numerically Sequenced Records. When the generated program finds that the current input record does not have the record codes specified for the expected numerically sequenced record, it transfers to OUT(XX). Input increments the OUT(XX) tag by +1 for each numeric specification card. There are several possibilities as to why the expected record codes might not be found: (a) there is an error in the input data; (b) there is an "or" possibility; (c) the record may appear more than once (checking routines are repeated for a multi-record type until another record type is read); and (d) the expected input record is optional. The processing followed by Input for these possibilities is expanded below:

a. Errors. If the saved specification shows that the record can appear only once in a sequence of records and is not optional, Input generates a branch to ERS2 (an error halt). The OUT(XX) tag is used for the instruction. Input then transfers to prepare the RN(XXXX) routine.

b. "OR" Specification. If there is an "or" relation specified, Input generates a branch to a routine for the other possible combination of record codes. (See RN(XXXX) Instructions for Numerically Sequenced Records and Tag and Exit for Routines Based on the Sequence of the Record.) Input then transfers to generate RN(XXXX) instructions.

c. Multi-Record Specifications. If the record specified may appear more than once, Input punches instructions to check that at least one of the records was found. The generated branch for no record goes to PL(CXX). CXX is the specification type with sequence digits from columns 2-3 of the specification card. If the record is not specified as optional, Input generates a branch to ERS1. The branch is given the PL(CXX) tag.

d. Optional Records. If the multi-record is specified as optional, Input generates a routine using the PL(CXX) tag on the first instruction. This instruction modifies SW1 to transfer to the routine for the next specified record. Input generates a routine to be used when the optional record is not found. This routine is the same for both single and multi-record specifications. The generated instructions turn on the optional record switch; move the record tested (not the optional record) to a save area; blank the read area; and branch directly to the routine generated by Data for processing the optional record.

   The generated program logic requires that the condition and status testing path be executed. The optional record switch will signal to the routine generated by Data that there is not actually a record. After the routine generated by Data is executed for the optional source record, control goes through SW1 again to the routine to process the next specified record. The set of instructions described above are generated following the PL(CXX) instruction for multi-record specification. Single record optional specifications cause these instructions to be generated with the OUT(XX) tag on the first instruction.

11. RN(XXXX) Instructions for Numerically Sequenced Records. After Input has analyzed "OR", OPTIONAL, and MULTI-RECORD specifications and generated the appropriate instructions, it must generate the RN(XXXX) instructions. The RN(XXXX) tag is generally the same as the DN(XXXX) tag with the substitution of the "R" for the "D". The first generated instruction is for modifying the SEQR1 routine. If the specified record is the first of a numeric sequence, the generated instruction moves a zero to the branch instruction following SEQR1 so that it becomes an unconditional branch. This allows a change in the control field. If the record is not the first of a sequence, the generated instruction modifies the branch to not allow a change in the control field.

    The second instruction generated modifies SW1 to transfer to the routine for the next record of the sequence. Input generates a data-formatting statement for the address of the next RN(XXXX) tag. (This instruction is not generated if the specification card is the second for an "or" record.) A branch to the DN(XXXX) tagged instruction is generated. If the record is to appear only once, or if the card is the second for an "or" record, Input transfers to Determine the Type of Specification of the card in the input area. Input generates an RN(XXXX + 1) routine for multi-records. The generated instructions set a switch to signal that the record was found at least once. Input generates another branch to DN(XXXX). Input then transfers to Determine the Type of Specification of the card in the input area.

12. End of Cards. When Input senses an asterisk in column 1, all cards have been read and processed. Input punches path connectors for the numeric sequence path. These are SW1 and the highest +1 of the RN(XXXX) tags. Each is a branch to RN0100. Input stores the last address used for the source record table. The address will be used by the Data section for searching the table.

    If necessary, Input punches data formatting statements for digit, zone, and optional specification processing. Using information stored during processing, Input punches data formatting statements for resulting condition switches; control field change switches; control field save areas; and the addresses of the routines to be generated by Data for each source record. The Input section ends by reading the Data bootstrap into core and branching to it, thus loading the Data section.

PROCESSING BY DATA SECTION

The Data section processing logic is organized around the source and condition parameters. Except for storage allocation for the field, which is similar for all cases, the processing of the information in the field specification cards follows two logic paths based upon the two types of specified sources. One path handles sources of page, serial, or record count. The other handles input record sources. Both paths end by transferring to examine the next source parameter in the card. See the section on The Source Loop, below.

Processing of the Data specification cards is as follows:

1. Storage Allocation for Fields. Every time a card is read which specifies a new field name, a data-formatting statement for the field is generated. The specified field name is used as the tag. The field length is used to describe the field, which is initially blank. The tag references the rightmost position of the field. There is a wordmark in the leftmost position. The program checks for and handles appropriately field lengths greater than 40, using DCW, DC, etc.

2. Test for Change of Source or Conditions. The specified source being processed is compared to a save field. If there is a change in source, the program goes to generate End-of-Block Links, to prepare the Tag on First Instruction in a Block, and to process Conditions. If there is no change in source, the specified conditions are compared to a save area to see if there is a change in condition specification. This change can be to new conditions specified or to no conditions specified. If there is a change, the program goes to prepare the Tag on First Instruction in a Block and to process Conditions. If there is no change in source or conditions, the program transfers to produce instructions for Operation on Data.

3. End-of-Block Links. Every time there is a change in source, an exit for linkage from the previous block of instructions is generated. (Unless the last source was PAG and there were no instructions generated.)

4. Tag on First Instruction in a Block. When there is a change in source, or a change in conditions, a new tag is used for the first instruction generated.

   a. Page, Serial, or Record Count Source. If the source specified is PAG, a test is made to see if there are conditions specified. The absence of conditions implies that no page number reset is desired and thus no instructions need be generated. In this case, no new tag is set up and the program transfers to analyze any additional source specification in the card. If there are conditions on the PAG source, or if the source specified is serial or record count, the current level tag of the SER(XXX) sequence is moved to the punch area to be the tag of the first instruction. The tag is incremented in a save area. It will be used for exits and in the next SER(XXX) routine.

   b. Input Source. If the source is an input record, the program searches a table for the tag to be used. The tags are those specified in the Input Section with the incremented portion showing the current level of the tags in the path being generated for the source record. If the source specified is not found in the table of input record sources, an error halt occurs. The tag found is moved to be the tag of the first instruction. The variable portion of the tag is incremented for exits and for use in the next routine of this source.

5. Conditions. A change in conditions specified can be to new conditions specified or to no conditions specified. When there are conditions specified, the necessary compare and branch instructions are generated. (There may be one or two conditions. If there are two, they are in an "and" relationship.) If instructions are generated, they are the first instructions of the block. The exits generated for cases when the conditions are not satisfied are to the next level of the sequenced tags.

6. Operation on Data

   a. Page, Serial, or Record Count Source. If the routine being generated is for specifications with a page source and there were conditions found, an instruction is generated to blank, that is, to reset the specified field. If the source specified is serial or record count, an instruction is generated to add +1 to the specified field. When the source is any one of these three, the status and resulting conditions specified are not analyzed. Transfer is made to analyze any additional source specifications in the card (see Source Loop).

b. Input Source. When the source is input, a number of possible options may be specified by the user. If month conversion is specified, the instructions to perform this operation are generated. The codes specified for October, November, and December are used in data-formatting instructions. The tags for these are sequenced M(XX), starting from M(00). Instructions are generated to test for these codes and according to whether or not any is present, to move the appropriate month number into the specified field. After Data has generated these instructions, it transfers to test if Status and Resulting Conditions must be checked. If month conversion is not specified, Data continues.

   If the specified length of the field in the input record is not the same as the field length reserved by storage definition, an instruction is generated to set a word-mark at the appropriate point in the input record. Therefore, processing of the field is regulated by the word mark in the input area rather than by the word mark in the storage area.

   The specification of numeric conversion also causes a set wordmark instruction to be generated. In addition, instructions are generated to remove zone bits by saving the contents of the source field, zeroing the source field, and performing a decimal add of the information back into the source field. A temporary field is used by the generated program for saving the source field. The field length of the specified field is compared to a save area. If it is greater than any previously processed, the length is saved.

   If a "Blank" status is specified, an identical routine is generated which is referenced by an incremented tag BLNK(XX). The BLNK(XX) routine is placed among the data-formatting instructions and is part of a generated path which is separate from the generated paths for particular sources. This path is entered after a line is printed to check if any field status has been changed by a blank-after-move specified to the Output section (see Output - Part 2). After the field status specifications have been analyzed, the program transfers to analyze any additional sources in the card or to read another card.

7. The Source Loop. The processing of all source specifications terminates by transferring to one point. At this point the source parameters in the card not yet analyzed are moved over so that the next, if any, is available for processing. A test is made to see if three source parameters have been processed. If not, a test is made to see if the available source specification is blank. If it is blank, transfer is made to make the next source specification in the card available. If the source specification is present, transfer is made to Test For Change in Source or Conditions. When the three possible source specification parameters in the card have been interrogated, the loop ends. The program then goes to read another card.

8. End of Cards. When a card is read with an asterisk in column 1, all data specification cards have been read and processed. Final links are generated for the last level of the sequenced source tags. All paths for input record sources terminate in a branch to ENTRY 3, unless the record is optional. In that case, the branch is to a routine for optional records; also, this routine is generated. The instructions generated for the optional record routine test the optional-record switch which may have been set earlier to see whether the record was not found. If the switch is off, the exit to ENTRY3 for

ordinary processing of the record is taken. If the switch is on, the current record is moved back to the read area and the exit is to SW1 and to the tests generated by the Input section for the next record after the optional record. (See the Input section for more explanation of optional records.)

The SER(XXX) path is linked to SW4. A data formatting statement is generated for the temporary work area to be used by numeric conversion. The longest length saved is used to describe the field. TEMPIN is the tag used. ENTRY 2, the converging point of paths generated in Input, is linked to TCAL00, total line processing. No linkage for the BLNK(XX) path is generated at the end of Data. The path will be continued by Calculation, using the current sequence level which is saved in an area of common storage, the area for communication between sections. The Data section ends by reading the Calculation bootstrap into core and branching to it, thus loading the Calculation section.

PROCESSING BY CALCULATION SECTION

The Calculation section processing logic is a step-by-step procedure for processing the information in each specification. Processing of the calculation specification cards is as follows:

1. Data Formatting for Result Field. Each time a card is read, a test is made to see if the operation specified is a compare. Compare specifications do not use result fields. When a compare is specified, the program transfers to Testing for Literal Factors. If the operation is not a compare, a test of the result field specification is made to see if it is blank. A blank implies that this card contains specifications for the result field of the previous card. In this case the field and its length are moved from a save area to the card input area, and Calculation transfers directly to Testing for Literal Factors, bypassing the generation of another data formatting statement for this field. Otherwise, for each new result field, a data formatting statement is generated. The tag is the specified name and references the rightmost character of the field. The specified field length is used to describe the field. Field lengths longer than 40 positions are appropriately handled with DC and DCW instructions.

2. Testing for Literal Factors. A test is made to see if the operation column is blank. A blank implies that there is not a first factor and the program transfers to test only the second factor for the presence of a literal. If there was a compare operation sensed, or if the operation column has been found to be not blank, the first factor is tested for zone bits. If there are no zone bits on the first character of the factor and the factor is not blank, the factor must be numeric. Calculation generates a data formatting statement for the constant, adding a + sign to the units position if no zone was present on the units position. The tag used is of the form KCAL(XX). This tag is moved to replace the literal in the input area. Further processing will utilize this tag for referencing the literal. After using the current level of this tag, Calculation increments the (XX) portion for use as the tag for the next literal found. The second factor is tested and if a literal is found, processing is the same as for the first factor.

3. Test for Same Calculation Type and Same Conditions. The previously processed calculation type is compared to the type in the current specification to find out whether there is a change in when the calculation is to be performed. (Total or Detail line preparation.) If there is a change,

Calculation transfers to put out the Exit Link at Change in Calculation Type, to prepare the Tag on First Instruction in a Block, and to process Conditions. If there is no change in the type of calculation, a compare is made of the conditions of the previous specification and of the current specification. A change in conditions causes Calculation to transfer to prepare a new tag for the path of the calculation type and to process conditions. A change in conditions specified can be to new conditions or from conditions to no conditions specified.

4. Arithmetic Calculations. If the operation specified is not a compare, there will be one or two arithmetic operations specified. There may be an operation (column 29) for combining two specified factors, and an operation (column 39) for combining this result with the "result" field. Or there may be only an operation (column 39) for combining a single factor with the result field. There are four possible types of operation for combining two factors: multiplication, division, addition, or subtraction. The combination of one factor with the result field or the combination of the result of two combined factors and the result field may be by straight addition or subtraction, or reset with addition or subtraction.

   Processing of arithmetic calculation specifications is organized according to the type of calculation specified. There is a routine for multiplication and a routine for division; these will be discussed together. There is one routine for addition or subtraction specifications. In conjunction with any arithmetic calculation, there may be a specification for half-adjustment (rounding), or position-adjustment (truncation). In any operation the longest field which need be utilized is saved if it is longer than any previous maximum. This length will be used after all cards have been processed for preparation of the temporary work area. If there is no change in conditions, Calculation transfers directly to process the Calculation Operations specified.

5. Exit Link at Change in Calculation Type. Whenever there is a change in the specification of calculation type, an exit link is put out. This exit connects the block of instructions just generated to any instructions for the same calculation type which will be subsequently generated.

6. Tag on First Instruction of a Block. When there is a change in the calculation type specified, or a change in the conditions specified, a new tag is used. The current level of the sequenced tags for the specified calculation type is moved to be the tag of the first instruction. The tag is incremented in a save area. The incremented tag will be used for exits and for the tag of the first instruction of the next block of instructions for this calculation type.

7. Conditions. When there are conditions specified, the necessary compare and branch instructions are generated. If instructions are generated, they are the first instructions of the block. The exits generated for cases when the conditions are not satisfied are to the next level of the sequenced tags for the calculation type.

8. Calculation Operations

   a. Compare. The calculation card may be used to specify a compare of two factors. When a compare is specified, the instructions to make the compare are generated and Calculation transfers to process Status and Resulting Condition specifications.

   b. Multiplication and Division Routines. The routines for multiplication and division are very similar. When an X is sensed in the operation column (29) of the specification card, transfer is to the routine which generates instructions for multiplication. If a slash is sensed, Calculation transfers to the division routine. Multiplication or division requires two factors.

The length of the work field required is found. For multiplication this is the sum of the lengths of factors 1 and 2, and for division this is the length of factor 1. The work field length is saved if it is greater than the previous longest length. A test is made for the availability of multiply and divide hardware. If the hardware is available, Calculation transfers to generate the multiply or divide instructions and then goes to process Half-Adjust and Position-Adjust specifications.

If the hardware is not available, a switch is turned on to signal that the subroutine to perform the required operation is to be punched when Calculation is at End of Cards processing. Linkage to the required subroutine is generated. This not only includes a branch to the subroutine, but also data formatting instructions for the addresses of the fields which will be used and for a zero constant. Calculation then transfers to process any Half-Adjust or Position-Adjust specifications.

c. Addition and Subtraction Routine. If the operation column (29) of the specification card is a plus, a minus, or blank, Calculation transfers to the addition and subtraction routine. A plus or minus specified requires two factors. A blank implies that there is one factor to be combined with the result field. The position-adjust parameter columns are added to field length which is then compared to the longest previous field length. If the combined length is found to be longer, it is saved as the longest field length. A test is made to see if position-adjust or half-adjust is specified. (These operations must take place in a work area.)

If neither half-adjust nor position-adjust is specified, instructions to operate directly on the result field are generated. If a reset operation is specified, the first instruction generated is a binary subtract for clearing the result field. Instructions to perform the required additions or subtractions are generated. Calculation then transfers to process Status and Resulting Conditions. If either half-adjust or position-adjust is specified, Calculation generates instructions to operate with the specified factors (or factor) in a temporary work area. Calculation then transfers to process Half-Adjust and Position-Adjust specifications.

d. Half-Adjust and Position-Adjust. Half-adjust (rounding) and position-adjust (truncation) operations take place in the temporary work area. The result of a multiplication, division, addition, or subtraction of two factors can be adjusted before combining it with the result field. A test is made to see if half-adjust is specified. If it is, Calculation generates the instructions to add 5 to the specified digit. The instructions use the sign of the data being rounded so that the absolute number is rounded upward and the sign is retained. Calculation tests for position-adjust specifications. If position adjustment is specified, an instruction is generated which preserves the sign of the field. The location of the new rightmost position is incorporated into the instruction which will combine the result in the work area with the result field.

e. Combining the Work Area and the Result Field. The result of multiplication and division is always in the work area. The result of addition and subtraction for which half-adjust or position-adjust is specified will be in the work area. Also, a single factor to be combined with the result field will be in a work area when position-adjust or half-adjust is specified. If the specified operation for the combination of data in the work field requires a reset, an instruction is generated to binary subtract the result field from itself. The required addition or subtraction instruction is generated. Calculation then goes to process Status and Resulting Condition specifications.

f. Status and Resulting Conditions. The status specification is analyzed and, if present, appropriate instructions are generated. There may be up to three status parameters. These may refer to the status of the result field or the status of a compare operation.

g. Status of the Result Field. The result field may be tested for a status after calculation. The possible status specifications are the same as those specified for the Data section. These are blank, negative, positive, or zero. The generation of instructions is the same by Data. If a status referencing the result field is specified, the appropriate instructions are generated by Calculation to make the required comparisons and set the condition switch appropriately. A storage definition statement is generated using the specified condition code as part of the tag (CON(XX)) where XX is the resulting condition number. If "blank" status is specified, an identical routine is generated. This block of instructions is part of the path sequenced by the tags BLNK(XX). (See Data section: Status and Resulting Conditions.)

h. Status of a Compare Operation. There are always status specifications for a compare operation. These may be high, low, equal, or unequal. Calculation generates the instructions to test for the status and set the condition switch appropriately. A storage definition statement is generated using the specified condition code as part of the tag (CON(XX)).

When all three status parameters have been checked, Calculation transfers to read another card.

9. End of Cards. When a card is read with an asterisk in column 1, all calculation specification cards have been read and processed. Final links are generated to connect the last level of the paths sequenced by the tags DCAL(XX) and TCAL(XX) to routines which will be generated by Output. The detail line path (DCAL(XX)) terminates in a branch to the alpha header path (HA0000) generated by Output. The total line path (TCAL(XX)) terminates in a branch to the total line path (NUMT0T). If there were any specifications for half-adjust, Calculation generates a data formatting statement for the constant 5, using F1V1 as the tag. If zero status was requested in either the Data or Calculation specification cards, the necessary data formatting statements for constants and save fields are punched.

Subroutines for multiplication and division are processed by Calculation. If a multiply operation was specified and there is no hardware for multiplication and division, a switch was turned on by Calculation. A divide switch would be on for any divide specification processed. Each subroutine is read, the appropriate switch is tested and, if on, the subroutine is punched. Otherwise, punching is bypassed. An instruction to link ENTRY 3 to the detail line calculation path (DCAL00) is punched. The exits from the BLNK(XX) path are punched. The Calculation section ends by reading the Output Bootstrap into core and branching to it, thus loading the first part of the Output section.

PROCESSING BY OUTPUT PART 1

Processing of the output specification cards is done by Part 1 of the Output section. The processing of the Output specifications is organized around the line of print. Each line of output as specified by the line specification card and the field specification cards is handled as a unit. Part 1 of Output generates (for each line described by the specifications) a routine for checking conditions, moving and editing fields, and outputting the line. The types of line (six types in all) are handled in almost the same way. Because of differences (due to tag formation and sequencing hierarchy) there are several parallel processing paths which merge when processing is identical. In the following paragraphs, distinction between the several paths is made only when there is a difference in logic.

