Programming Systems Analysis Guide

IBM 7070 Series Report Program Generator

IBM

IBM

Programming Systems Analysis Guide

IBM 7070 Series Report Program Generator

Preface

This manual has been prepared by Programming Systems to provide detailed information on the internal logic of the IBM 7070/7074 Report Program Generator and its generated output. It is intended for technical personnel who are responsible for adapting the program to special usage, or for diagnosing the system operation.

A program listing of the IBM 7070/7074 Report Program Generator may be obtained by sending a full reel of magnetic tape to the IBM 7070 Program Librarian at Data Processing Library Services, 112 East Post Road, White Plains, New York.

Familiarity with the following manuals is necessary to understand the material in this manual:

Reference Manual, IBM 7070/7074 Compiler Systems: Report Program Generator, Form C28-6113.

Programming Systems Analysis Guide, 7070 Series Autocoder, Form C28-6149.

Reference Manual, IBM 7070 Series Input-Output Control System, Form C28-6175.

IBM 7070/7074 Compiler Systems: Operating Procedure, Form J28-6105.

Contents

List of Illustrations

IBM Programming Systems Department has developed the Report Program Generator for the IBM 7070 Series Data Processing Systems. The purpose of this program is to facilitate the preparation of reports with a minimum of programming time and effort.

A description of the report and what is to be included in it is specified to Report Program Generator on four types of cards. These report specification cards are designed to give a pictorial representation of the report. They are in a machine-independent form so that use of Report Program Generator is not limited to programmers, but can be used by those with a knowledge of reports.

Report Program Generator does not *produce* the report but *generates* a report program which prepares the report. The generated report program is written in load-card format and can be used as often as needed. However, since Report Program Generator is a generalized program, it may not generate the most efficient program for every report that is to be prepared. Thus, the user may want to modify the report program to fulfill the requirements of a certain report. As report specification cards are machine independent but not free-form, that is, each card is limited in format and information, the power of Report Program Generator is limited. Because it was designed to be used by nonprogramming personnel as well as by programmers, however, report specifications are simple and user modifications readily inserted.

Because Report Program Generator is a preprocessor to Autocoder, modifications to the report program may be inserted in Autocoder source language. The report program is generated in Autocoder source language by the preprocessor, and Autocoder is automatically called to compile the report program. Before Autocoder is called, a copy of the report program is written on tape in card-image format. This copy of the report program is rewound and unloaded; it can be punched, and user modifications can be inserted. The updated deck is then the source input to an Autocoder run.

In addition to generating a report program, Report Program Generator can generate a presort program, if desired. The presort program is generated if the data file from which the report is to be prepared is not in the correct order to produce a meaningful report. The presort program passes the data file and creates a shorter, intermediate file composed only of those fields from the data file which will be used in preparing the report. The intermediate file is used as input to a Sort 90 run and the sorted intermediate file becomes the input data file to the report program. On this type of run, the presort program leaves the necessary Sort 90 control card information in storage.

The processing flow depends on the need for a presort program; Chart AA illustrates the processing flow when no sort is required (when a presort is not generated). The report specifications are read from cards or tape and the report program is generated by Report Program Generator. The symbolic report program is then compiled by Autocoder and written in load card format. The report program processes the data file and prepares the report, either on-line, off-line, or both as specified by the user.

When sorting is required (when a presort program is generated) the processing flow is more complex (Chart AB). The report specifications are read from cards or tape and two programs are generated — the symbolic report program and the symbolic presort progam. Generation of a multifile run is not practical because at least nine tape units would be required. Con-