Processing of the Output specification cards is as follows:

1. Card Reading. The L-card (line description card) is read and moved to a save area. The next card is read into the read area. The L-card in the save area will be analyzed using information, if needed, from the card in the read area.
2. Overflow Conditions. The L-card is analyzed immediately to see if the line described is conditioned by overflow. If the line is conditioned by overflow:
   a. OVERSW is set on to signal to Part 2 the need of overflow linkage.
   b. The tag for the routine for the line it set up, according to whether this is the first line or a "next line."
   c. The tag is incremented for exits from this routine. The operand of the branch to a "next line" is saved if there is a next line.
   d. SW4OFL is turned on. This will signal that the line is conditioned by overflow. Therefore, the SWICHD branch to generate links to overflow checks will be bypassed.

If the line is the first line, Output transfers to Process Line Conditions. If the line is a "next line," Output transfers to Field Placement Routines.

3. Line Type. If the line is not conditioned by overflow, Output determines the line type and transfers to the appropriate routines for generating tags and setting up exits.
4. Sequence Type. If the line is a header or total line, Output tests for specification of numeric sequencing. Output branches to the routine for the specification (alpha or numeric).

5. Detail, Alpha Header and Alpha Total Lines. Specifications for detail, alpha header, and alpha total lines are processed in a similar way.

   a. First Tag of the Routine and Exits. The beginning tag of the routine is set up according to whether there are conditions specified. (No conditions imply that this is a "next line.") The level of the tag of the line type is incremented if necessary. This will be used as an exit when tests for conditions are not met. If the line specification card does not specify a "next line," this tag will be used for exit after output.

   b. SWICHC and SWICHD. SWICHC and SWICHD are modified to branch to the appropriate routines for the specified line type for final processing of the line after the Field Placement Routines have been executed.

If the line is the first of a set (conditioned), Output transfers to Process Line Conditions. Otherwise, Output tests the card in the input area to see if it implies "or" conditions. If there is not an "or" condition, Output transfers to the Field Placement Routines.

   c. "OR" Conditions. If there is an "or" relation, the conditions in the card in the read area are moved to overlay those in the save area. A new card is read and Output branches to test for Overflow Conditions and to determine the Line Type.

6. Numeric Header Lines. Numeric headers are processed in the same way as alpha header lines except for the generation of the path exit after print. If there is a "next line" specified, this exit is, as above, the specified name of the "next line." However, if there is no "next line," generating of the exit is qualified by numeric hierarchy. If the specified line is not of the lowest hierarchy level, a tag for the lower level is used. This tag is the one used for first lines (not "next line") and is incremented (current HNXXXX+1). If the specified line is of the lowest hierarchy level (and no "next line" is specified) the exit is to DD0000, the tag of the first of the detail line routines. Output transfers, after generating tag and exits for numeric header routines, to set SWICHC and SWICHD and to execute the remainder of the processing path for Detail, Alpha Header, and Alpha Total Lines.

7. Numeric Total Lines. The processing path for numeric total line specification cards is different because of the more complicated linkage in the generated routines.

   a. Tag of First Level. If the line specification is the first of the lowest level, the T(XX) is saved for use in exits from higher level routines.

   b. Set SWICHC and SWICHD and Save the Highest Hierarchy Level. When line conditions have been processed or if the line is not conditioned (it is a "next line"), Output sets SWICHC and SWICHD (see above). If the hierarchy is not level 1, the level is saved. Output then transfers to Field Placement Routines.

   c. Tag of the Numeric Total Routine. If the line is conditioned, it is the first line of a set. Output sets up the tag of the routine for the beginning of the compares for conditions.

   d. "OR" Conditions. If the card in the input area implies "or" conditions, Output generates the exit for conditions not met. This is to the tag T(XX)T00 where T(XX) is the specified name given by the card in the input area. Output then transfers to Process Line Conditions.

e. Numeric Total Routine Exit (Conditions not Met). If an "or" relationship is not specified, Output moves the TT0000 tag to the save area for the exit when conditions are not met. (The routine for the lowest hierarchy level of numeric total lines transfers to the first routine for alpha total lines.) Output tests the level of the line specification being processed. If it is the lowest level, Output transfers to Process Line Conditions. If the level is not the lowest, Output moves the tag of the next lower level routine to the save area. (The generated program checks conditions for line output in descending order of numeric sequence.) Output then transfers to Process Line Conditions.

f. Forcing Lower Hierarchy Levels. When Output has completed its routine to Process Line Conditions, it tests to see if the line specification is for a numeric total line. If the line is a numeric total, Output generates instructions to turn on switches for the line and all lower level lines. Output then transfers to set up the Tag of the Numeric Total Routine. If there are no "or" conditions, Output transfers to Set SWICHC and SWICHD and Save the Highest Hierarchy Level.

8. Process Line Conditions. Every first line specification card contains conditions for the output of the line and any "next line." Output generates instructions for testing the condition switches specified and for branching when the conditions are not met. These exit branches are prepared before the routine to process the line conditions is executed; they vary according to the line type and sequencing specified. When Output has completed generating these instructions, it tests to see if the line is a numeric total line. If it is, Output branches to generate instructions for Forcing Lower Hierarchy Levels. If the line is not a numeric total line, Output tests for an "or" relation. If there is not an "or" relation, Output transfers to the Field Placement Routines. If there is an "or" specification, Output moves the new conditions to the save area, reads another card, and transfers to test for Overflow Conditions and determine the Line Type.

9. Field Placement Routines. The field placement routines generate the instructions for building the output line. The field specifications for each line are read and processed in the input area. When Output transfers to the Field Placement Routines, the first card is in the input area. Output continues reading and processing until an L-card (or an asterisk card) is read. Output tests for conditions on the field. If any are specified, Output generates compare and branch instructions to test the conditions. If there is an edit word specified, Output generates the instructions to perform the edit. Instructions to move the field are generated. If the card specifies blanking, Output generates the required instruction to blank the field after its contents have been moved to the output area. A constant specification causes Output to generate the appropriate data formatting statement and move instruction.

10. Finishing the Routine. When an L-card is read, all field cards for the previous line specification have been processed. Output completes the routine for the line-card in the save area. Output generates links to output subroutines according to specifications processed by the Control section and stored in a communication area. The spacing and skipping specifications in the line card are analyzed and the appropriate links to the Carriage Control Subroutine are generated. If a skip was specified, no test for overflow or channel 12 is required. Output transfers to SWICHC. If the line is conditioned by overflow, no test for overflow or channel 12 is required. Output transfers to SWICHC.

a. SWICHD. If no skip was specified and the line is not conditioned by overflow, Output generates appropriate channel 12 and overflow checks for the line type. Total lines must be put out before skipping to a new page. Therefore, only a link to test channel 12 is generated. Header and detail line specifications cause Output to generate overflow tests. Output generates a link to the overflow routine and thus to the routines for lines conditioned by overflow. When these instructions have been generated, Output transfers to SWICHC.

11. SWICHC. SWICHC is used to transfer processing to the appropriate routines for generating final links. Output generates an exit branch after print. This exit was set up during the first processing of the line. This can be to a "next line" or to the appropriate level of the variable tags for condition checking. In the case of numeric total lines, Output tests before generating the exit instruction for a "next line" specification. If there is not a next line specification, Output generates instructions to test the hierarchy switches before the exit instruction. Output tests the card in the read area. If it is an L-card, Output moves it to the save area, reads a new card, and transfers to test for Overflow Conditions and determine the Line Type.

12. End of Cards. When Output senses an asterisk in column 1, all cards have been read and processed. Output reads the bootstrap into core and branches to it, thus reading in Output Part 2.

PROCESSING BY OUTPUT PART 2

The second part of Output generates final links and any necessary routines and data formatting statements for completing the logic of the generated program. Processing of the Output Specifications cards is as follows:

1. Links. Output punches a link from the alpha header line routine to the numeric header line routine, and a link from the numeric header line routine to detail line routines. Instructions are punched to modify the exit of the BLNK routine so that exit is to SER00, if from the total line path and to READ1, if from the detail line.

2. Total Logic.

   a. Output tests to see if a numeric total line was specified. If any were, Output punches a link from NUMTOT to the highest TXXT00 tag. The data formatting statements for the hierarchy switches are punched. A set of instructions are punched to reset the hierarchy switches. If there were no numeric total lines specified, Output punches a link from NUMTOT to TT0000. If there were no alpha total lines specified, Output punches a link from TT0000 to TBLNK1.

   b. Punched Output Only and Overflow Routines. If there were total line specifications processed, Output (after completing the total logic) tests for a switch setting. The switch is on if the output of the generated program will be punched only. If the switch is on, Output goes to punch a link between TT0000 and TBLNK1. Output bypasses punching of overflow routines. This link is also punched because there were no total lines specified. If the switch for punched output only is on, overflow routines are not punched. If the switch is off, overflow checking instructions are bypassed but OVRFL0 links are punched. If total line specifications were processed and the switch for punched output only is off, overflow checking routines are punched. The first instruction will have the highest level TT(XXXX) tag. If the OVRSW is on (see Overflow Condition - Part 1), there were lines conditioned by overflow. Links are punched to the routines conditioned by overflow.

c. Numeric Header Links. If there was more than one hierarchy level of header lines, links from header to detail line routines are punched.

d. Final Halt. The final halt for the generated program is punched. The halt will give an A-address register display of zero, and the B-address register will indicate 4777(octal).

e. CARRIAGET Card. OUTPUT reads a comment card and the CARRIAGET card. The CARRIAGET card contains information about the output page design. Output punches data formatting statements for the lines per page, the line count, and the channel 12 equivalent line number. Output processes and punches the specified channel and line number equivalents.

f. Final Statements. Output uses information processed by the Control Section and stored in a communication area for reserving core for the physical input data record. Output punches DCW's and reserve statements according to record length, blocking, and banner character specifications. Output completes the generated program with bootstrap definition, a clear statement, and an END card.

SECTION IV

THE GENERATED OBJECT PROGRAM

BLOCK DIAGRAM OF THE GENERATED OBJECT PROGRAM

Figures 4-1 and 4-2 show block diagrams of the logic path followed by the object program generated by the Series 200 Report Generator with card and tape inputs, respectively. The various steps through which the (card input) program moves in executing the Series 200 RG object program are numbered on the diagram in Figure 4-1 and explained in the correspondingly numbered paragraphs.

Punched Card Input

1. The object program first generates an instruction to read the data card (or card image) in the input area. This instruction, labeled READ 1, is the only read instruction used in the object program.

2. The first test made by the object program on the record in the input area is to determine whether this card is the last card in the file. If the last card is present, all control-field change indicators are set to ON and SW4 is modified to a halt. The program then branches to the routine labeled TCALxx (step 11). The total calculations specified are then performed; all total lines are printed (including those which contain last card, LC, conditions); and the program comes to a normal halt.

3. Immediately following the test for last card, if the result is negative, the programmer may exercise the option to insert subroutines or own coding in accordance with the rules stated earlier (see Section I: Insertion of Own Coding and Subroutines).

4. The object program next uses a routine which tests the input data for the presence of non-sequential records. A routine has been generated in the object program for each non-sequential record specified by the programmer as being in the input data file. The first such routine is labeled DA0100, the second DA0200, and so on to the last non-sequential record labeled DAxxxx (where xxxx is the highest numerical designation assigned to the non-sequential records in the file). As each routine is entered, a test is made of the record which has been read to determine whether it contains the record codes specified for a non-sequential record in one line entry of the Input Specifications sheet. If the distinguishing record codes are not found, the program branches to the next routine and checks the record for the codes specified in the next Input Specifications sheet entry. This process continues until the routine finds one of the non-sequential records specified. The routine then sets to ON the resulting condition indicator which designates the particular record codes and sets SW4 to branch to the routine which will remove the data fields from this input record type. The resulting condition indicator stays ON until the next input record is read. The branch routine for removing the data fields from the input record is entered at step 14.

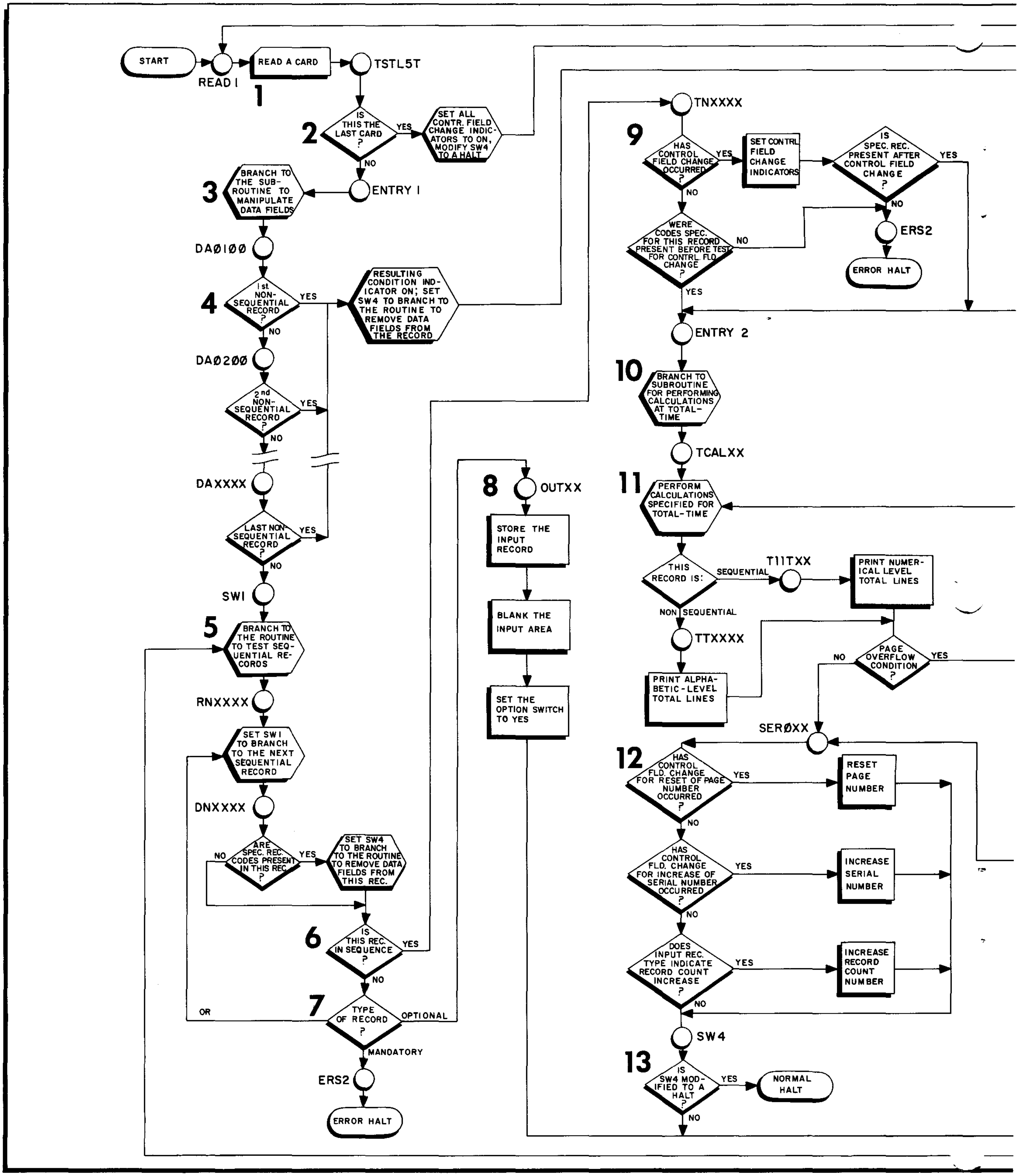

Figure 4-1. Block Diagram of the Generated Object Program (Card Input)

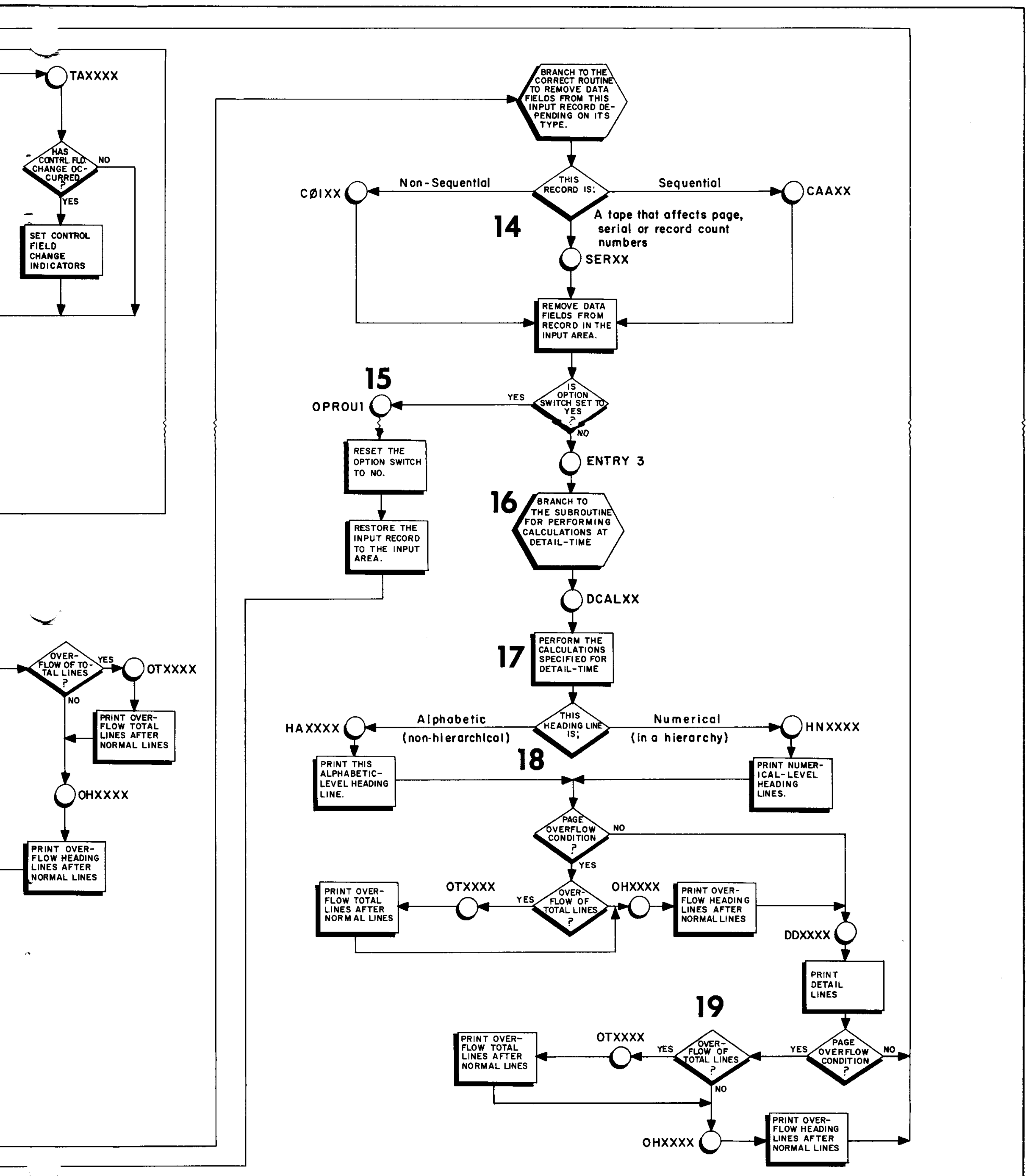

Figure 4-1 (cont). Block Diagram of the Generated Object Program (Card Input)

5. When the routine described in step 4 has fully tested the input record to determine whether it is one of the non-sequential records specified and none of the non-sequential records is found, the program moves to the routine labeled SW1 and starts testing for sequential records (numerically designated sequence entries on the Input Specifications sheet). The routine begins with a portion labeled RN0100 which sets SW1 to branch to the routine for the next sequential record. The routine labeled DN0100 tests for the record codes specified for the record entry on the Input Specifications sheet. If the specified record codes are present, SW4 is set by the routine to branch to step 14 where the data fields are removed from this record.