AA

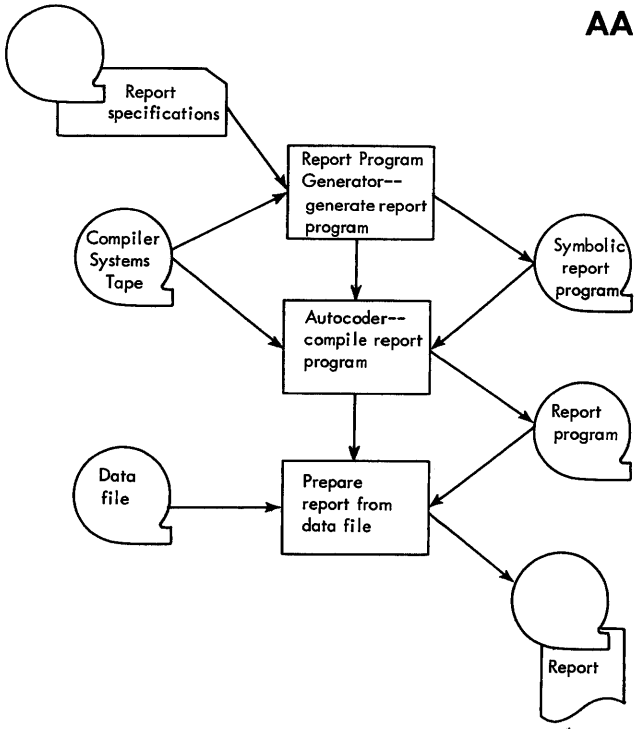


Chart AA. RPG Processing Flow, Sorting Not Required

AB

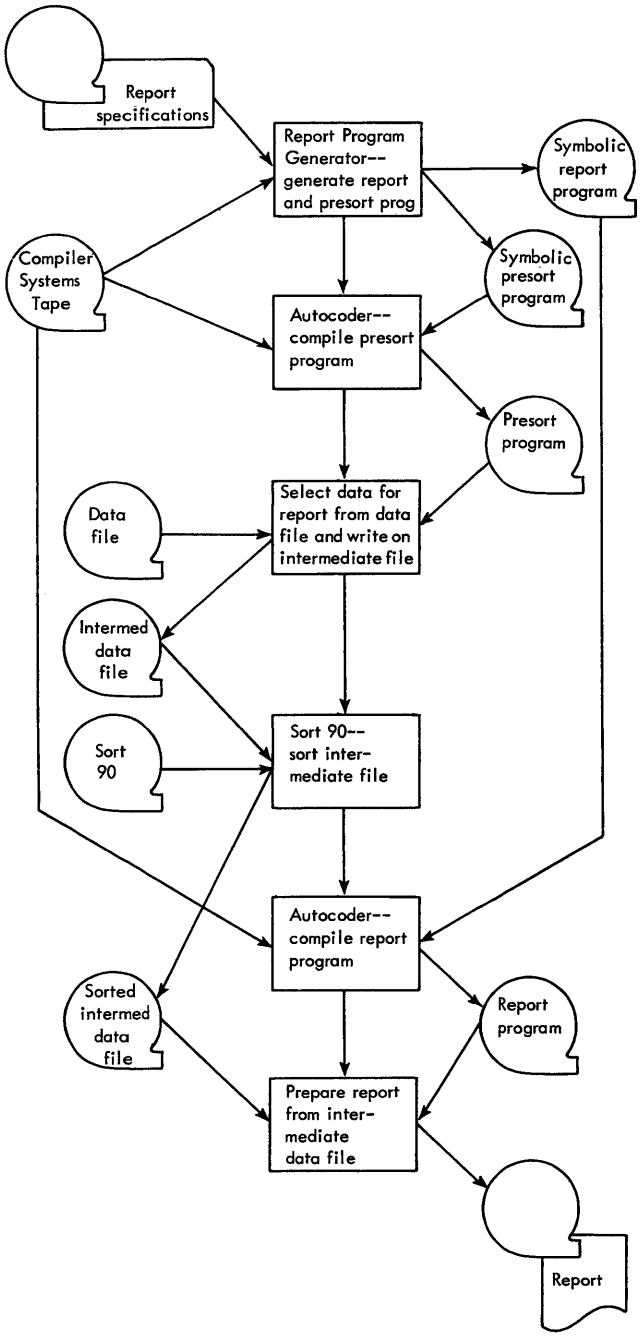


Chart AB. RPG Processing Flow, Sorting Required

sequently, the symbolic presort program is compiled by Autocoder at this time and the symbolic report program is compiled on a subsequent Autocoder run.

After the presort program is compiled, it is used to process the input data file and create the intermediate file. The presort program leaves the necessary Sort 90 control card information in storage so that Sort 90 can be loaded immediately and run with the intermediate file. After the symbolic report program has been compiled on a separate Autocoder run, the intermediate file is processed by the report program and the report is written out.

Features of Report Program Generator

Report Program Generator (RPG) contains features necessary and/or desirable in a printed report.