6. If a record is out of numbered sequence, it will not be found by the program. In the event that the record specified on the Input Specifications sheet is not present at the completion of step 5, the program goes to an error halt, ERS2. The programmer can insert in the object program an error routine labeled ERS2 which will be used for all errors in record sequence. An error halt also occurs when a specified sequential record is absent during the TNxxxx routine (step 9) and is still not present after a control field change occurs.

7. The object program tests every record to determine what kind of record is present in the input area. A sequential record is first tested to determine whether the record codes specified for it are present. If the record codes are present, handling is as stated in step 5. If the record is out of numbered sequence, as indicated in step 6, before proceeding to the error halt, ERS2, the object program tests to further determine the record type. A check is made to determine if another record in an "or" relationship has been specified. If such is the case, the program branches to SW1 (step 5) to search for the second record. If, however, only one sequential record with the particular record codes was specified, and that record, whose presence is mandatory is not found, the program proceeds to the error halt unless the record is an optional type.

8. An optional routine, labeled OUTxx, is generated for every type of sequential record which has been specified as optional. The object program searches for a specified optional record, and if the record is not found the program branches to the optional routine, OUTxx. It is the function of the optional routine to store the record which is in the input area for later use; to blank the input area; and to set the option switch (step 15) to YES. The program then branches to the routine to remove data fields from the input record. Since the record in the input area has been blanked, the data fields which are removed will consist of blanks.

9. It can be seen in steps 2 through 7 that the object program tests the input record to determine what kind of record it is, and therefore, how it shall be handled. When these factors have been determined, the program branches to the routine to test for control-field changes which may have occurred since the previous record was processed. Separate routines, labeled TAxxxx and TNxxxx, exist for non-sequential and sequential records, respectively. If control fields are alike in all sequential records, only one routine will have been generated for this testing operation. The same procedure applies to non-sequential records when all have the same control fields.

10. After the point in the program where control-field changes have been tested for, the programmer can insert calculation routines before entering the routine to perform calculations specified for total-time. Total-time calculations will not contain information extracted from the record

in the input area. At this point in the program, no data fields have been removed from the input record; fields are removed at step 14, as part of the routine labeled SW4. The calculations which are specified for performance at total-time exist in the object program as routines labeled TCALxx. These routines are now entered by the program and the total-time calculations performed.

11. Printed output of the total lines occurs after total calculations, provided that the output conditions for these lines have been fulfilled. The routines for printing these lines are labeled T11Txx (in the case of lines which are in a hierarchy designated by numerical level) or TTxxxx (for those lines which are not in a hierarchy; i. e., alphabetic-level designation). At this time, a test for page overflow is made, and if an overflow condition exists, the routine labeled OTxxxx for overflow total lines and/or the routine labeled OHxxxx for overflow heading lines is entered. Normal total lines are printed before any overflow lines.

12. A routine labeled SER0xx is entered after total-line printing at this point to perform page-number, serial-number, and record-count number operations. Incrementing of these numbers is performed by the routine if the conditions governing their increase are fulfilled.

13. The entrance to the routine labeled SW4 is a branch instruction which causes the program to enter the proper routine to remove the data fields from the input record.

14. The line entries on the Input Specifications sheets which specify each record in the input file are made in such a way that sequential and non-sequential records are distinguishable. Non-sequential records have alphabetic-level designations (such as CAA, CBB, etc.), and sequential records have numerical-level designations (such as C01, C02, etc.). For each of these designations, a corresponding routine exists in the object program, as well as the routine labeled SERxx, for processing the page, serial, and record count numbers.

15. When a sequential record is optional, absence of the record in the sequence causes a branch to the optional routine which is labeled OPROU1. The routine is entered by the object program after removing the blank data fields from the record. The object program branches when the Option switch is set to YES. The record saved in step 8 is restored to the input area and the Option switch reset to NO. The object program then branches back to the SW1 routine and tests for the record which should be next in the sequence. The next record in sequence should be the one which has been restored to the input area.

16. The third point in the object program where the programmer may insert own coding and subroutines is just before the routine which performs detail calculations. The insertion point is labeled ENTRY 3.

17. The detail calculation routine, labeled DCALxx, is entered, and the calculations specified for detail-time are performed on the input data.

18. Heading lines whose conditions are now fulfilled are printed. Heading lines which are in a hierarchy (routine labeled HNxxxx) are printed after those heading lines which are not in a hierarchy (routine labeled HAxxxx). A test for overflow is made, and if the test is positive, either overflow heading lines or overflow total lines or both are printed by entrance into the routine labeled OHxxxx or OTxxxx, respectively.

19. The object program enters the routine labeled DDxxxx to print detail lines. A test is again made for overflow, and the overflow lines (if any) are printed after normal lines, as in step 18. The object program then returns to READ 1, to process the next input record.

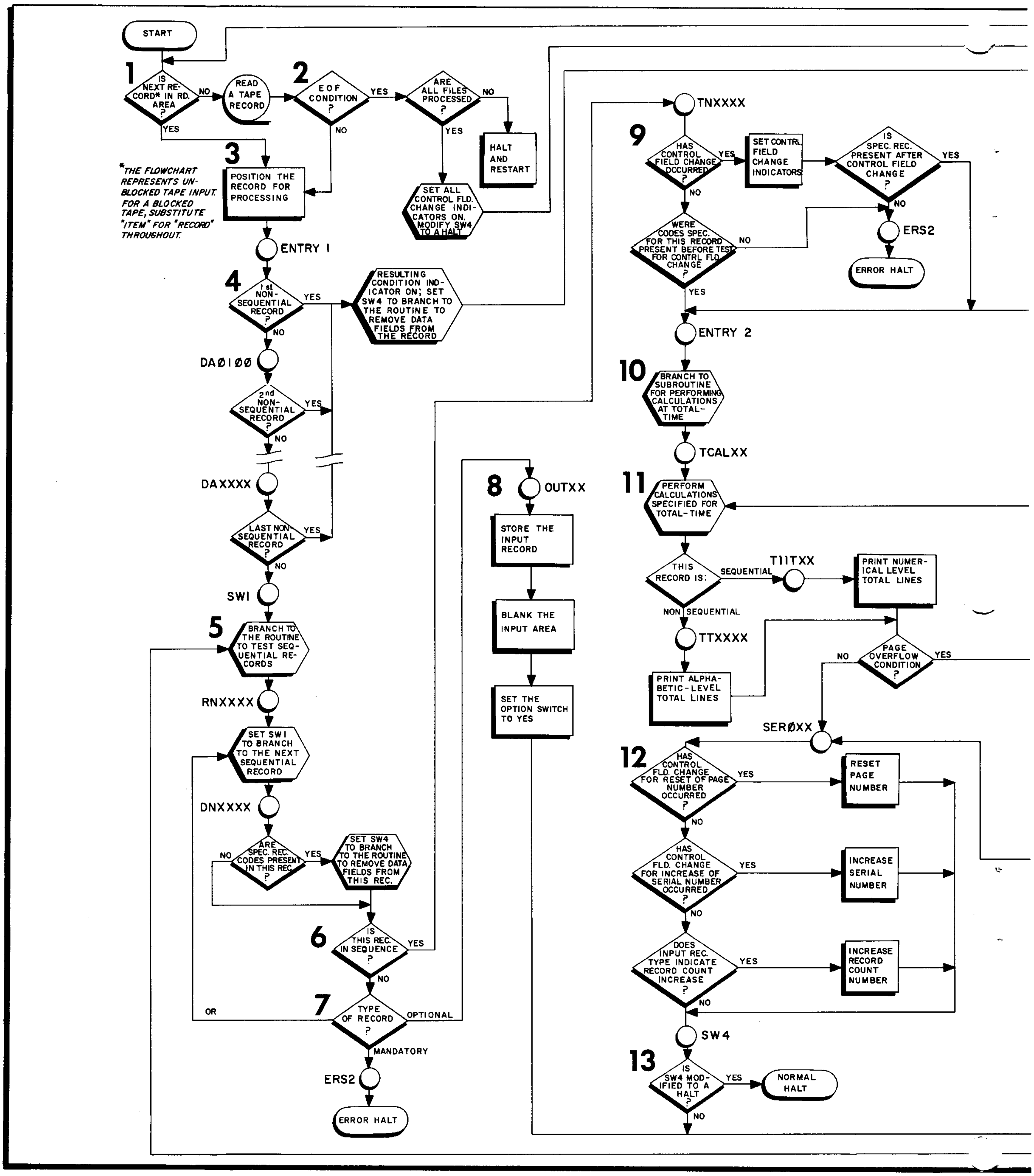

Figure 4-2. Block Diagram of the Generated Object Program (Tape Input)

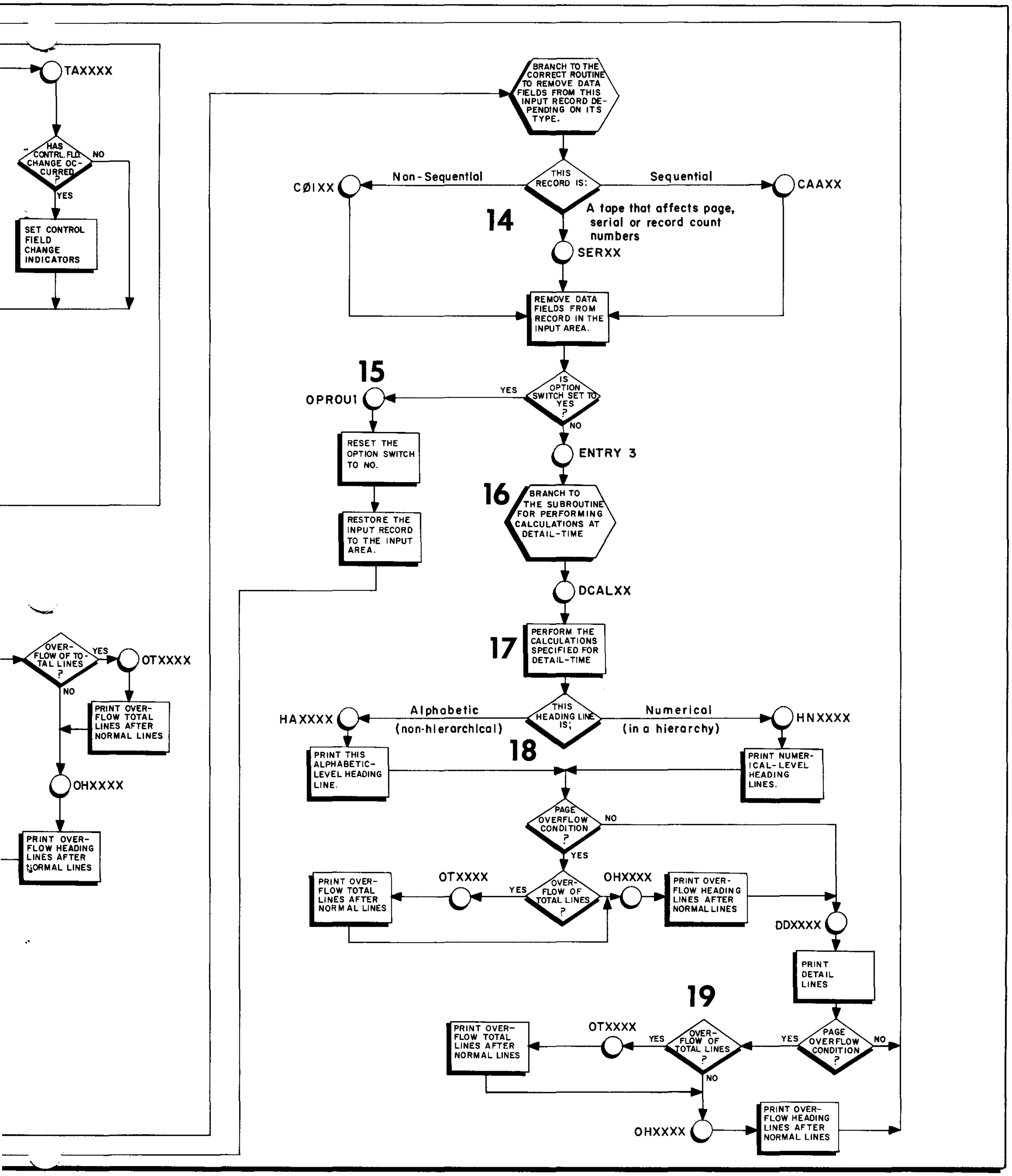

Figure 4-2 (cont). Block Diagram of the Generated Object Program (Tape Input)

Magnetic Tape Input

Processing logic of the generated object program when using magnetic tape input is in nearly all respects identical to that of the program when using punched card input. Figure 4-2 shows the Block Diagram of the Generated Object Program with magnetic tape input. The various steps through which the (tape input) program moves in executing the Series 200 RG object program are numbered on the diagram and explained in the correspondingly numbered paragraphs below.

1-3. The generated object program, when using magnetic tape input in which the tape file contains blocked records, will either read a tape (step 2) or position the item for processing (step 3) after determining whether or not the next item is in the read area (step 1). When dealing with unblocked tape records, the object program always reads a tape record, and does not use step 3.

Whenever a tape read instruction is executed, the object program immediately checks to determine whether the record is a 1EOFΔ record (End of File). (See also Tape File Termination, Section VII.) If the end-of-file condition is present, the program enters into the end-of-file routine, either branching to the routine to perform calculations specified for total-time (TCALxx) if all files have been processed, or halting if more files remain to be processed. In the event that the latter course is taken, the next reel may be mounted and the program resumed in the normal way (see Section VI, OPERATING PROCEDURES).

4-19. Steps 4 through 19 are followed in exactly the same way by the object program as in the case of card input.

SECTION V

SAMPLE SERIES 200 REPORT GENERATOR PROGRAM

PROGRAM DESCRIPTION

The report specifications which are illustrated on the following pages are used by the Series 200 Report Generator to generate a program whose output will be a Monthly Special Checking Account Report such as a bank might produce for the purpose of furnishing monthly statements to its depositors.

Input to the generated program will be in the form of punched cards; output of the generated program will be mainly a printed report. The final balance at the end of the month will not only be printed but will also be punched out on a card in such a form that it may be used as input for the next month's report; i.e., the punched card will serve as the balance brought forward for the following month's report.

There are three sample pages of the printed report shown in Figures 5-13 through 5-15, inclusive. Other illustrations of the Monthly Special Checking Account Report contained herein are:

1. Figure 5-1, Report Layout.
2. Figure 5-2, Control Card for Sample Program.
3. Figure 5-3, CARRIAGET Card for Sample Program.
4. Figure 5-4, Input Specifications.
5. Figure 5-5, Data Specifications.
6. Figure 5-6, Calculation Specifications.
7. Figures 5-7 through 5-11, inclusive, Output Specifications.
8. Figure 5-12, Section of Input Data File.

EXPLANATION OF THE SAMPLE PROGRAM

The purpose of the sample program is to produce, each month, a statement of the activities taking place in each individual special checking account since the last reporting date. The bank prepares statements at the close of each month's business. Statements are prepared in duplicate; the original report copy is sent to the depositor, the carbon retained in the bank's files.

Sample Program Report Layout

The desired format of the report is illustrated in Figure 5-1. Note that the constant information has been lettered on the layout form, exactly as it will appear on every page of each report.

Variable information, dependent upon the activities taking place within each account and entered as input data, is circled on the layout form with the field name of each variable shown.

Sample Program Input Specifications

The input specifications shown in Figure 5-4 define the input file which will be used to prepare the report each month. The first three entries in Figure 5-4 define fixed-sequence records which must always be present. The last two entries define optional records (X in column 5), in no fixed sequence, and indicate that any number of these records may be present (N in column 4). Each record type is identified by a record code in column 80 of the card. Each record is further identified by a unique resulting condition number, and each record has a single control field, seven characters in length, ending in column 7. This control field, designated F1 (for control field 1), is the checking account number. A change in control field number causes printing of all totals (total checks cashed, total charges for checks, total deposits, and final balance), a skip to the head of the next form, and start of printing on the next monthly report in the sequence of account numbers.

Sample Program Data Specifications

The data specifications define the data fields from the input data file that will be used in producing the desired report. These data fields are contained within the records specified in the input specifications. As shown in Figure 5-5, the record C01 includes data fields which contain the checking account number, and the depositor's name and address. The C02 record contains, in addition to account number, only the amount of balance brought forward for that account. The C03 record contains the service charge for the monthly statement and the data of the report. Note that the date is contained in two fields, one used only for the year, thus necessitating a relatively small change to revise the report date each month. The two optional records, CAA and CBB, are used to enter the date and amount of checks and the date and amount of deposits, respectively. The last data specification, PAGENO, is used to reset the page number to 1 upon a control field change. In the sample program illustrated, each depositor will receive a one-page statement. More active accounts, however, might require several report pages for a complete statement. In such a case, when another checking account number is encountered, printing of all total lines (which takes place only with a control-field change) and a skip to head of form with numbering of the new report from page one would take place.

Sample Program Calculation Specifications

In the calculation specifications are defined the arithmetic manipulations which shall be performed on the data fields in order to produce the various outputs for the monthly special checking account report. The meaning of each line entry in Figure 5-6 is listed in the following tabulation.

Card Number	Meaning
010	The field, BALANC, is first set to zero, then the seven-character BBFORD field is added to it. Conditions for the operation are that a control-field change shall not have occurred (NF1) and the record from which BBFORD is obtained must be record type C02. This is a detail line.
020	The two-character field, CHRGE, is added to the AMOUNT field and the sum is moved to the seven-character BALIN field after the latter has first been set to zero. Conditions for performing the operations are that a control-field change shall not have occurred and the record type must be C04. This is a detail line.
030	The two-character field, SERVIC, is subtracted from the seven-character BALANC field and the remainder is added to the BALNCS field after the latter has been set to zero. The status of the BALNCS field after the operation, if negative, sets a condition indicator (X09) to show that the account is overdrawn at this point. Conditions for performing the subtraction operation are that a control-field change shall not have occurred and the record type from which the SERVIC field is obtained must be a C03. This is a detail line.
040	The contents of the BALIN field are subtracted from the BALNCS field (field name is left blank in this line entry since the last data field specified as the result of a preceding calculation is affected by the present calculation). After the subtraction, the sign of the BALNCS field is tested; if it is negative, a condition indicator (X06) is set, showing an overdraft condition at this point. Conditions for performing the subtraction operation are: (1) a control-field change shall not have occurred; (2) the record type shall be CAA; and (3) the record-type shall not be C03. This is a detail line.
050	The seven-character field, DEPOST, is added to the BALNCS field. After the addition, the field status is tested and, if it is negative, a condition indicator (X07) is set to show that an overdraft condition exists. Conditions governing the addition operation are that a control-field change shall not have occurred and the record type from which the DEPOST field is obtained must be CBB and cannot be C03. This is a detail line.
060	The one-character field named NUMBER is added to the field NUMCHK. Since the contents of the NUMBER field is always one, this operation serves to increment by one the figure shown in the "Number of Checks Paid" column of the printed report for each debit item. Conditions for performing the addition operation are that a control-field change shall not have occurred and that the NUMBER field is obtained from a CAA record type. This is a detail line.
070	The two-character field, NUMCHK, upon a control-field change, is subtracted from itself at total time. The effect here is to zero the NUMCHK field in preparation for printing the next report.
080	The seven-character BALNCS field is reset-added into the FINBAL field at total time and after a control-field change has occurred. After the reset-add operation, the status of the FINBAL field is tested and, if it is found to be negative, an indicator (X08) is set to show an overdraft condition.