Heading, detail, and total lines are specified on Report Format (type 2) cards. Detail lines are printed each time information from an input data record is edited into such a line. Heading and total lines may be printed once per report, once per page, or they may be linked to a control break. This option is specified by the user for each format line. A control break occurs when information in a field designated as a control break field changes from one input data record to the next. For example, suppose SALESMAN is located in word 0 (0,9) and word 1 (0,9) and it is a control break field: if the content of SALESMAN is HARRIS until record number 51, but in record 51 the content changes to JOHNSON, a control break has occurred. Control breaks are specified by level to allow the importance of control break fields to be properly interpreted. For example, SALESMAN may be the control break field on level 1 and SALESDISTR the control break field on level 2, and so on. A maximum of nine control break levels can be specified; level 1 is the least significant control break level and level 9 is the most significant level.

Heading and total lines linked to a control break level have a format line number (card columns 2-4 of type 2 cards) of Hn- or Tn-, where H or T designate heading or total line, n designates the associated control break level, and the hyphen (-) represents a sequence number for lines on the same control break level. As many as nine heading lines and nine total lines may be associated with the same control break level by using the sequence number. Heading and total lines to be printed once per report have a format line number of HF- or TF-, where H or T is the heading or total line designation, F indicates a first-time-only heading line or a final total line, and hyphen (-) is again a sequence number. Heading and total lines to be printed on each page have a format line number

of HP- or TP-, where P indicates page heading or page total and hyphen (-) again is a sequence number.

Control break processing can be specified in Field Dictionary (type 3) cards. For each of the nine levels, up to five fields per level may be specified. That is, if the contents of any of the five fields change between two successive input data records, a control break occurs. The heading and total lines associated with a control break level are edited and put out as well as all lower level control break heading and total lines. In other words, if a control break occurs on level 3, all T3- lines will be put out preceded by T1- and T2- lines that contain less significant totals. Then H3-, H2-, and H1- lines are put out in that order (most significant headings first).

Selective data may be put into a report by specifying conditions under which data fields will be edited into a print line in Data Selection Requirements (type 4) cards. In type 4 cards a data field may be specified as positive, negative, non-zero, equal to a given value, or between a range of given values in order to be printed. Further, these conditional requirements can be compounded so that, for example, a certain data field must be positive and between the values of 0-10, 15-17, 21-29, and so on, before being edited into a print line. In this way, a selective report can be prepared from a general file.

Fields can be accumulated before being printed into total lines. A field is accumulated only from those data records in which the associated conditional requirements for that field are satisfied. If a field has no conditional requirements and is to be accumulated, it will be accumulated from every input data record.

Group printing of fields or dollar signs may be specified for fields to be printed in detail lines. Fields or dollar signs which are group printed will appear only following the last heading line printed after a control break or after a skip to a new page.

In the following example, SALESMAN, PRODUCT CODE and the AMT dollar sign have been group printed.

Heading line	QUANT	SALESMAN	PRODUCT CODE	AMT
Detail line	3	JAMISON	01-312-412	$42.37
Detail line	1			15.00
Detail line	7			97.95

The advantage of group printing is that it produces an uncluttered report that is easier to read because only information significant to the first detail line is printed.

The report can be printed on-line, off-line, or both. A report that is being written on tape for off-line printing could also have significant heading and total lines printed on-line to produce an immediately available, condensed version of the report.

Other features of Report Program Generator are line counting which controls the maximum number of detail and heading lines printed on a page, the ability to specify constant fields in report format lines, and carriage control which is under control of the user.

Specifications to the Report Program Generator

The specifications to the Report Program Generator are entered on four types of input cards: the Tape File and Printer Specifications card (type 1 card), the Report Format cards (type 2 cards), the Field Dictionary cards (type 3 cards) and the Data Selection Requirements cards (type 4 cards).

Type 1: This card describes the input and output files, record form, and blocking. If a sort of the input file is desired, the work tapes available to the sort must be described here.

Type 2: A representation of each print line format is contained in a type 2 card (with two continuation cards for 135 print positions). Blank areas, fields to be printed every time, and fields to be printed only when filled with data, as well as designation of lines as heading, detail, and total lines, are shown in the type 2 cards.

Type 3: All fields which may appear in the report are defined. Each field is named and its position in the record is defined. The different print lines a field may appear in, and the rightmost print position of the field in each of these print lines, are designated on type 3 cards.

Type 4: The conditional requirements to be met before a field named in the field dictionary is printed and/or accumulated are specified.

For a more detailed discussion of the features available in Report Program Generator, see *IBM 7070/7074 Compiler Systems: Report Program Generator,* Form C28-6113.

Description of Generated Output

Generated Report Program

The generated report program assumes a standard format within the options chosen by the user. If a feature of RPG is not specified, the routines necessary to process this feature will not appear in the generated report program. However, it may be necessary for the label of the routine to appear. For example, if no fields are to be accumulated, the accumulation routine will not be generated; however, the label of the accumulation routine, QACCUMLAT, must appear and will be a branch to the beginning of the program to process the next data record. The label is necessary because the branches to the accumulation routine are generated before there is an indication that no accumulation is to occur.

The report program is generated in Autocoder statements. These statements include many macro-instructions for editing, moving, logical selection, and accumulation. They also include the input-output macro-instructions, PUT and GET. The generated report program utilizes the IBM 7070 Input-Output Control System. Therefore, a DIOCS entry is generated as well as the DTF's for the input and output files and the input-output areas for these files. Because the generated report program utilizes Autocoder and IOCS, it is relatively easy for the user to insert modifications to expand the abilities of the generated report program. For example, the user might specify a second output tape and calculate totals that would be written on this tape in total lines not used by the generated report program. In this case the RPG run is discontinued when the copy of the generated report program is rewound and unloaded. The copy tape is punched and the desired modifications, written in Autocoder, inserted into the deck. This deck is then used as source input to an Autocoder run.

General Data Flow

DESCRIPTION, FLOW CHART AC

The input and output files are opened and the first-time switch is turned on (01). A data record is obtained from the input file (02) and tested to see if the data selection requirements are met (03). A digit switch corresponding to each conditional requirement is turned on if the record meets the requirement. Condition 99 is a special condition that has a corresponding program switch. If condition 99 is met (04), no further processing is to be done on this record and control branches to block 02 to obtain the next data record. Otherwise, the record is tested to see if the control break fields differ in content from the corresponding fields in the previous record (05). If this is the initial pass through the report program, however, control automatically passes to block 08.

On a control break or the first time through, the contents of the control break fields from the current data record are moved to the test area (08). A test is made of the first-time switch (09) and, if on, it is turned off and the front page heading lines put out (10); control then passes to block 12. If it is not on, the required total lines are put out (11). The accumulation areas for all total lines that were written (or for *all* total lines on the initial pass) are zeroed (12). Heading lines are put out (13) and control passes to block 06 where selected information is edited into the detail lines and the lines are written out (06). Selected fields that are to be accumulated are added to the accumulation areas for their respective total lines (07) and control returns to block 02 to obtain the next data record. When end-of-file occurs (02), all total lines are put out (14), the files closed (15), and the end-of-job halt executed (16).

Initialization and Data Selection

The generated report program is described in detail to aid in making modifications and to explain why the report specifications may have caused errors in the generated output (Charts AD-AJ).

DESCRIPTION, FLOW CHART AD

At the symbolic location QSTART, the input and output files are opened (01), the 7070 is put in sense mode for sign change and field overflow, and a first-time switch, QFIRST, is turned on (02). The control break counter, QCBCTR, is loaded with +0000010009 (03) to force a control break on level 9, the highest level, the first time through the generated report program. An input data record is obtained from the data file, QINPUT (04). On end-of-file condition, control branches to the end-of-file routine (Chart AJ). Under normal conditions, the program switch QCN99 is turned off (05). This switch is a special-condition switch available to the user. If the user specifies in a type 4 card