Card Number	Meaning
090	The seven-character DEPOST field is added to the DEPTOT field. Conditions for performing the addition are that the DEPOST field must be obtained from a CBB record type and that a control-field change shall not have occurred. This is a detail line.
100	The DEPTOT field is subtracted from itself (zeroed) at total time and upon fulfillment of the resulting condition 02 (a BBFORD record is present).
110	The two-character CHRGE field is added to the TOTCHG field when two conditions are met: (1) the CHRGE field must be obtained from a CAA record type; and (2) no control-field change has occurred. This is a detail line.
120	The TOTCHG field is zeroed by subtracting it from itself upon the condition that a C02 record type (BBFORD data field) is present at total time.
130	The seven-character AMOUNT field is added to the CHKTOT field when two conditions are met: (1) the AMOUNT field must be obtained from a CAA record type; and (2) no control-field change has occurred. This is a detail line.
140	The CHKTOT field is zeroed by subtracting it from itself upon the condition that a C02 record type is present at total time.

Sample Program Output Specifications

The output specifications for the monthly special checking account report (Figures 5-7 through 5-11) define the position and contents of each line that is to be printed. To understand the output specifications line entries better, compare these forms with the Report Layout, Figure 5-1.

Three kinds of lines appear on the report: heading, detail, and total lines. On this report, all heading and all total lines are in a hierarchy, while the detail lines are of two kinds – those in a hierarchy, and those which are independent. In this report there are eight levels of heading line; a change in control field one forces a skip to the head of the next form and printing of all eight heading lines. Of the two types of detail lines which may be present, only that one line which is in a hierarchy, D11, will always be present. This detail line is the balance brought forward; it will be printed on every report. The DAA and DBB detail lines which show checks cashed and deposits, respectively, will only be printed if these activities have taken place during the reporting period. Last, the total lines in a hierarchy specify printing of report totals. Two levels of total lines exist; the major level, T2x, forces printing of the minor level, T1x.

Note that line T22 does not appear on the report layout. The output of this line is punched into a card and consists of the final balance, checking account number, and a B punched in column 80. This punched card will be used as input to the report program for the following month to furnish the balance-brought-forward information.

REPORT LAYOUT

PROGRAM ______ PROGRAMMER ______ DATE ______ PAGE Ø1 OF Ø9 REPORT NAME MONTHLY SPECIAL CHECKING ACCOUNT REPORT

COMMENTS ______

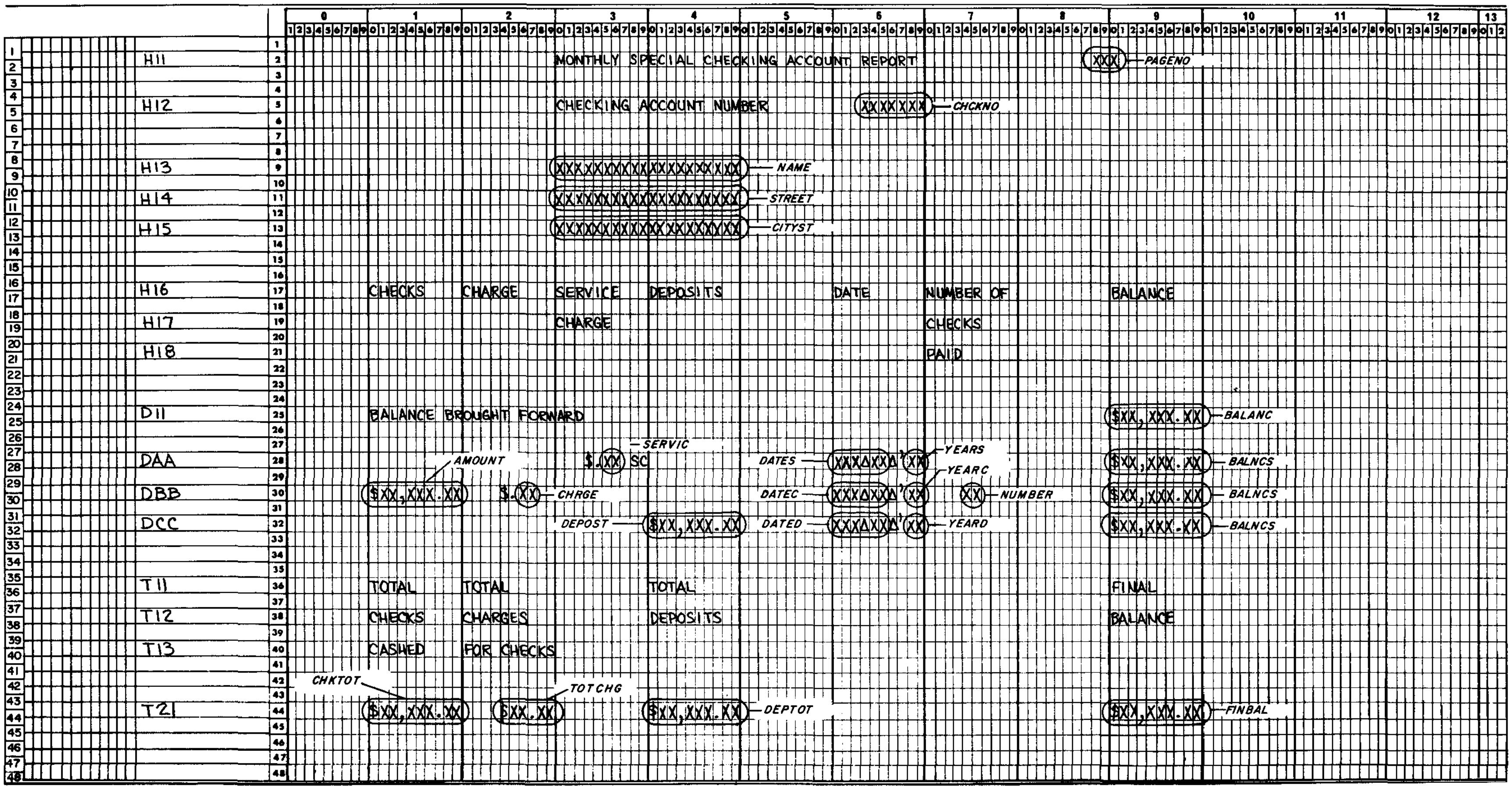

Figure 5-1. Report Layout of Monthly Special Checking Account Report

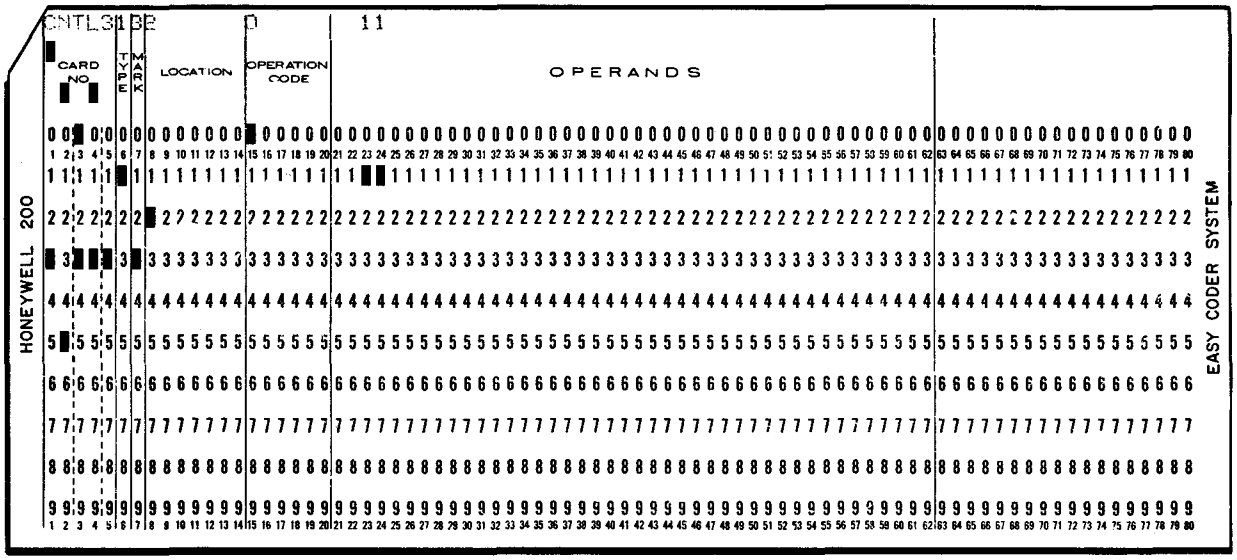

Figure 5-2. Control Card for Sample Program

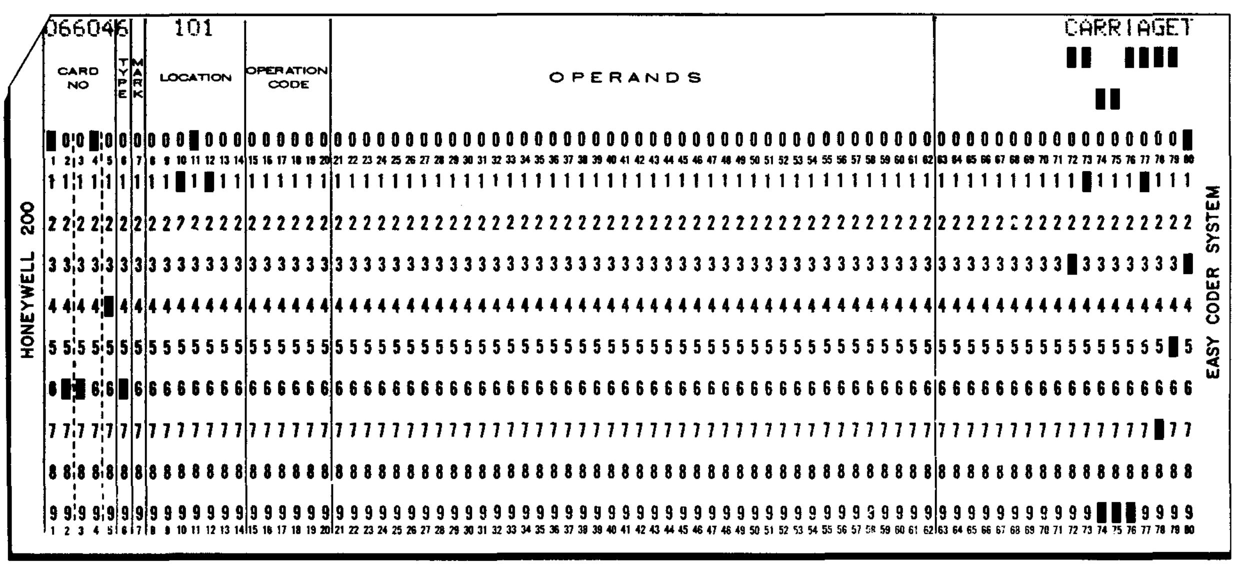

Figure 5-3. CARRIAGET Card for Sample Program

INPUT SPECIFICATIONS

REPORT NAME MONTHLY SPECIAL CHECKING ACCOUNT REPORT PROGRAMMER ____ DATE ____ PAGE Ø2 (76 77) OF 09

RECORD SEQUENCE: C (1)	SEQUENCE (2-3)	NUMBER (4)	OPTION (5)	RECORD CODES: POSITION (6-8)	NEGATE (9)	Z/D/C (10)	CODE (11)	POSITION (12-14)	NEGATE (15)	Z/D/C (16)	CODE (17)	POSITION (18-20)	NEGATE (21)	Z/D/C (22)	CODE (23)	POSITION (24-26)	NEGATE (27)	Z/D/C (28)	CODE (29)	POSITION (30-32)	NEGATE (33)	Z/D/C (34)	CODE (35)	POSITION (36-38)	NEGATE (39)	Z/D/C (40)	CODE (41)	RESULTING CONDITION (42-43)	CONTROL FIELDS: FIELD 1 END (44-46)	FIELD 1 LENGTH (47-48)	FIELD 2 END (49-51)	FIELD 2 LENGTH (52-53)	FIELD 3 END (54-56)	FIELD 3 LENGTH (57-58)	FIELD 4 END (59-61)	FIELD 4 LENGTH (62-63)	FIELD 5 END (64-66)	FIELD 5 LENGTH (67-68)	FIELD 6 END (69-71)	FIELD 6 LENGTH (72-73)	(74-75)	CARD NUMBER (78-80)
C	Ø1	1		Ø8Ø		C	#																					Ø1	ØØ7	Ø7												010
C	Ø2	1		Ø8Ø		C	B																					Ø2	ØØ7	Ø7												020
C	Ø3	1		Ø8Ø		C	S																					Ø3	ØØ7	Ø7												030
S	CF	1																																								040
C	AA	N	X	Ø8Ø		C	C																					Ø4	ØØ7	Ø7												050
C	BB	N	X	Ø8Ø		C	D																					Ø5	ØØ7	Ø7												060
																																										070

Figure 5-4. Input Specifications, Monthly Special Checking Account Report

DATA SPECIFICATIONS

REPORT NAME MONTHLY SPECIAL CHECKING ACCOUNT REPORT PROGRAMMER ____ DATE ____ PAGE Ø3 (76 77) OF Ø9

D (1)	FIELD NAME (2-7)	FIELD LENGTH UNEDITED (8-10)	FIELD STATUS: STATUS (11)	RESULTING CONDITION (12-13)	STATUS (14)	RESULTING CONDITION (15-16)	STATUS (17)	RESULTING CONDITION (18-19)	DATA FIELD SOURCES: FIELD SOURCE (20-22)	NUMERIC (23)	FIELD END (24-26)	FIELD LENGTH (27-29)	OPERATION (30)	CONDITION (31-33)	AND CONDITION (34-36)	FIELD SOURCE (37-39)	NUMERIC (40)	FIELD END (41-43)	FIELD LENGTH (44-46)	OPERATION (47)	CONDITION (48-50)	AND CONDITION (51-53)	FIELD SOURCE (54-56)	NUMERIC (57)	FIELD END (58-60)	FIELD LENGTH (61-63)	OPERATION (64)	CONDITION (65-67)	AND CONDITION (68-70)	(71-75)	CARD NUMBER (78-80)
D	CHCKNO	ØØ7	B	77					CØ1		ØØ7																				010
D	NAME	Ø2Ø							CØ1		Ø27																				020
D	STREET	Ø2Ø							CØ1		Ø47																				030
D	CITYST	Ø2Ø							CØ1		Ø67																				040
D	BBFORD	ØØ7							CØ2		Ø14																				050
D	SERVIC	ØØ2							CØ3		Ø31																				060
D	DATES	ØØ7							CØ3		Ø56																				070
D	YEARS	ØØ2							CØ3		Ø59																				080
D	NUMBER	ØØ1							CAA		ØØ8																				090
D	AMOUNT	ØØ7							CAA		Ø19	ØØ7																			100
D	CHRGE	ØØ2							CAA		Ø29	ØØ2																			110
D	DATEC	ØØ7							CAA		Ø56																				120
D	YEARC	ØØ2							CAA		Ø59																				130
D	DEPOST	ØØ7							CBB		Ø49	ØØ7																			140
D	DATED	ØØ7							CBB		Ø56																				150
D	YEARD	ØØ2							CBB		Ø59																				160
D	PAGENO	ØØ3							PAG					ΔF1																	170

Figure 5-5. Data Specifications, Monthly Special Checking Account Report

CALCULATION SPECIFICATIONS

REPORT NAME MONTHLY SPECIAL CHECKING ACCOUNT REPORT PROGRAMMER ________ DATE ________ PAGE Ø4 OF Ø9

A (1)	Field Name (2–7)	Field Length Unedited (8–10)	Field Status: Status (11)	Resulting Condition (12–13)	Status (14)	Resulting Condition (15–16)	Status (17)	Resulting Condition (18–19)	Factor 1: Multiplicand Dividend Augend or Minuend (20–25)	Field Length Unedited (26–28)	Operation + - × / c (29)	Factor 2: Multiplier Divisor Addend or Subtrahend (30–35)	Field Length Unedited (36–38)	A/S/Ø/Ø (39)	Condition (40–42)	—And— Condition (43–45)	—And— Condition (46–48)	Total/Detail (49)	Half Adjust (50–51)	Position Adjust (52–53)	(54–75)	Card Number (78–80)
A	BALANC	ØØ7										BBFORD	ØØ7	± Ø	NF1	Ø2		D				010
A	BALIN	ØØ7							AMOUNT	ØØ7	+	CHRGE	ØØ2	± Ø	NF1	Ø4		D				020
A	BALNCS	ØØ7	N	Ø9					BALANC	ØØ7	-	SERVIC	ØØ2	± Ø	NF1	Ø3		D				030
A			N	Ø6								BALIN	ØØ7	S	NF1	Ø4	NØ3	D				040
A			N	Ø7								DEPOST	ØØ7	A	NF1	Ø5	NØ3	D				050
A	NUMCHK	ØØ2										NUMBER	ØØ1	A	NF1	Ø4		D				060
A												NUMCHK	ØØ2	S	F1			T				070
A	FINBAL	ØØ7	N	Ø8								BALNCS	ØØ7	± Ø	F1			T				080
A	DEPTOT	ØØ7										DEPOST	ØØ7	A	Ø5	NF1		D				090
A												DEPTOT	ØØ7	S	Ø2			T				100
A	TOTCHG	ØØ4										CHRGE	ØØ2	A	Ø4	NF1		D				110
A												TOTCHG	ØØ4	S	Ø2			T				120
A	CHKTOT	ØØ7										AMOUNT	ØØ7	A	Ø4	NF1		D				130
A												CHKTOT	ØØ7	S	Ø2			T				140

Figure 5-6. Calculation Specifications, Monthly Special Checking Account Report

OUTPUT SPECIFICATIONS

REPORT NAME MONTHLY SPECIAL CHECKING ACCOUNT REPORT PROGRAMMER ________ DATE ________ PAGE Ø5 OF Ø9 (76 77)

Format (1)	Line Type (2)	Line Level (3)	Line Number (4)	Output Print (5)	Output Punch (6)	Reserved (7)	Next Line Type (8)	Next Line Level (9)	Next Line Number (10)	Space Before (11–12)	Space After (13–14)	Skip Before (15–16)	Skip After (17–18)	Stacker (19)	Line Output Conditions: Condition (20–22)	—And— Condition (23–25)	—And— Condition (26–28)	Field Name (29–34)	Field End (35–37)	Field Output Conditions: Condition (38–40)	—And— Condition (41–43)	—And— Condition (44–46)	Zero Suppress (47)	Field Length (48–50)	Constant or Edit Control Word (51–75)	Card Number (78–80)
L	H	1	1	X			H	1	2			Ø1			Ø1											010
K																			Ø53					Ø24	MONTHLYΔSPECIALΔCHECKING	020
K																			Ø68					Ø15	ΔACCOUNTΔREPORT	030
F																		PAGENO	Ø9Ø							040
L	H	1	2	X			H	1	3	Ø2																050
K																			Ø52					Ø23	CHECKINGΔACCOUNTΔNUMBER	060
F																		CHCKNO	Ø69							070
L	H	1	3	X			H	1	4	Ø3																080
B																		NAME	Ø49							090
L	H	1	4	X			H	1	5	Ø1																100
B																		STREET	Ø49							110
L	H	1	5	X			H	1	6	Ø1																120
B																		CITYST	Ø49							130
L	H	1	6	X			H	1	7	Ø3																140
F																		WORDØ1	Ø19							150
F																		WORDØ2	Ø29							160
K																			Ø39					Ø10	SERVICE	170
F																		WORDØ3	Ø49							180
K																			Ø63					ØØ4	DATE	190
K																			Ø78					ØØ9	NUMBERΔOF	200

Figure 5-7. Output Specifications, Sheet 1 of 5, Monthly Special Checking Account Report