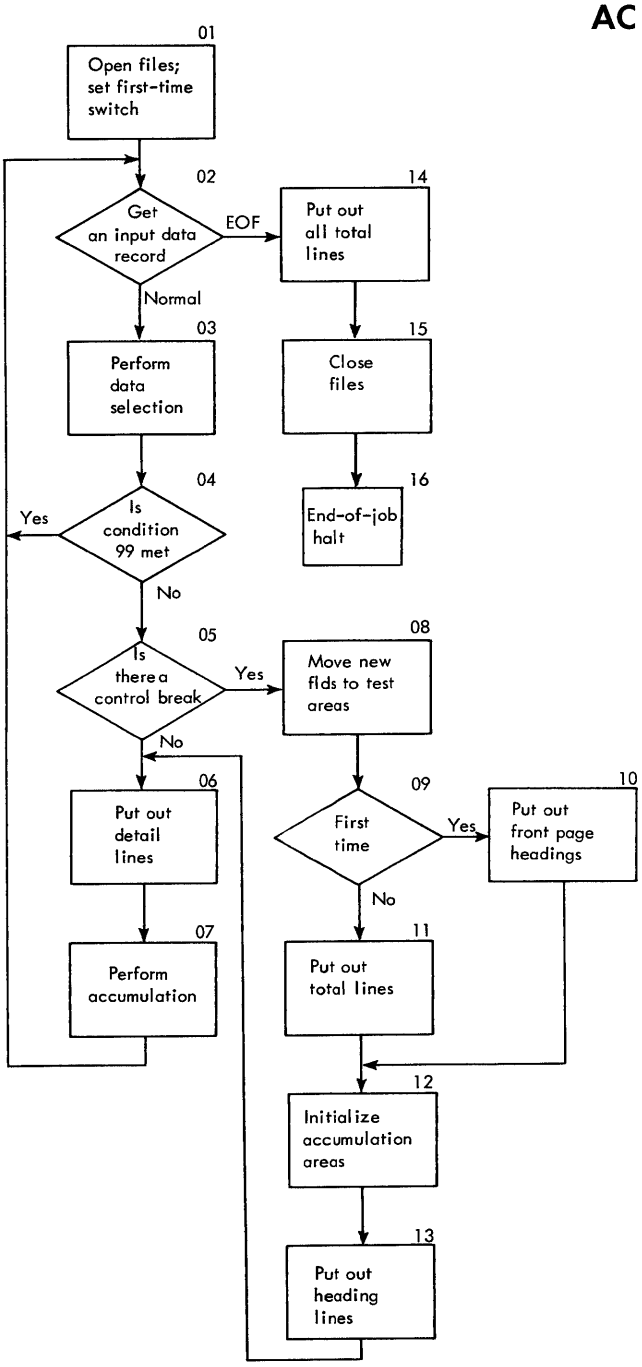


Chart AC. Generated Report Program, Data Flow

AC

that a field, upon satisfying the conditions stated in the type 4 card, has satisfied condition 99, the program switch QCN99 is set to a branch. The branch returns control to QGETREC (04) to obtain the next data record.

The data selection specified in a type 4 card is performed by LOGIC macro-instructions. Before each LOGIC macro-instruction, the conditional-requirement switch is turned off if it has not been previously tested. This insures that the switch does not remain on as a result of processing the previous data record (06). The selection switches contain, in their labels, the number specified as the conditional requirement in card columns 54-55 of the type 4 card. For example, the selection switch for condition 01 is QCN01. The LOGIC macro-instruction tests the fields from the current data record that are specified to meet the first condition. If they satisfy the condition, the selection switch for that condition is set on (07). A SETSW and at least one LOGIC macro-instruction is generated for each condition specified by a type 4 card.

If more than one condition determines a resultant condition, additional LOGIC statements are generated to test the other conditions and set the resultant conditional-requirement switch. Successive LOGIC statements following the first leave the resultant conditional-requirement switch on if it is already on. Conditions are tested in ascending order; if the lowest condition specified was condition 03, it would be the first condition tested. The program switch, QCN99, can be either a branch or a NOP (08). If it is a branch, that is, if condition 99 has been met, indicating that no further processing is to be performed on this data record, a branch is made to QGETREC to process the next data record. If QCN99 is off (NOP), condition 99 has not been met and processing is to continue for this data record. The QFIRST switch is tested (08a), and if it is on, control branches to the control break routine; otherwise, the data record is tested for a control break. If this is the first time through the report program (if QFIRST is on), initialization is performed by forcing a control break on level 9, the highest possible level. Because a control break on the highest level reinitializes the program completely, the original initialization can be performed by forcing a control break on the highest level. The QFIRST switch allows the program to perform additional functions (such as putting out the HF-lines) not included in the level 9 processing. At QCOMP-(09), the fields specified as comprising the highest control break level are compared to the control break test area for that control break level. This control break test area will contain the control break fields from the first data record or from the first data record following the last control break on that level.