OUTPUT SPECIFICATIONS

REPORT NAME MONTHLY SPECIAL CHECKING ACCOUNT REPORT PROGRAMMER ____ DATE ____ PAGE Ø6 OF Ø9

Format (1)	Line Type (2)	Line Level (3)	Line Number (4)	Output Print (5)	Output Punch (6)	Reserved (7)	Next Line Type (8)	Next Line Level (9)	Next Line Number (10)	Space Before (11-12)	Space After (13-14)	Skip Before (15-16)	Skip After (17-18)	Stacker (19)	Line Output Condition (20-22)	And Condition (23-25)	And Condition (26-28)	Field Name (29-34)	Field End (35-37)	Field Output Condition (38-40)	And Condition (41-43)	And Condition (44-46)	Zero Suppress (47)	Field Length (48-50)	Constant or Edit Control Word (51-75)	Card Number (78-80)
F																		WORDØ4	Ø96							010
L	H	1	7	X			H	1	8	Ø1																020
F																		WORDØ2	Ø39							030
F																		WORDØ1	Ø79							040
L	H	1	8							Ø1																050
K																			Ø73					ØØ4	PAID	060
L	D	1	1	X						Ø3					Ø2											070
K																			Ø32					Ø23	BALANCEΔBROUGHTΔFORWARD	080
F																		BBFORD	Ø99					Ø1Ø	$ΔΔ,ΔΔØ.	090
L	D	A	A	X						Ø2					Ø3											100
B																		SERVIC	Ø36					ØØ4	$.	110
K																			Ø39					ØØ3	ΔSC	120
F																		DATES	Ø66							130
F																		WORDØ5	Ø67							140
F																		YEARS	Ø69							150
F																		BALNCS	Ø99					Ø1Ø	$ΔΔ,ΔΔØ.	160
F																		WORDØ6	1Ø2	Ø9						170
L	D	B	B	X						Ø1					Ø4											180
B																		AMOUNT	Ø19					Ø1Ø	$ΔΔ,ΔΔØ.	190
B																		CHRGE	Ø27					ØØ4	$.	200

Figure 5-8. Output Specifications, Sheet 2 of 5, Monthly Special Checking Account Report

OUTPUT SPECIFICATIONS

REPORT NAME ______ PROGRAMMER ______ DATE ______ PAGE Ø7 OF Ø9 (76 77)

FORMAT	LINE TYPE	LINE LEVEL	LINE NUMBER	OUTPUT PRINT	OUTPUT PUNCH	RESERVED	NEXT LINE TYPE	NEXT LINE LEVEL	NEXT LINE NUMBER	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	STACKER	LINE OUTPUT CONDITION	AND CONDITION	AND CONDITION	FIELD NAME	FIELD END	FIELD OUTPUT CONDITION	AND CONDITION	AND CONDITION	ZERO SUPPRESS	FIELD LENGTH	CONSTANT OR EDIT CONTROL WORD	CARD NUMBER
1	2	3	4	5	6	7	8	9	10	11 12	13 14	15 16	17 18	19	20 22	23 25	26 28	29 34	35 37	38 40	41 43	44 46	47	48 50	51 75	78 80
B																		DATEC	Ø66							010
F																		WORDØ5	Ø67							020
F																		YEARC	Ø69							030
F																		BALNCS	Ø99					Ø1Ø	$ΔΔ,ΔΔØ.	040
F																		NUMCHK	Ø75				Z		ΔØ	050
F																		WORDØ6	1Ø2	Ø6						060
L	D	C	C	X						Ø1					Ø5											070
B																		DEPOST	Ø49					Ø1Ø	$ΔΔ,ΔΔØ.	080
B																		DATED	Ø66							090
F																		WORDØ5	Ø67							100
F																		YEARD	Ø69							110
F																		BALNCS	Ø99					Ø1Ø	$ΔΔ,ΔΔØ.	120
F																		WORDØ6	1Ø2	Ø7						130
L	T	1	1	X			T	1	2	Ø3					F1	N77										140
F																		WORDØ7	Ø14							150
F																		WORDØ7	Ø24							160
F																		WORDØ7	Ø44							170
K																			Ø94					ØØ5	FINAL	180
L	T	1	2	X			T	1	3	Ø1																190
F																		WORDØ1	Ø19							200

Figure 5-9. Output Specifications, Sheet 3 of 5, Monthly Special Checking Account Report

OUTPUT SPECIFICATIONS

REPORT NAME MONTHLY SPECIAL CHECKING ACCOUNT REPORT PROGRAMMER ________ DATE ________ PAGE Ø8 OF Ø9

Format (1)	Line Type (2)	Line Level (3)	Line Number (4)	Output Print (5)	Output Punch (6)	Output Reserved (7)	Next Line Type (8)	Next Line Level (9)	Next Line Number (10)	Space Before (11-12)	Space After (13-14)	Skip Before (15-16)	Skip After (17-18)	Stacker (19)	Line Output Condition (20-22)	And Condition (23-25)	And Condition (26-28)	Field Name (29-34)	Field End (35-37)	Field Output Condition (38-40)	And Condition (41-43)	And Condition (44-46)	Zero Suppress (47)	Field Length (48-50)	Constant or Edit Control Word (51-75)	Card Number (78-80)
K																			Ø26					ØØ7	CHARGES	010
F																		WORDØ3	Ø49							020
F																		WORDØ4	Ø96							030
L	T	1	3	X						Ø1																040
K																			Ø29					Ø2Ø	CASHEDΔΔΔΔFORΔCHECKS	050
L	T	2	1	X			T	2	2	Ø3					F1	N77										060
F																		CHKTOT	Ø19					Ø1Ø	$ΔΔ,ΔΔØ.	070
F																		TOTCHG	Ø29					ØØ6	$ΔØ.	080
F																		DEPTOT	Ø49					Ø1Ø	$ΔΔ,ΔΔØ.	090
F																		FINBAL	Ø99					Ø1Ø	$ΔΔ,ΔΔØ.	100
F																		WORDØ6	1Ø2	Ø8						110
L	T	2	2		X																					120
F																		FINBAL	Ø14							130
F																		CHCKNO	ØØ7							140
F																		WORDØ8	Ø8Ø							150
W																		WORDØ1						Ø1Ø	CHECKS	160
W																		WORDØ2						Ø1Ø	CHARGE	170
W																		WORDØ3						Ø1Ø	DEPOSITS	180
W																		WORDØ4						ØØ7	BALANCE	190
W																		WORDØ5						ØØ1	)	200

Figure 5-10. Output Specifications, Sheet 4 of 5, Monthly Special Checking Account Report

OUTPUT SPECIFICATIONS

REPORT NAME MONTHLY SPECIAL CHECKING ACCOUNT REPORT PROGRAMMER ________ DATE ________ PAGE Ø9 OF Ø9

Format (1)	Line Type (2)	Line Level (3)	Line Number (4)	Output Print (5)	Output Punch (6)	Output Reserved (7)	Next Line Type (8)	Next Line Level (9)	Next Line Number (10)	Space Before (11-12)	Space After (13-14)	Skip Before (15-16)	Skip After (17-18)	Stacker (19)	Line Output Condition (20-22)	And Condition (23-25)	And Condition (26-28)	Field Name (29-34)	Field End (35-37)	Field Output Condition (38-40)	And Condition (41-43)	And Condition (44-46)	Zero Suppress (47)	Field Length (48-50)	Constant or Edit Control Word (51-75)	Card Number (78-80)
W																		WORDØ6						ØØ3	ΔOD	010
W																		WORDØ7						ØØ5	TOTAL	020
W																		WORDØ8						ØØ1	B	030

Figure 5-11. Output Specifications, Sheet 5 of 5, Monthly Special Checking Account Report

Sample Program Input Data File

A section of the input data file used to produce the sample report is shown in Figure 5-12. This illustration shows all the data cards required for checking account number 2116271; a similar set of input data cards is required for each checking account number in the file.

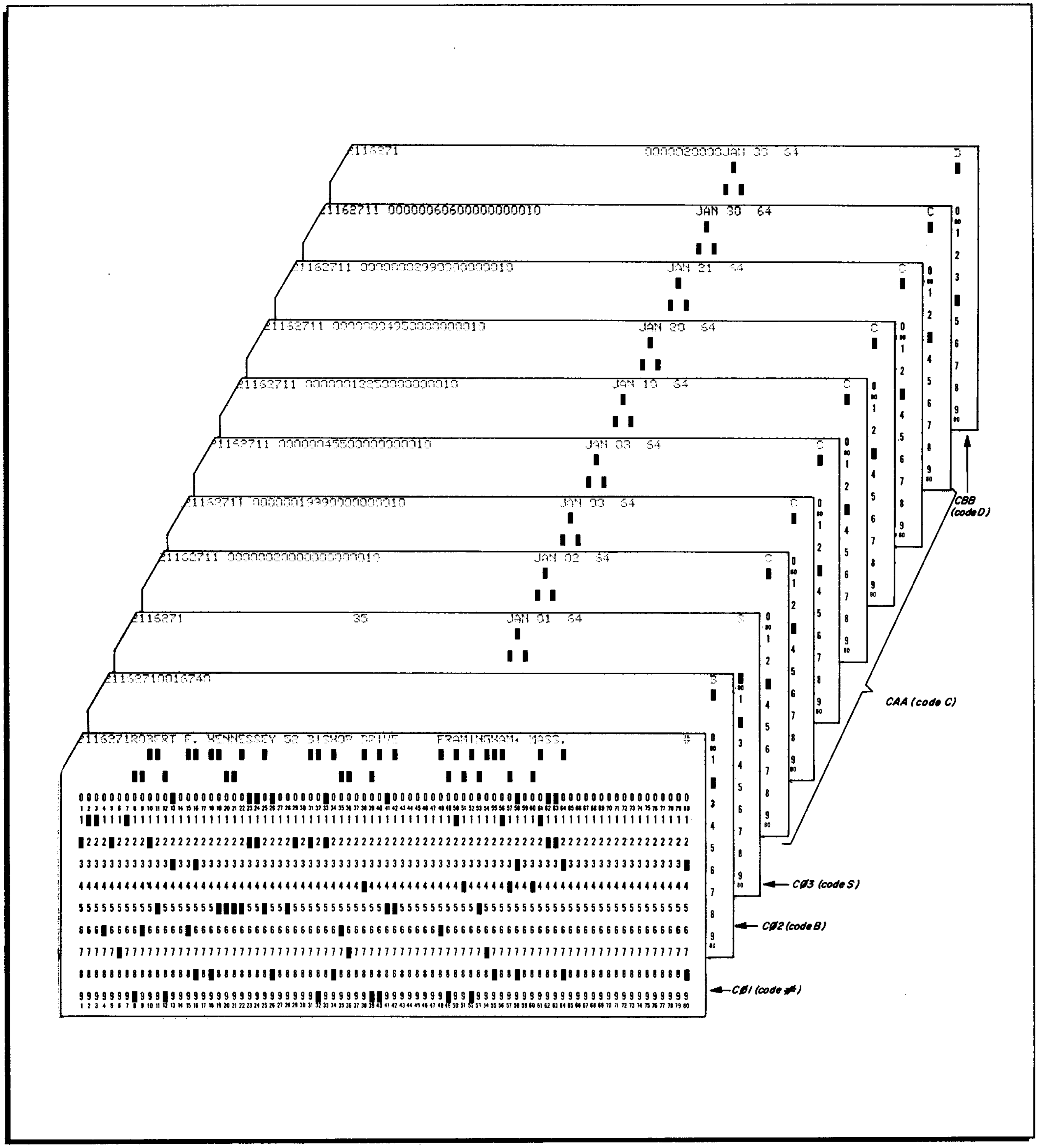

Figure 5-12. Section of Input Data File

MONTHLY SPECIAL CHECKING ACCOUNT REPORT 001

CHECKING ACCOUNT NUMBER 2116271

ROBERT F. HENNESSEY
52 BISHOP DRIVE
FRAMINGHAM, MASS.

CHECKS	CHARGE	SERVICE CHARGE	DEPOSITS	DATE	NUMBER OF CHECKS PAID	BALANCE
BALANCE BROUGHT FORWARD						$ 167.40
		$.35 SC		JAN 01 '64		$ 167.05
$ 20.00	$.10			JAN 02 '64	1	$ 146.95
$ 19.99	$.10			JAN 03 '64	2	$ 126.86
$ 45.50	$.10			JAN 03 '64	3	$ 81.26
$ 12.25	$.10			JAN 10 '64	4	$ 68.91
$ 4.95	$.10			JAN 20 '64	5	$ 63.86
$ 2.99	$.10			JAN 21 '64	6	$ 60.77
$ 60.60	$.10			JAN 30 '64	7	$ .07
			$ 200.00	JAN 30 '64		$ 200.07

TOTAL CHECKS CASHED	TOTAL CHARGES FOR CHECKS	TOTAL DEPOSITS	FINAL BALANCE
$ 166.28	$ .70	$ 200.00	$ 200.07

Figure 5-13. Printed Report, Page 1 of 3,
Monthly Special Checking Account Report

MONTHLY SPECIAL CHECKING ACCOUNT REPORT

CHECKING ACCOUNT NUMBER 2116272

JOHN F. SMITH
50 JONES AVENUE
E. OSTGOSH, PENN.

CHECKS	CHARGE	SERVICE CHARGE	DEPOSITS	DATE	NUMBER OF CHECKS PAID	BALANCE
BALANCE BROUGHT FORWARD						$ 166.40
		$.35 SC		JAN 01 '64		$ 166.05
$ 20.00	$.10			JAN 02 '64	1	$ 145.95
$ 19.99	$.10			JAN 03 '64	2	$ 125.86
$ 45.50	$.10			JAN 03 '64	3	$ 80.26
$ 12.25	$.10			JAN 10 '64	4	$ 67.91
$ 4.95	$.10			JAN 20 '64	5	$ 62.86
$ 2.99	$.10			JAN 21 '64	6	$ 59.77
$ 60.60	$.10			JAN 30 '64	7	$ .93 OD
			$ 200.00	JAN 30 '64		$ 199.07

TOTAL CHECKS CASHED	TOTAL CHARGES FOR CHECKS	TOTAL DEPOSITS	FINAL BALANCE
$ 166.28	$ .70	$ 200.00	$ 199.07

Figure 5-14. Printed Report, Page 2 of 3,
Monthly Special Checking Account Report

```
                    MONTHLY SPECIAL CHECKING ACCOUNT REPORT                 001

                     CHECKING ACCOUNT NUMBER          2116273


                   MARY BROWN
                   7 SMITH STREET
                   S. BOSTON, MASS.

CHECKS    CHARGE    SERVICE   DEPOSITS            DATE       NUMBER OF        BALANCE
                    CHARGE                                   CHECKS
                                                             PAID

BALANCE BROUGHT FORWARD                                                       $   167.40

                       $.35 SC                    JAN 01 '64                  $   167.05
$    20.00   $.10                                 JAN 02 '64      1           $   146.95
$    19.99   $.10                                 JAN 03 '64      2           $   126.86
$    45.50   $.10                                 JAN 03 '64      3           $    81.26
$    12.25   $.10                                 JAN 10 '64      4           $    68.91
$     4.95   $.10                                 JAN 20 '64      5           $    63.86
$     2.99   $.10                                 JAN 21 '64      6           $    60.77
$   160.60   $.10                                 JAN 30 '64      7           $    99.93 OD
                              $   100.00          JAN 30 '64                  $      .07
                              $   100.00          JAN 30 '64                  $   100.07

TOTAL     TOTAL               TOTAL                                           FINAL
CHECKS    CHARGES             DEPOSITS                                        BALANCE
CASHED    FOR CHECKS

$   266.28   $ .70            $   200.00                                      $   100.07
```

Figure 5-15. Printed Report, Page 3 of 3, Monthly Special Checking Account Report

SECTION VI
OPERATING PROCEDURES

GENERATING THE SYMBOLIC PROGRAM

Section II described the process of portraying the desired report format and formulating the input, data, calculation, and output specifications used in producing the printed report. The desired report is laid out on the Honeywell Report Layout chart exactly as it will be printed, as the first step. The report specifications are then written, using the report presentation depicted on the Report Layout chart, and governed by the input data being used and the calculations which must be performed.

After writing of the specifications is completed, the information contained on these sheets is punched into cards. One card is punched for every line entry on each of the specifications sheets used. These cards will compose the specifications source deck used in producing the object program (whose makeup constitutes the second step in using the Honeywell Series 200 Report Generator). This section describes the procedures involved in producing and executing the Series 200 RG object program.

Format of Control Card Specifications

In order to complete the specifications source deck, the programmer must also prepare a control card, since this is required by the Series 200 Report Generator. The same control card can be used for a variety of reports, provided that the same routine will be followed in generating each report. The format of the Control card is presented in Table 6-1, together with an explanation of the entries, column by column.

Table 6-1. Format of Control Card Specifications

Parameter	Columns	Contents	Explanation
CARD ID	1 - 4	CNTL	Identifies this card as the Control card.
MEMORY SIZE	5	One decimal digit, from 1 to 8	Specifies the number of memory banks available in the Series 200 system which is being used for the program. 1 = 4,096 characters of memory; 2 = 8,192 characters of memory, etc.
PRINTED OUTPUT	6 - 8	Blank, or non-blank (any valid character other than a blank)	Blank - implies no printed output. Non-blank - output will be printed.
PERIPHERAL ADDRESSES	9 - 14	See below	These columns are used to designate the addresses assigned to the card reader, card punch, and printer. If these columns are left blank, the RG will use the recommended Series 200 address assignments.

Table 6-1 (cont). Format of Control Card Specifications

Parameter	Columns	Contents	Explanation
Card reader	9-10	40 - 47_8 or blank	Address assigned to the object-program's card reader. If left blank, the RG will use the Series 200 recommended trunk number, 41_8.
Card punch	11 - 12	00 - 07_8 or blank	Address assigned to the object-program's card punch. If left blank, the RG will use the Series 200 recommended trunk number, 01_8.
Printer	13 - 14	00 - 07_8 or blank	Address assigned to the object-program printer. If left blank, the RG will use the Series 200 recommended trunk number, 02_8.
DATA INPUT/ OUTPUT	15	0, 1, 2 or 3	0 - data input is on cards, i.e., no tapes are used. 1 - data input is on tape; 2 - tape is used for output only; 3 - tape is used both for input and for output.
Common control unit	16	1 or blank	1 - the generated program will use an integrated control unit. Blank - the generated program will not use an integrated control unit.
TAPE DESCRIPTION	17 - 21	See below	These columns are used to describe the peripheral address assignments, tape drive, and parity (i.e., odd or even) of the input data tape.
I/O trunk	17 - 18	00 - 07_8, 40 - 47_8, or blank	Address to the input data tape; blank- 40_8 is assumed if input tape is used.
Tape drive	19	0 - 7, or blank	Tape drive on which input data tape is mounted; blank if no input tape is used.
Format of tape output	20	1 or blank	1 - output tape contains spacing control characters and lines (133-character record). Blank - output tape without spacing control characters and lines (132-character record). If column 20 is left blank and a line card in the Output Specifications (L in column 1) requests tape output only (X in column 7) then columns 11, 12, and 15 through 18, inclusive, in the line card must be blank and columns 13 and 14 must contain a 0.
Parity	21	6, 7, or blank	6 - input data tape has odd parity; 7 - input data tape has even parity. If column 21 is left blank, the RG assumes odd parity.

Table 6-1 (cont). Format of Control Card Specifications

Parameter	Columns	Contents	Explanation
Banner Character	22	0, or 1, or blank	This column specifies whether or not the input tape has a banner character. 0 - input tape without banner character. 1 - input tape with banner character. Blank - no input tape.
OPTIONS	23 - 27	See below	These columns are used to specify the options available for use by the RG.
Stacker select	23	1 or blank	1 - Stacker-select option available; Blank - no stacker-select option[1]
OUTPUT MEDIA	24	Blank or 1	Blank - no card output (tape or printed output); 1 - punched card output.
Multiply/divide	25	M or blank	M - Multiply/divide hardware available; Blank - No multiply/divide hardware.
------	26 - 27	Blank	Not used.
Tape label	28	0 or 1, or blank	0 - input tape without header label; 1 - input tape with header label. Blank - no tape input.
Item size	29 - 32	0001 - 1000, VVVV, or blank	Designates item size of tape records: 0001 1000 - actual number of characters per item; VVVV - V's are inserted in these columns to denote variable-length records; Blank - no tape input
Blocking factor	33 - 34	02-99 or blank	Designates the tape blocking factor: 02-99 - actual number of items per record; Blank - not blocked, or no tape input.
Data files	35	1-9 or blank	Specifies the number of data files; if blank, one is assumed.
Program generation	36[2]	1 or blank	1 - generate the program on tape 1, then punch the program from the tape. Blank - generate the program directly. NOTE: If column 36 contains a 1, column 37 must not be left blank.
RG output	37[2]	0-7 or blank	0-7 - tape control for tape on which Report Generator B will write its output in card-image form and from which the output will subsequently be punched on cards.

Table 6-1 (cont). Format of Control Card Specifications

Parameter	Columns	Contents	Explanation
RG output (cont)			Blank - Report Generator output will be produced directly on cards (i.e., not first generated on tape and then punched on cards from tape).
------	38 - 74	Blank	Not used.
OBJECT PROGRAM IDENTIFICATION	75 - 80	Identification	Object-program identification name (left-justified), if it is to be punched in symbolic object program deck.

[1] If no stacker-select option is available, the object-program cards must be sorted on the basis of a 1 in column 73.

[2] These parameters are used only by Report Generator B. There is no effect on the generated program when Report Generator is told to generate the program on tape and punch from the tape. The only effect is the way in which the program is generated. Instead of punching the program as it is generated from each specification card, the punched card images are written onto logical tape 1. Once the complete program has been generated on tape, it is punched by Format (the last phase of Report Generator B). The tape is passed over twice: during the first pass, the PROG card, ORG card, and all instructions are punched; during the second pass all constants, work areas, and the CLEAR and END cards are punched. Therefore, the generated program is automatically punched in the correct order, eliminating the need for a mechanical sort or the stacker-select option.

Format of Carriage Control Card Specifications

If a 1401 RPG program which is being converted for use as a Series 200 Report Generator program uses a carriage-control paper tape to control line spacing, a Carriage Control card must be prepared to simulate the function of the paper tape. This requirement holds true when using all Honeywell high-speed printers which do not control line spacing by means of a carriage-control paper tape. The Carriage Control card is prepared according to the format shown in Table 6-2.

Table 6-2. Format of Carriage Control Card

Parameter	Columns	Contents	Explanation
LINES TO BE PRINTED	1 - 3	Three-digit decimal number	Specifies the total number of lines of the page to be printed. (Number of lines per inch times length of print area from top to bottom of form.) If the entry is less than three digits, it is padded with leading zeros.
CHANNEL-12 OVERFLOW	4 - 6	Three-digit decimal number	Indicates the line position of channel-12 overflow. If the entry is less than three digits, it is padded with leading zeros.
------	7 - 9	Blank	Not used.

Table 6-2 (cont). Format of Carriage Control Card

Parameter	Columns	Contents	Explanation
CARRIAGE SKIP POSITIONS	10 - 12	See below	These columns specify the carriage skip positions.
First channel designation	10	1 - 9, B, C, or D	Used to indicate the first channel: 1-9 for channels 1-9, respectively B for channel 10, C for channel 11, D for channel 12.
First channel	11 - 12	Two-digit decimal number	Used to specify line position in relation to top of form. A leading zero must be used for line numbers less than 10. For lines numbers greater than 99, a 12-zone punch (R) is used in the second character position of the entry.
ADDITIONAL CARRIAGE SKIP POSITIONS	13 - 45	See below	Used to define additional carriage skip positions.
Channel designation	13	Same as columns 10	Columns 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, and 43 are used to designate channels 2-12, respectively, when all 12 channels are used.
Line position	14 - 15	Same as columns 11-12	Columns 14-15, 17-18, 20-21, 23-24, 26-27, 29-30, 32-33, 35-36, 38-39, 41-42, and 44-45 are used to designate channels 2-12 line positions, respectively, when all 12 channels are used. NOTE: Channel 1-skip positions defined first, followed by channel 2, and so on through channel 12. Entries are made beginning with columns 10 through 12. No blanks can occur until the last entry has been made. Unused positions within this card field are left blank. While carriage-skip positions are often punched twice on the same tape to provide a tape loop convenient for threading, skip position entries must not be duplicated on the Carriage Control card. Channel 1 must be specified here if no overflow lines and no skipping to a channel is specified in the output specifications.
------	46 - 71	Blank	Not used.
CARD IDENTI-	72 - 80	CARRIAGET	Identifies this card as a Carriage Control card.

THE ASSEMBLY PROGRAM

To obtain the object program from the Series 200 Report Generator using the various report specifications, the programmer must first prepare a specifications source deck on punched cards. This source deck, therefore, represents the specifications of the particular report.

The Series 200 RG specifications source deck is composed of the input, data, calculation, and output specifications described in Section I and the CARRIAGET and control cards described in this section. One 80-column card is punched for each line entry on each of the specifications sheets. The cards are then arranged in the source deck in the following order:

1. Control card;
2. Input specifications cards;
3. Data specifications cards;
4. Calculation specifications cards;
5. Output specifications cards, and
6. Carriage control card.

Report Generator A generates a symbolic card deck which is then assembled using Easycoder Assembler A. Report Generator B generates a symbolic card deck which is then assembled using either Easycoder Assembler A or B.

The Report Generator B program is composed of seven symbolic program decks as follows:

1. CONTRL
2. DUM3CH
3. INPUTΔ
4. DATAΔΔ
5. CALCΔΔ
6. MULDV3
7. FORMAT

With the exception of the DUM3CH and MULDV3 decks, each symbolic deck must be assembled using Easycoder Assembler A or B in order to obtain condensed decks.

The Report Generator program is composed of seven symbolic program decks as follows:

1. CONT1Δ
2. DUM2CH
3. INPΔΔΔ
4. DAΔΔΔΔ
5. CAΔΔΔΔ

6. MULDV2
7. FORΔΔΔ

With the exception of the DUM2CH and MULDV2 decks, each symbolic deck must be assembled using Easycoder Assembler A in order to obtain condensed decks.

Input Deck with RG on Cards

Definition: Asterisk Card

Card with an * (X, 8, 4) in column 1 followed by any comment in columns 2-80. The recommended comment to be punched is shown within the brackets below.

The format of the input deck for Report Generator B is as follows (see Figure 6-1):

1. CONTRL condensed deck.
2. Asterisk Card (PLACE CONTROL CARD AFTER THIS CARD.)
3. Control Card.
4. Asterisk Card (PLACE CONTROL CARD BEFORE THIS CARD.)
5. DUM3CH symbolic deck with PROG and END card removed.
6. INPUTΔ condensed deck.
7. Asterisk Card (PLACE INPUT SPECS AFTER THIS CARD.)
8. Input specification cards.
9. Asterisk Card (PLACE INPUT SPECS BEFORE THIS CARD.)
10. DATAΔΔ condensed deck.
11. Asterisk Card (PLACE DATA SPECS AFTER THIS CARD.)
12. Data specification cards.
13. Asterisk Card (PLACE DATA SPECS BEFORE THIS CARD.)
14. CALCΔΔcondensed deck.
15. Asterisk Card (PLACE CALC SPECS AFTER THIS CARD.)
16. Calculation specification cards.
17. Asterisk Card (PLACE CALC SPECS BEFORE THIS CARD.)
18. MULDV3 symbolic deck with PROG and END card removed.
19. FORMAT condensed deck - Part 1. Part 1 is the first part of the FORMAT condensed deck down to and including the EXECUTE card. This card is identified as card number 40000 in columns 76 to 80.
20. Asterisk Card (PLACE FORMAT SPECS AFTER THIS CARD.)
21. FORMAT specification cards.
22. Asterisk Card (PLACE FORMAT SPECS BEFORE THIS CARD.)
23. FORMAT condensed deck - Part 2. This is the rest of the FORMAT condensed deck.
24. Asterisk Card (PLACE CARRIAGE CONTROL CARD AFTER THIS CARD.
25. Carriage Control Card - 3 blank cards

The format of the input deck for Report Generator A is as follows:

1. CONT1Δ condensed deck.
2. Same as for Report Generator B.
3. Same as for Report Generator B.
4. Same as for Report Generator B.
5. DUM2CHΔ symbolic deck with PROG and END card removed.
6. INPΔΔΔ condensed deck.
7. Same as for Report Generator B.
8. Same as for Report Generator B.
9. Same as for Report Generator B.
10. DAΔΔΔΔ condensed deck.
11. Same as for Report Generator B.
12. Same as for Report Generator B.
13. Same as for Report Generator B.
14. CAΔΔΔΔ condensed deck.
15. Same as for Report Generator B.
16. Same as for Report Generator B.
17. Same as for Report Generator B.
18. MULDV2 symbolic deck with PROG and END card removed.
19. FORΔΔΔ condensed deck - Part 1. Part 1 is the first part of the FORΔΔΔ condensed deck down to and including the EXECUTE card. This card is identified as card number 09860 in columns 76 to 80.
20. Same as for Report Generator B.
21. Same as for Report Generator B.
22. Same as for Report Generator B.
23. FORΔΔΔ condensed deck - Part 2. This is the rest of the FORΔΔΔ condensed deck.
24. Same as for Report Generator B.
25. Same as for Report Generator B.
26. Same as for Report Generator B.

These cards, when arranged in the Report Generator master deck as shown in Figure 6-1, are the input data which will be processed by a Series 200 system to produce a generated symbolic program deck. The user may then assemble the object program from the generated symbolic program deck using Easycoder Assembler A (or Easycoder Assembler B, if desired, for Report Generator B). This operation requires two machine passes to obtain a finished object program.

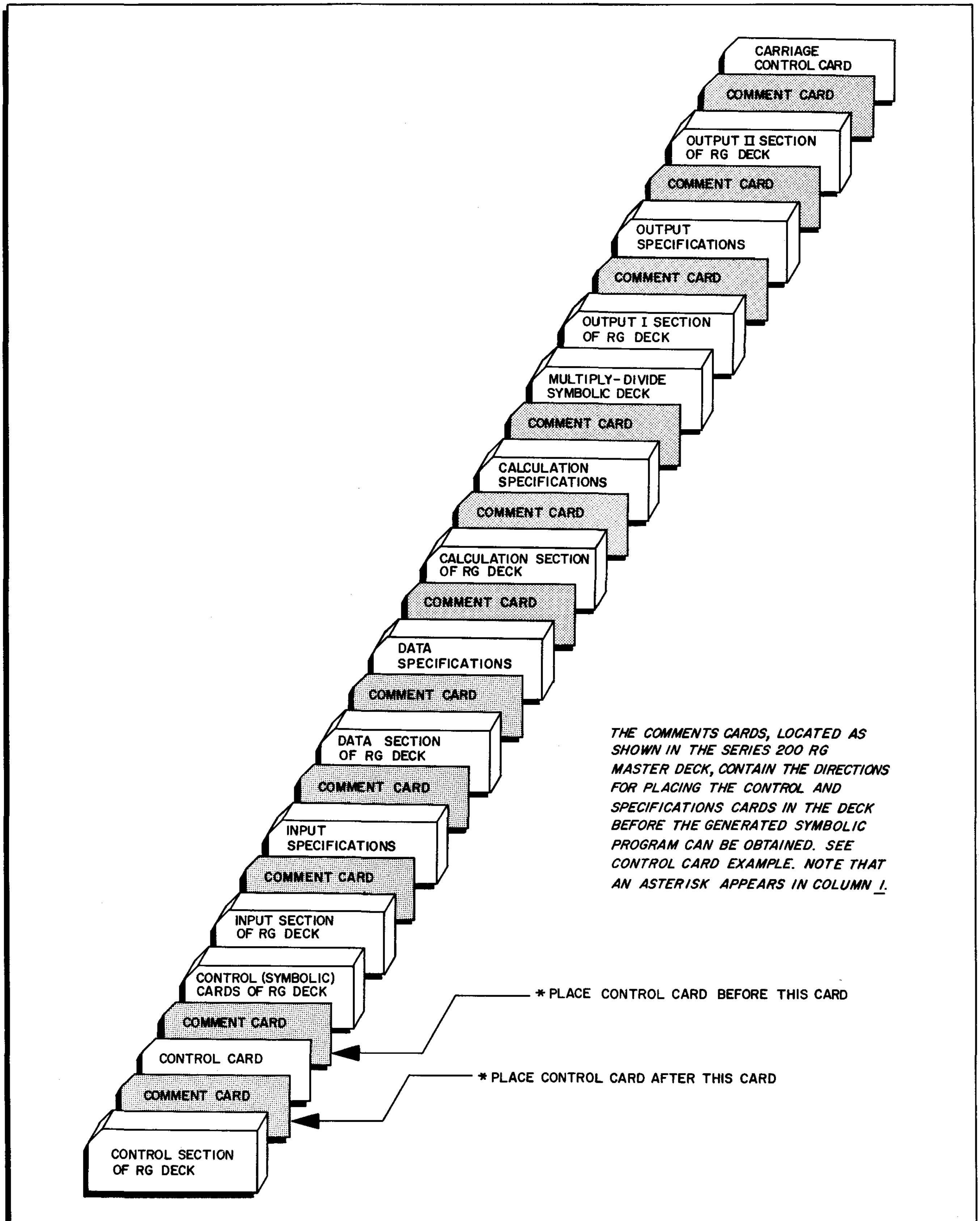

Figure 6-1. Sample Deck for Series 200 Report Generator Run

The assembly program accepts the generated symbolic program cards and automatically produces a corresponding machine-language program which is the object program itself. During assembly, mnemonic codes are converted into machine-language codes, absolute storage addresses are assigned to instructions and symbolic operand references, and the final program is completely assembled. A secondary output of the assembly program is a complete, printed summary of the symbolic program and the corresponding machine-language entries.

Should the programmer find it necessary to alter the program after it has been assembled, he can isolate the affected areas in the symbolic program, enter the required changes, and then use the assembly program to reassemble a corrected version of the object program.

During the first machine pass, which generates the symbolic program deck for input to Easycoder Assembler A, the Type 227 Card Reader/Punch equipped with the stacker-select option separates the cards as follows (see Figure 6-2):

1. The Series 200 Report Generator master deck plus the specifications source deck are selected into the 1 stacker pocket.
2. The generated symbolic program is selected into two stacker pockets; the first part falls in the 4 stacker, and the second part in the 8/2 stacker. The END card is run out by the operator and falls in the NP stacker pocket. The generated symbolic program cards are removed from the stackers for assembly in the order: 4 stacker first, 8/2 stacker second, followed by the END card from the NP pocket.

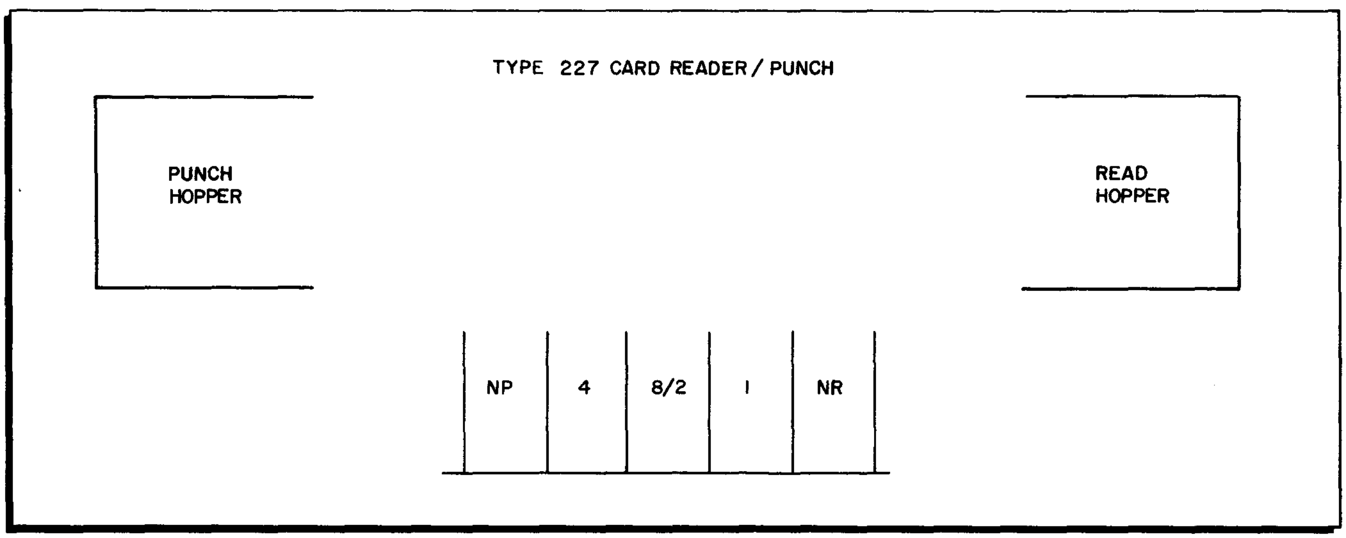

Figure 6-2. Type 227 Card Reader/Punch Stacker Pocket Array

When the stacker-select option is not incorporated in the reader/punch, the object program card deck must be sorted on the basis of a 1 punch in column 73. The cards with a 1 punch in column 73 go at the end of the deck, followed by the END card.

Insertion of Own-Coding and Subroutines

The programmer has the option of inserting own-coding and required subroutines, such as table-lookup or square-root operations, in the generated program. To execute own-code routines, the programmer inserts these routines in the generated symbolic deck (once it has been generated). An instruction must be inserted at the entry point (ENTRY 1, ENTRY 2, or ENTRY 3) which will cause a branch to the own-code routine. In order to return to the correct point in the program, the own-code routine should begin with an SCR (Store Control Registers) instruction, which stores the B-address register, and should end with a B (Branch) to that location.

These instructions and all instructions in the own-code routine may be added to the generated program either before or after assembly. The programmer may write the routine in Series 200 assembly language and then assemble it. The entry points at which own-coding and subroutines may be inserted are labeled ENTRY 1, ENTRY 2, and ENTRY 3.

All labels assigned in own-code routines must be entirely alphabetic and cannot duplicate any field names used on the various specifications sheets. Because all labels produced by the Report Generator are field names from the specifications sheets or are numerical in at least one position of the name, processing errors are likely to occur if the above rule for own-code labels is not observed.

ERROR INDICATIONS AND ERROR CORRECTION

There are three categories of halts:

1. The programmed, or normal, halts encountered during Easycoder Assembly;
2. Peripheral stalls;
3. Machine malfunctions.

A halt has occurred when the lights on the operator's control panel stop blinking and there is no action on the peripheral devices. When a halt occurs, the instructions for programmed halts should be performed first; however, if the A and B addresses do not contain coding which identifies the halt condition, check the operator's control panel and the appropriate peripheral-device control panels for abnormal or malfunction indications. Refer to the Honeywell Series 200 Equipment Operator's Manual (Model 200), Order No. 040.

The programmed halts which may be encountered during the Easycoder Assembly are discussed in the Easycoder 4K Operating Procedures, (DSI-243C).

While the Series 200 Report Generator object program is being generated, some instances of incorrect coding and certain oversights in preparing the source deck from the specifications will result in machine halts. A list of these error conditions is shown in Table 6-3 below.

Table 6-3. Series 200 Report Generator Error Indications

Specification Type	Error Condition Causing Program Halt
Control Card	No control card present in the Report Generator deck.
Input	No C or S contained in column 1 (one or the other must be present in every line entry of the specifications sheet).
(cont)	No resulting condition specified (columns 42-43 blank when the record code entry describes one input-record type).
Data	Not a data card, or a data card with no D entered in column 1. Undefined field source (i.e., not specified in input specifications). Illegal operation code specified in column 30. Incorrect status (must be B, Z, N, or P). Numeric Conversion cannot be handled. Optional records cannot be processed.
Calculation	Not a calculation card. Incorrect status (must be B, Z, N, P, E, U, L, or H). Total/Detail column (column 49) does not contain either a T or a D.
Output	First Output-Specifications sheet entry must be a line specification (L in column 1). No line designation specified (H, D or T). Incorrect designation in column 1 (must be L, F, B, K, or W). Incorrect card order. Printed output must be specified (X in column 5).
Carriage Control	No CARRIAGET card.