A control break on the same level could occur in two

succeeding records or it could be any number of records apart. The control break fields are tested in descending order by level. If an unequal comparison is the result of comparing the control break fields in the current record to the control break fields in the test area (10), a control break has occurred and the control break counter, QCBCTR, is loaded with the level of the break (11). For example, if a control break occurs on level 3, the control break counter is loaded with +0000010003. Control then passes to the control break routine QCBEND (Chart AF).

If no control break occurs, control branches to QSK (Chart AE) to perform detail line processing. The label of the control break testing routine reflects the number of the control break level being tested; for example, the routine which tests for a control break on level 6 would have as a symbolic label QCOMP6. If no control break occurred on level 6, the QCOMP6 routine will branch to QCOMP5. Control break levels must be specified in ascending sequence, starting with level 1; otherwise, an incorrect assembly occurs.

Detail Line Processing

The detail line routine can be entered from either block 10 in Chart AD or from block 38 in Chart AF. The processing flow is logically identical for each detail line specified; however, only one detail line is considered in the following discussion.

DESCRIPTION, FLOW CHART AE

At symbolic location QSK, the first conditional-requirement switch is tested (39). If the condition has not been satisfied, that is, if the switch is off, the print positions for the unselected fields are zeroed (40). If the conditional-requirement switch is on, the dollar sign group-print switch, QDLW, is tested (41). If this switch is off, the print positions containing dollar signs to be group printed are zeroed (42) and control passes to block 46. If QDLW is on, the group-print switch, QGGW, is tested (43). If this switch is off, the print positions for fields to be group printed are zeroed (44) and control passes to block 46.

The QDLW and QGGW switches are turned on following a control break or a carriage overflow and are turned off after they are used the first time, thereby controlling the group printing function. The selected fields are edited into their respective print positions in the detail line and QPSW, the put-out switch, is turned on (45). If there are more conditional requirements, they will be processed exactly as described (blocks 39 through 45). The put-out switch, QPSW, is tested (46) and if it is off (that is, the detail line is not to be put out) control passes to the accumulation routine (Chart AH). If QPSW is on, QSKW is tested (47). The function of the QSKW switch is to indicate that the carriage should be spaced before the detail line is printed. The carriage spacing option is specified in card column 7 of the D01 card by a digit 1-9 representing the number of spaces desired. Normally, these spaces are taken before the level heading lines are printed following a control break or a carriage overflow. Where no level heading lines are specified, however, the QSKW switch controls the spacing option by allowing the spaces to be taken before the first detail line is printed following a control break or a carriage overflow. If QSKW is on, it is turned off and control passes to the QBLANK routine to effect the carriage spacing (48). Control then passes to QSK- where QPSW is turned off and the detail line written as specified in card column 6 of the type 2 card; the group-print switches are turned off (49) and control branches to the accumulation routine (Chart AH). Note that if there are more detail lines, they will be logically processed in the same way as the above detail line, starting at block 39. Control branches to the accumulation routine after all detail lines have been processed.

The group-print switches are only turned off by the highest detail line. This causes group-print fields to continue printing until the last detail line is printed. For example, if three detail lines have been specified and only the D01 and D02 lines print following a control break or carriage overflow, the group-print fields in the D01 and D02 lines will print each time until the D03 line is printed.

Control Break Processing

DESCRIPTION, FLOW CHART AF

At QCBEND, the alternate control break counter, QCBCTRA, is loaded with interchange from QCBCTR, the control break counter (12). QCBCTRA is then used as a counter in moving the control break fields from the current break level, and all less significant levels to their respective control break test areas (13). If a control break occurs on more than one level in the same data record, the routines significant to the lower level control breaks will be performed as a function of the highest control break. For example, if a control break occurs in a data record on levels 1 and 3, the control break on level 1 would cause the heading and total lines, specified as level 1 format lines, to be put out and the new control break value for level 1 to be moved to the level 1 test area. However, the control break on level 3 causes the heading and total lines for levels 1 and 2, as well as level 3, to be put out, and the new control break values for levels 1, 2, and 3 to be moved to their respective test areas. The new level 2 control break values moved into the level 2 test area will be the same as the values already present in that area because no control break occurred on level 2. On