The error conditions presented in Table 6-3, above, show the more common errors and/or oversights likely to be encountered in preparing the specifications. Careful checking of the specifications before the symbolic program is generated will help eliminate program halts due to these errors. Should one of these errors be present, it will then be necessary to correct the error before continuing with generation of the symbolic program.

ASSEMBLY OPERATING PROCEDURES

The assembly operating procedures for the Series 200 Report Generator are the same as those presented in the Easycoder 4K Operating Procedures, DSI-243C.

EXECUTION OF THE REPORT GENERATOR OBJECT PROGRAM

When the object program has been assembled as described in the preceding pages, the program can then be executed. The object program, together with the file of input data cards containing the specific information that will appear in the report, is passed through a Series 200 computer. The output of this operation is a printed report in the desired format, containing all the information specified by the programmer.

SERIES 200 RG OPERATING PROCEDURES

The procedures necessary for executing the Report Generator object program are specified in the Easycoder 4K Operating Procedures.

The Series 200 Report Generator system (i.e., the linked deck), consists of five assembled decks, the last of which is executed in two parts (Output Sections I and II). The Series 200 RG master deck sections are coded in machine language. In addition to the machine language decks, there are decks in Easycoder symbolic language (e.g., Symbolic Control Cards of the RG deck in Figure 6-1), which are read, processed according to the specification cards, and utilized in the generated program. The Series 200 RG master deck also contains comments cards interspersed through it. The comments cards contain an asterisk in card column 1 and state where the specifications cards for each section should be inserted. After the user's specification cards for all the sections have been keypunched and visually checked for clerical errors, they should be placed in the linked Series 200 RG master deck at the positions indicated by comments cards. The CARRIAGET card is physically the last card of the specifications source deck. It is followed by three blank cards when the deck is processed.

Operating Procedures for Report Generator A on Cards

Series 200 Report Generator A operates in the two-character address mode and uses the card reader and card punch. The Bootstrap card for the first section of the Series 200 RG master deck is manually entered into location 7257 (octal). The Bootstrap card for the following section is read in when all the specification cards of the first section have been processed. No SENSE switch settings are necessary unless non-standard peripheral address assignments are specified for the card reader/punch. If the card reader/punch address assignments are not 41 for the reader and 01 for the punch, the operator must set SENSE switch 1 (S1) to ON for load and halt. The peripheral address of the reader must be manually entered into octal address 7777 and the address assigned to the punch manually entered into octal address 7776. After the address assignments for the card reader and card punch have been entered, the operator must set SENSE switch 1 to OFF and press the RUN button.

The input data file should be followed by a termination card punched 1ERI in the first five positions, and an 8, 2 punch in the sixth column. The normal end-of-run halt is identified by zero and octal 4677 in the A- and B-address registers, respectively.

Operating Procedures for Report Generator B on Cards

With the input deck for Report Generator B arranged in the specified order, the following steps must be performed to generate a symbolic program for input to Easycoder Assembler A or B.

1. If Report Generator B is to be run on a computer equipped with a combination reader/punch, mount a work tape on logical tape drive 1 and place it in PERMIT status. Rewind the work tape.
2. Press the STOP button on the operator's control panel.
3. Press the INITIALIZE button on the operator's control panel.
4. Energize the card readerand card punch; place the correctly ordered input deck in the card reader input hopper.
5. Set SENSE switch 1 ON for load and halt only if the card reader/punch address assignments are not 41_8 for the reader and 01_8 for the punch. If the assignments are different, the operator must set SENSE switch 1 (S1) to ON. A halt will occur during the run (B-address register = 4000), at which time the reader and the punch address assignments must be entered manually into octal locations 17777 and 17776, respectively.
6. The Bootstrap card for the first section of the Report Generator B master deck is manually entered into octal location 17257.
7. Press the RUN button on the operator's control panel. The normal end-of-run halt is identified by zero and octal 14677 in the A- and B-address registers, respectively.

Machine Error Halts

In the event of a hardware failure when using Report Generator A or B, use the standard procedures for error halt interpretation described in the Easycoder 4K Operating Procedures. When the operator displays the contents of the A- and B-address registers following an error halt in Report Generator A, the more common hardware failures are designated by the register contents:

AAR	BAR (octal)	Meaning
0000	0110	Card reader hole-count error or illegal punch.
0000	0120	Card punch error.
0000	0150	Card reader inoperable.
0000	0160	Card punch inoperable.

NOTE: The B-address register contents when standard peripheral address assignments are used. The first bit of the B-address register is zero; the next five bits designate the peripheral address of the error device and the last six bits identify the type of error.

When the operator displays the contents of the A- and B-address registers following an error halt in Report Generator B, the more common hardware failures are designated by the register contents:

AAR	BAR[1] (octal)	Meaning
10000	0110	Card reader hole-count error or illegal punch.
10000	0120	Card punch error.[2]
10000	0150	Card reader inoperable.
10000	0160	Card punch inoperable.
10000	0001	Write error on logical tape 1.[3]
10000	0041	Read error on logical tape 1.[3]

NOTES: [1]The B-address register contents are as shown when the standard peripheral address assignments are used. The first bit of the B-address register is zero; the next five bits designate the peripheral address of the error device; and the last six bits identify the type of error. If the machine error halt can be corrected, the operator need only press the RUN button to continue.

[2]If a punch error occurs when punching the card image from tape, pressing RUN after the punch error halt will cause the last two cards to be repunched.

[3]There are 64 attempts made to write a record on logical tape 1 before an uncorrectable write error halt occurs. When reading from this tape, 64 attempts are made before an uncorrectable read error occurs, and all noise records are bypassed.

If the machine error halt can be corrected, the operator need only depress the RUN button to continue. (Refer to Section V of the Honeywell Series 200 Equipment Operator's Manual (Model 200), Order No. 040).

RG Specifications Error Halts

When the program halts due to an error in the Series 200 RG Specifications cards, the type of error can be identified from the tabulation of error halts in Table 6-4. The table shows the contents of the B-address register in octal (contents of the A-address register will be zero in each case), the type of error, and whether or not a restart from the error halt point is permissible.

In order to continue with program execution, the operator must first identify the type of error halt from the table and also determine from the table if a restart is permissible. If the card can be corrected and reread or should be skipped, the generated symbolic program must be very carefully checked for possible duplicate instructions. Of course, duplication of symbol definition statements will cause assembly errors; since these are of no consequence, they may be left in to avoid unnecessary handling of the unsequenced deck. After the error halt has been cleared and the cards are once again in their proper order in the card reader, the operator can depress the RUN button to continue the program.

Table 6-4. Coded Halts for Series 200 RG Specifications

Contents of BAR (octal)		Type of Error	Restart Permissible	Operator Action
Report Generator A	Report Generator B			
4100	14100	No Control Card	Yes	Insert Control Card, refeed cards, and press RUN.
4201	14201	Not an Input Specification card	Yes	Remove card and press RUN to continue processing.
4300	14300	A or B field is blank	No	Rerun program after correcting error card.
4301	14301	Not a Data Specification card	Yes	Remove card and press RUN to continue processing.
4302	14302	Illegal status code in column 11 of a Data Specification card	Yes	Correct card, refeed, and press RUN to continue.
4303	14303	Illegal operation code in column 30 of a Data Specification card	Yes	Correct card, refeed, and press RUN to continue.
4304	14304	Undefined source in columns 20-22 of a Data Specification card	Yes	Correct card, refeed, and press RUN to continue.
4401	14401	Not a Calculation Specification card	Yes	Remove card and press RUN to continue processing.
4403	14403	Illegal status code in column 11 of a Calculation Specification card	Yes	Insert corrected card, refeed cards and press RUN to continue processing.
4500	14500	First Output Specification entry not a line specification	No	Correct specifications, insert card and rerun program.
4501	14501	Illegal line code in column 2 of an Output Specification card	No	Insert information in proper place and rerun program.
4502	14502	Not the next line specificied on line specification	No	Place specifications cards in correct order and rerun program.
4503	14503	No output medium specified on line specification	No	Correct Output Specification card and rerun program.
4600	14600	No CARRIAGET card	Yes	Insert CARRIAGET card and press RUN to continue processing.
4601	14601	Illegal channel specified in CARRIGET card	Yes	Correct card, refeed, and press RUN to continue.

Table 6-5. RG Generated Program Error Halts

Contents of B-address Register	Meaning	Operator Action
0CU00	Device inoperable	Activate the device indicated by the CU number (41 for card reader, 02 for printer, etc.) and depress RUN to continue.
0CU1X	Tape read or card hole-count error	A tape read error after trying 20 times may be ignored by pressing RUN to continue. X = the drive number where the tape read error has occurred. A card hole-count error is rectified by correcting the card, refeeding it, and pressing RUN. X will be 1 for a card hole-count error. CU = peripheral address of error device.

Table 6-5 (cont). RG Generated Program Error Halts

Contents of B-address Register	Meaning	Operator Action
0CU24	Tape write error	Depress RUN to backspace, erase, and try to write 40 more times. CU = peripheral address of tape control unit.
0CU34 (Report Generator A); or 0CU32 (Report Generator B); or	Physical end of print tape	Report Generator A: Mount another tape on logical drive 4, rewind, and press RUN. Report Generator B: Mount another tape on logical drive 2, rewind, CU = peripheral address of tape control unit.
0CU20	Card punch error	There is no correction process for punch errors in the generated program. The program must be rerun after correction of the error card. This halt is used in both the RG and the generated programs. CU = peripheral address of punch control unit.
0CU4X	End of Input file	Set or reset SENSE switches 1 and 2 to control file processing of the next file, i.e., with SENSE switches OFF, depress RUN to process next file; set SENSE switch 1 ON to bypass next file. See Note below. X = drive number. CU = peripheral address of error device.
7004	The settings of SENSE switches 1 through 4 are recorded.	Set or reset SENSE switches 1 and 2 to control file processing of first fiel. See Note below.
4000	A necessary record in sequence is missing, or an undefined record type is present.	Depressing RUN will cause the program to check the record against the next sequence type (therefore, if an undefined record type is present, no recovery is possible).
4010	An undefined record type is present. (This halt, rather than the 4000 halt, occurs only if no numeric records are defined.)	Depress RUN to bypass the undefined record.
4777	End of run.	------

NOTE: SENSE switches 1, 2, 3 and 4 may be used to control processing conditions in the generated program. If the generated program accepts tape input, a halt with the contents of the B-address register equal to 7004 occurs after the settings of SENSE switches 1 through 4 are recorded by the program. After this halt, the user may set SENSE switches 1 and 2 to control tape input conditions as follows:

> SENSE switch 1 ON is used to bypass the next input file at the time that any 0CU4X halt occurs. SENSE switch 2 is normally only set ON by the user for use in conjunction with SENSE switch 1 (also ON) to bypass read error halts when an input file is also being bypassed. All read error halts may also be bypassed by setting SENSE switch 2 ON at the 7004 halt or at any 0CU4X halt before processing an input file. However, since erroneous information is accepted as valid information in such instances, the results are unspecified.

Input Deck with RG on Tape

Definition: Asterisk Card

Card with an * (X-, 8-, 4-punch) in column 1 followed by any comment in columns 2-80. The recommended comment to be punched is shown within the brackets below.

The format of the input deck for Report Generator A and B is as follows:

1. Asterisk Card (PLACE CONTROL CARD AFTER THIS CARD.)
2. Control Card.
3. Asterisk Card (PLACE CONTROL CARD BEFORE THIS CARD.)
4. Asterisk Card (PLACE INPUT SPECS AFTER THIS CARD.)
5. Input Specification Cards.
6. Asterisk Card (PLACE INPUT SPECS BEFORE THIS CARD.)
7. Asterisk Card (PLACE DATA SPECS AFTER THIS CARD.)
8. Data Specification Cards.
9. Asterisk Card (PLACE DATA SPECS BEFORE THIS CARD.)
10. Asterisk Card (PLACE CALC SPECS AFTER THIS CARD.)
11. Calculation Specification Cards.
12. Asterisk Card (PLACE CALC SPECS BEFORE THIS CARD.)
13. Asterisk Card (PLACE FORMAT SPECS AFTER THIS CARD.)
14. FORMAT (Output) Specification Cards.
15. Asterisk Card (PLACE FORMAT SPECS BEFORE THIS CARD.)
16. Asterisk Card (PLACE CARRIAGE CONTROL CARD AFTER THIS CARD.)
17. Carriage Control Card.
18. 3 Blank Cards.

Placing the RG Program on Tape

The Series 200 RG master deck exists as a deck of punched cards. The master deck is placed on tape using Update A, a Honeywell-supplied card deck which is used to create and maintain a file of programs on a self-loading tape (SLT). A detailed discussion of the Update A program is presented in the Honeywell publication, Easycoder 4K Operating Procedures, DSI-243C.

The Report Generator B (or Report Generator A) program can be placed on any existing SLT and in any desired position on that tape. The RG programs, however, must be in the exact sequential order shown in Figure 6-3. Report Generator B can then be executed by loading the Control program, CONTRL, by using the Search routine (the Control program for Report Generator A is CONT1). The execution of Report Generator B does not destroy Search. Each of the seven RG programs can be updated separately in the same manner as any other program on the SLT by using Update A.

Update A is a program for creating or updating a self-loading tape (SLT). The process is regulated by a control deck which contains one action director card for each program. Any combination of the four action directors (viz., INSert, REPlace, COPy, and DELete) may be used when updating an existing SLT; however, only the INS director may be used when creating a new SLT. When updating an existing SLT, each program on the old SLT must be represented by an action director which is placed in the control deck in the same order as the corresponding programs appear on the old SLT.

Inputs to Update A include:

1. Update A deck or Update A recorded on an existing SLT;
2. Control deck (always on cards); and
3. Input program(s) on cards or on tape (self-loading, single-instruction or condensed programs).

Outputs of Update A are a new SLT and an SLT directory (optional).

The source of program input is controlled by SENSE switch 2, and the printing of an SLT directory is controlled by SENSE switch 4, as specified below:

SENSE switch 2 - ON = Input programs on cards
OFF = Input programs on tape two

SENSE switch 4 - OFF = SLT directory listing
ON = No directory listing.

To operate the Update A program, perform the following steps:

1. Depress the STOP button on the operator's control panel.
2. Depress the INITIALIZE button on the operator's control panel.

3. Set SENSE switches 2 (input) and 4 (SLT directory) for desired operation according to the SENSE switch settings listed above.

4. Energize and load the card reader, the new SLT tape unit (tape one), and other required peripheral devices which correspond with SENSE switch selections of step 3, above.

5. Place the Update A deck in the card reader input hopper or mount the existing SLT on tape 0, as applicable.

6. Place the complete control deck, or the control deck intermixed with the program decks, behind the update deck in the card reader input hopper, as applicable.

7. Perform the following steps at the operator's control panel:

   a. Set the ADDRESS buttons to the bootstrap address of the Update A program (i.e., 0001 for the Update program on cards, or 0000 for the Update program on an existing SLT).

   b. Set the CONTENTS buttons to designate the card reader (if the Update A program is on cards) or tape 0 (if Update A is on an existing SLT).

   c. Depress the BOOTSTRAP button.

   d. Depress the RUN button. (If the Update A program is on an existing SLT, perform steps "e" and "f" below.

   e. At halt, enter "UPDATE" into octal locations 145 to 152.

   f. Depress the RUN button.

8. Display the B-address register for "end of run" halt code (i.e., 4777, octal). Refer to Table 6-6 for Update A coded halts.

The Series 200 RG symbolic section (dummy instructions and the Multiply/Divide routines) is placed on the SLT as follows (see Figure 6-3): Bootstrap card, Load card, the RG symbolic deck, and the End Primer and End cards. The Bootstrap, Load, End Primer, and End cards may come from any program (or from the control section of the RG deck), but all four cards must be from the same program. Note also that the program from which the four cards are taken must be in the three-character address mode in order to be used with Report Generator B, or in the two-character mode for use with Report Generator A. The program name to be entered must be DUM3CH (for the three-character address mode) in card columns 67 through 72, inclusive, of the Load card when the dummy instructions are being placed on the SLT. The control, input, data, calculation, and output sections of the RG master deck are themselves condensed decks and consequently may be placed directly on the SLT, preceded only by an INS director card. The Multiply/Divide routine can be placed on the SLT in the same manner as the RG symbolic section dummy cards. The program name for the Multiply/Divide routine is MULDV3 for three-character address mode. An end-of-file card (1EOF) must be placed at the end of the deck, as shown in Figure 6-3.

Once on tape, the Series 200 RG is loaded by depressing BOOTSTRAP, RUN, and RUN (see steps 7c, 7d, 7e, and 7f, above).

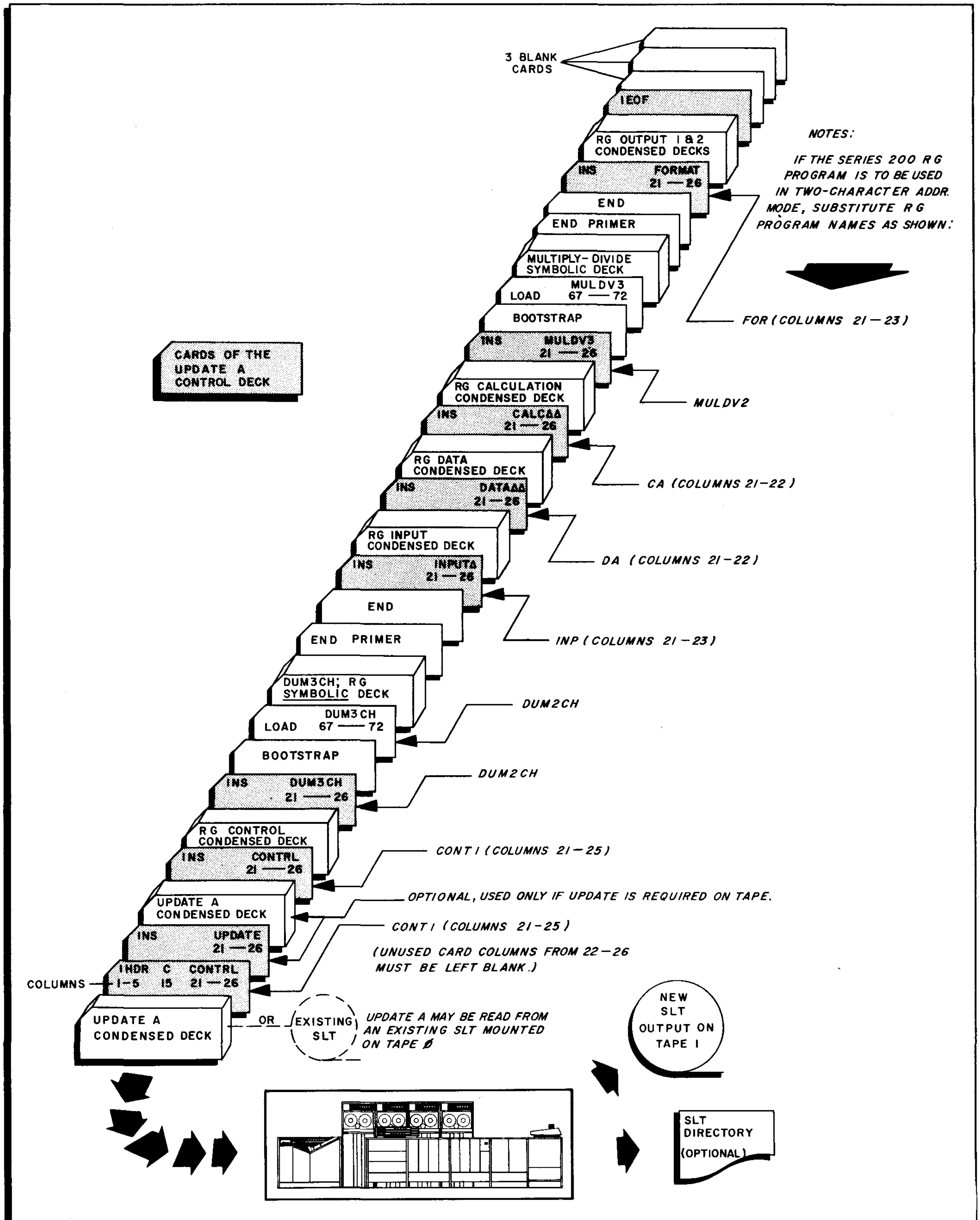

Figure 6-3. Placing the Series 200 RG on Tape

If the RG program is to be used in two-character address mode, it may be placed on the SLT by using the same procedure as that for the three-character mode program. The only change the programmer is required to make in order to load the program in two-character mode is in the header card for the update control section. The name must be changed from CONTRL (three-character) to CONT1Δ (two-character). The RG program names for two-character address mode are shown in the Note in Figure 6-3.