every control break, the program control switches are reinitialized. The dollar sign ($) group-printing switch, QDLW, the field group-printing switch, QGGW, and the spacing-code switch, QSKW, are set on; the detail line put-out switch, QPSW, is turned off (14). A report need not contain detail lines; if it does not, these switches will not appear. A test is made of QFIRST (15). If this switch is on, control branches to QFON to perform program initialization procedures. At QFON (24), the first-time switch is turned off, and the program branches to QFRS where all HF- lines are put out. At QFCON, the accumulation areas for all total lines are initialized to zero (25), and the variable non-accumulated fields in the total lines are initialized (26). The line counter, QLCT, is initialized to the maximum number of lines specified in the type 1 card and if page numbering is being used, the PAGE field is initialized to 1 (27). The program then branches to QHP to put out all HP- lines (28) and control passes to QFOFF1 (31).

If this is not the first time through the generated program, QFIRST will be off at block 15 and control passes to block 16 where QCBCTRA is again loaded with interchange from QCBCTR. At QFOFF (17), the program BLX's to QTOTAL + QCBCTRA, the index word containing the level of the break.

The symbolic location QTOTAL is followed by nine branch instructions. These nine instructions contain a branch to the total line level routine for each control break level specified, or a branch to 0+x94. If a control break level above three has not been specified, QTOTAL will be followed by three branches to the three total line level routines and six branches which return control to QFOFF+1 (29). Here a test is made to determine if the number of total lines indicated in QCBCTRA have been put out. If not, QCBCTRA is incremented (30) and control returns to QFOFF (17). Because the counter QCBCTRA is used, and is loaded with interchange from QCBCTR, the total lines are put out in ascending sequence; that is, the least significant total line is put out first followed by the more significant total lines.

At QPT-, the line counter increment is set with the value of the carriage control character for the total line being processed. Control branches to the page turning routine (18). At QPT, the line counter is incremented and checked to see if it exceeds the maximum number of lines per page (18a). If not, the accumulation areas are edited into the total line and the line is written out on-line, off-line, or both as specified in card column 6 of the type 2 card (19). If a total line has an N in card column 7, meaning skip to a new page after the current control break, entry will not be made to the QPT routine before printing that line or any higher level total lines. Because all less significant total lines are put out as a result of a higher level control break, the skip to a new page would occur at the time the total line with the N was put out if the QPT routine were entered.

The accumulation areas, for any fields that are to be accumulated in this level of total lines, are moved to the specified print positions in the total lines (18). The total lines are put out either on tape, on the on-line printer, or on both as specified in card column 6 of the type 2 cards (19). At QLT- (20), the accumulation areas for this level of total lines are reinitialized by being set to zero. Any totals which are to be retained to the end of the report must be included in TF- lines. Counters are not "rolled" in the unit record sense from minor totals into intermediate totals. Instead, there is a separate counter for each specified appearance of an accumulated field as defined in its type 3 card. These accumulation areas, or counters, are defined as subsequent entries under one DA, labeled QACAREA. The labels of the subsequent entries contain the format line number and rightmost print position (RPP). For example, the accumulation fields for total line T21 might be:

QACAR21047	36, 39 A 4.0
QACAR21069	43, 49 A 6.1
QACAR21095	52, 54 A 3.0
QACART21	30, 59

The label QACART21 is used as the operand of a ZERO macro-instruction to zero the accumulation area following a control break and the printing of the T21 line.

The conditional-requirement switches for the non-accumulated fields that are to appear in a total line are tested (21). Those non-accumulated fields that are selected are moved into the specified total lines (22) and the print positions for the unselected non-accumulated fields are zeroed (23).

NOTE: If there are more total lines on a level, they are processed by in-line coding logically identical to that described (blocks 18 through 23).

Control then passes to QFOFF+1 (29) where the alternate control break counter QCBCTRA is tested to determine if all total lines for this break have been put out. If not, QCBCTRA is incremented (30) and control branches to QFOFF (17) to put out the next higher level of total lines. When all total lines have been put out, control passes to QFOFF1 (31). Control also returns to block 31 after the first-time routine that ended at block 28 and after each heading line is put out (block 38).