If both the two-character and three-character RG programs exist on the same SLT, a decision must be made: (1) in setting up the tape, which version will be used most often so that the 1HDRΔ card will be punched with the program name of the control section for the most used version (i.e., CONTRL or CONT1Δ); and (2) at running time the version selected in step 1 is called by bootstrapping the tape and pressing RUN; at the halt, depress RUN again. If the version selected in step 1 is not to be called, but instead the other version is to be called, then at the halt after bootstrapping and running, the name of the latter control section must be entered into octal locations 145-152, inclusive.

For example, if it is decided that the three-character version will most often be used, CONTRL will be entered in columns 21 through 26 of the header card. To call in the two-character version, enter 1Δ in octal locations 151 and 152 at the halt. Since the Search routine automatically sets locations 145 through 152 to CONTRL, it is necessary, then, to change only the last two locations (151 and 152).

Table 6-6. Coded Halts for Update A Program

Cause	A Address	B Address	Operator Action
Card reader not in operation	0000	0001	Energize card reader, depress START button and depress RUN button on control panel.
Read error on existing SLT	Buffer address	0010	Depress RUN button on control panel to try to re-read.
Card read error	Buffer address	0011	Run out cards. Check last card (error card) for damage and replace if necessary. Place error card in front of other cards in card reader input hopper and depress START button on card reader to refeed. Depress RUN button on control panel.
Read error on input program tape	Buffer address	0012	Depress RUN button on control panel to try to re-read.
Write error on new SLT	Buffer address	0021	Depress RUN button on control panel to try to re-write, after checking drive for PERMIT status.
Printer error	Buffer address	0022	Depress RUN button on control panel to reprint line containing error.

Table 6-6 (cont). Coded Halts for Update A Program

Cause	A Address	B Address	Operator Action
End of tape on new SLT	0000	0031	Depress RUN to properly terminate new SLT. NOTE This halt occurs when the number of programs to be written on the new SLT exceeds the physical length of the tape. Depressing the RUN button terminates the new SLT with the three records 1EOF, 1ERI, and 1ERI to enable use of the programs which have been recorded on the new SLT. It should be noted, however, that the last program being written on the new SLT is either incomplete and/or additional programs have not been written on the new SLT.
1HDR card missing	0000	4001	Check order of cards in control deck and place 1HDR card first. Place control deck in card reader input hopper, as described in step 6 of the procedures above. Depress START button on card reader to refeed, and depress RUN button on control panel.
Illegal director	0000	4002	Correct control deck by placing cards in proper order and/or replacing illegal director card with correct action director. Start again at beginning of bootstrap and run procedures.
Bootstrap card missing from input program	0000	4003	Correct input program by placing cards in proper order with bootstrap card first. Start again at beginning of bootstrap and run procedures.
Input-program card other than CLEAR between Bootstrap and Load cards	0000	4004	Correct input program by placing cards in proper order. Start again at beginning of bootstrap and run procedures.
Named program not on tape 2	Address of program name from director	4005	Correct control deck by using proper name and action director. Start again at beginning of bootstrap and run procedures.
End of run	7777	4777	The new SLT is complete. Take proper steps to preserve required input and output programs and SLT directory listing, as applicable.

Tape File Termination

The following is a list of the characteristics of a tape file which will terminate that file:

1. If the file is bannered, then either:
   a. A tape mark or
   b. 1EORΔ in character positions 1-5 of the record or
   c. 1EOFΔ in character positions 1-5 of the record or
   d. 1ERIΔ$^{8}_{2}$ in character positions 1-6 of the first item in the record (i. e. , if the file is made up of variable length records, characters 5-10 of the record would contain 1ERIΔ$^{8}_{2}$ in order to terminate the file).
2. If the file is bannerless, then either:
   a. A tape mark or
   b. 1ERIΔ$^{8}_{2}$ in the first six characters of the first item in the record.

Tape Formats of the Generated Program

Pictorial representations of the input magnetic tape formats acceptable to the generated program are presented in Figures 6-4 (standard formats) and 6-5 (non-standard formats). A pictorial representation of the standard, and non-standard print tape formats which are used for magnetic tape output is presented in Figure 6-6.

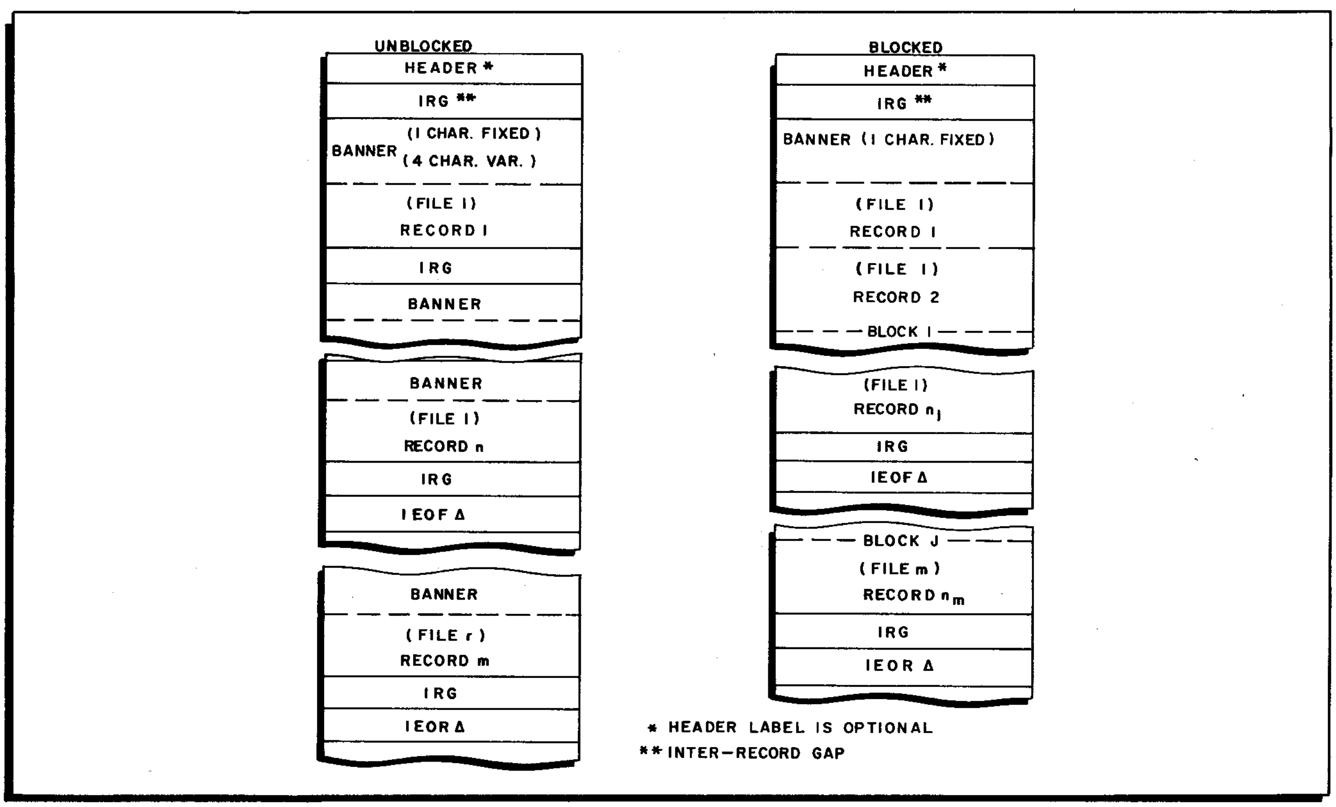

Figure 6-4. Input Magnetic Tape Formats, Standard

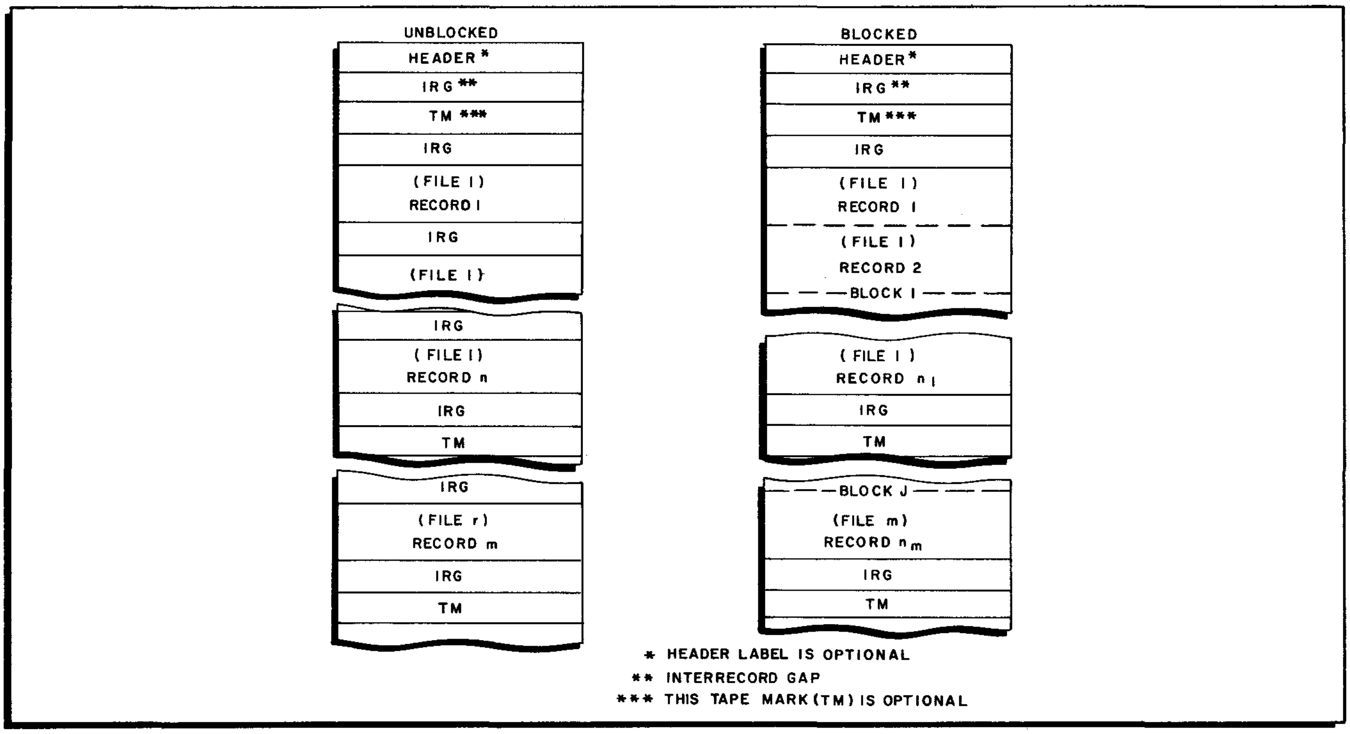

Figure 6-5. Input Magnetic Tape Formats, Non-Standard

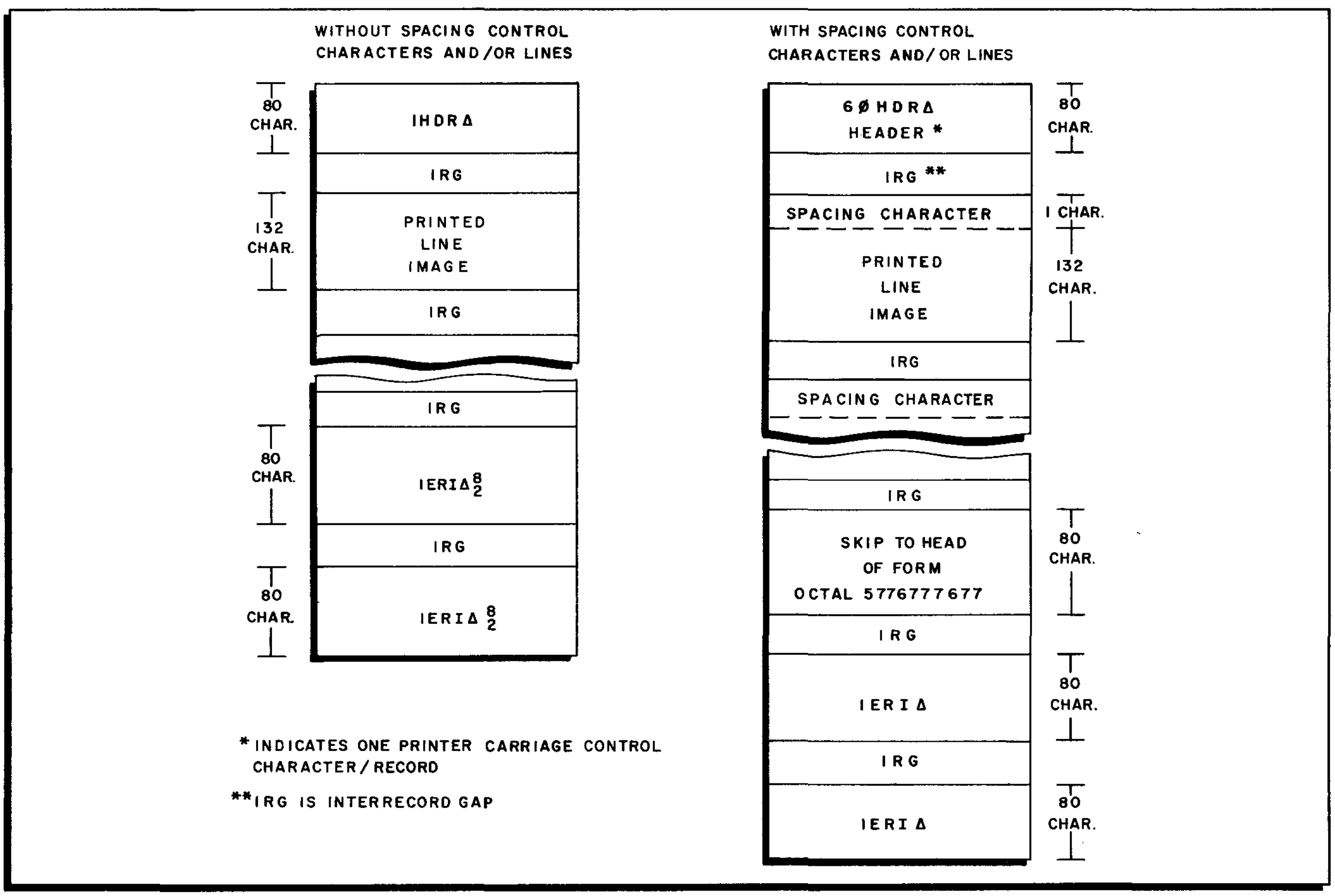

Figure 6-6. Magnetic Tape Output, Print-Tape Formats

APPENDIX A

SERIES 200 CHARACTER CODES

Key Punch	Card Code	Central Processor Code	Octal	High Speed Printer	Key Punch	Card Code	Central Processor Code	Octal	High Speed Printer
0	0	000000	00	0	-	X	100000	40	—
1	1	000001	01	1	J	X, 1	100001	41	J
2	2	000010	02	2	K	X, 2	100010	42	K
3	3	000011	03	3	L	X, 3	100011	43	L
4	4	000100	04	4	M	X, 4	100100	44	M
5	5	000101	05	5	N	X, 5	100101	45	N
6	6	000110	06	6	O	X, 6	100110	46	O
7	7	000111	07	7	P	X, 7	100111	47	P
8	8	001000	10	8	Q	X, 8	101000	50	Q
9	9	001001	11	9	R	X, 9	101001	51	R
	8, 2	001010	12	'		X, 8, 2	101010	52	#
#	8, 3	001011	13	=	$	X, 8, 3	101011	53	$
@	8, 4	001100	14	:	*	X, 8, 4	101100	54	*
Space	Blank	001101	15	Blank		X, 8, 5	101101	55	"
	8, 6	001110	16	> [1]		X, 8, 6	101110	56	≠ [1]
	8, 7	001111	17	&		X, 0	101111	57	I [1]
&	R	010000	20	+		8, 5	110000	60	< [1]
A	R, 1	010001	21	A	/	0, 1	110001	61	/
B	R, 2	010010	22	B	S	0, 2	110010	62	S
C	R, 3	010011	23	C	T	0, 3	110011	63	T
D	R, 4	010100	24	D	U	0, 4	110100	64	U
E	R, 5	010101	25	E	V	0, 5	110101	65	V
F	R, 6	010110	26	F	W	0, 6	110110	66	W
G	R, 7	010111	27	G	X	0, 7	110111	67	X
H	R, 8	011000	30	H	Y	0, 8	111000	70	Y
I	R, 9	011001	31	I	Z	0, 9	111001	71	Z
	R, 8, 2	011010	32	;	,	0, 8, 2	111010	72	@
●	R, 8, 3	011011	33	.		0, 8, 3	111011	73	'
□	R, 8, 4	011100	34	)	%	0, 8, 4	111100	74	(
	R, 8, 5	011101	35	%		0, 8, 5	111101	75	C_B
	R, 8, 6	011110	36	□		0, 8, 6	111110	76	□ [1]
	R, 0	011111	37	? [1]		0, 8, 7	111111	77	¢ [1]

[1] Symbol which will be printed by a printer which has a 63-character drum.

COMPUTER-GENERATED INDEX

HONEYWELL EDP TECHNICAL PUBLICATIONS
USERS' REMARKS FORM

TITLE: REPORT GENERATOR A AND B SOFTWARE MANUAL

DATED: DECEMBER, 1965

FILE NO: 123.6605.000A.1-080

ERRORS NOTED:

Fold

SUGGESTIONS FOR IMPROVEMENT:

Fold

FROM: NAME ______________________ DATE ____________

COMPANY ______________________

TITLE ______________________

ADDRESS ______________________

____

Cut Along Line

FIRST CLASS
PERMIT NO. 39531
WELLESLEY HILLS
MASS.

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

POSTAGE WILL BE PAID BY

HONEYWELL
ELECTRONIC DATA PROCESSING DIVISION
60 WALNUT STREET
WELLESLEY HILLS, MASS. 02181

ATT'N: PUBLICATIONS AND WRITING SERVICES DEPARTMENT

Cutting Line

Honeywell

ELECTRONIC DATA PROCESSING

HONEYWELL EDP TECHNICAL PUBLICATIONS
USERS' REMARKS FORM

TITLE: SERIES 200
REPORT GENERATOR A AND B
ADDENDUM #1
SOFTWARE MANUAL

DATED: MARCH, 1966

FILE NO: 123.6605.000A.1-080

ERRORS NOTED:

Fold

SUGGESTIONS FOR IMPROVEMENT:

Fold

FROM: NAME ______________________ DATE ____________

COMPANY ______________________

TITLE ______________________

ADDRESS ______________________

____

Cut Along Line

FIRST CLASS
PERMIT NO. 39531
WELLESLEY HILLS
MASS.

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

POSTAGE WILL BE PAID BY

HONEYWELL
ELECTRONIC DATA PROCESSING DIVISION
60 WALNUT STREET
WELLESLEY HILLS, MASS. 02181

ATT'N: TECHNICAL COMMUNICATIONS DEPARTMENT

Cut Along Line

Honeywell

ELECTRONIC DATA PROCESSING

CRP