At QFOFF1 the same type of counting is used that was used to move the control break fields to the test areas (13) and to branch to the total line level routines (17). In this instance, the control break counter, QCBCTR, is utilized to count the heading lines that are to be put out as a function of this control break. Because QCBCTR is used to control the order in which heading lines are

put out, the heading lines are put out in descending order by control break level. For example, if a control break occurs on level 3, QCBCTR is loaded with +0000030001 and the branch to QHDR+QCBCTR puts out the heading lines of level 3 followed by the heading lines of level 2 and level 1 (31). Before the highest level heading line is put out, the carriage is spaced the number of lines specified in card column 7 of the D01 card.

At QPH- (32), the line count increment, QLCA, is set with the value of the carriage control code specified for this heading line, and the program branches to the subroutine QPT. At QPT, the line counter, QLCT, is incremented by QLCA and checked to see if it exceeds the maximum number of lines per page. The conditional requirement switches controlling any variable fields which can be moved into this level of heading line are tested (34). The selected fields of the current data record are edited into the print positions in the heading format lines of this level (35), and the print positions in the heading lines for the unselected fields are zeroed (36). The heading lines on this level are put out on-line, off-line or both as specified in each type 2 card (37). A test is made to determine if all heading lines necessary for this control break have been put out (38); if so, control branches to the detail line routine, symbolic location QSK (Chart AE). If all the heading lines pertinent to this break level have not been put out, control returns to QFOFF1 (31) to put out the next less significant level of heading lines.

Page Turning Subroutine

DESCRIPTION, FLOW CHART AG

At symbolic location QPT the line counter, QLCT, is incremented by the value of QLCA. QLCA is set with the carriage control character of the line to be printed before entering this subroutine. If the line count does not exceed the maximum number of lines per page as specified in the type 1 card, control returns to the calling routine (53). If the line count does exceed the maximum, QLCT is reset to the value contained in QLCA because the line which caused the forms skip is the first line to be printed on the new page. The subroutine QSET is entered to turn on the group-print switches QGGW and QDLW so that the same controls can now occur that normally occur after a control break, namely, fields and dollar signs will be group printed. The page heading lines are put out (57) and control returns to the calling routine.

Accumulation Routine

DESCRIPTION, FLOW CHART AH

At symbolic location QACCUMULAT (58) a test is made of the conditional-requirement switches for variable fields that are to be accumulated into final total lines. If any of these fields have been selected, they are accumulated into their respective areas for the final total lines (59). If no fields have met the conditional requirements for their accumulation, control passes directly from block 58 to block 60. If any fields have been selected for page and level total lines (60), these fields are accumulated into accumulation areas for each total line in which the fields are to appear (61). If no fields were selected to be accumulated or after such accumulation has been performed, control returns to QGETREC to obtain the next data record, (Chart AD). In the accumulation routine, as in all other routines that process lines of different levels, the processing is performed in a sequential manner. Chart AH illustrates the logic for only one line; however, the logic is identical for each level of total lines.

End-of-File Routine

DESCRIPTION, FLOW CHART AJ

The end-of-file routine is entered when a record is not available through IOCS at symbolic location QGETREC (04). The end-of-file routine is entered at symbolic location QEOF (62) where the control break counter QCBCTR is loaded with +0000010009; this counter is used to put out all total lines by branches to QTOTAL+QCBCTR (63). The coding that was used during the processing of a control break is utilized here to put out the total lines. At QPT-, the accumulation areas are edited into the specified total lines and the total lines are put out as specified in card column 6 of the individual format cards (64).

The accumulation areas for the total lines are zeroed, and any selected fields from the last data record are moved into these lines (65). These latter two functions are unnecessary at this point, but are performed as part of the total line routines. The counter QCBCTR is tested to determine if all total lines have been put out (66). If they have not, control returns to block 63 to put out the next higher level of total lines. When all the level total lines have been put out, control passes to QPTF (67) where the accumulated fields for final total lines are edited into the lines and the lines are put out (68). Each level of final total lines is edited and put out, in the manner described in blocks 67 and 68, by in-line coding. After all total lines have been put out, the input and output files are closed (69) and a halt and proceed 9999 is executed (70). Following this halt instruction is a branch to 308. If the user is running more than one job, he may press START to branch to the load program.

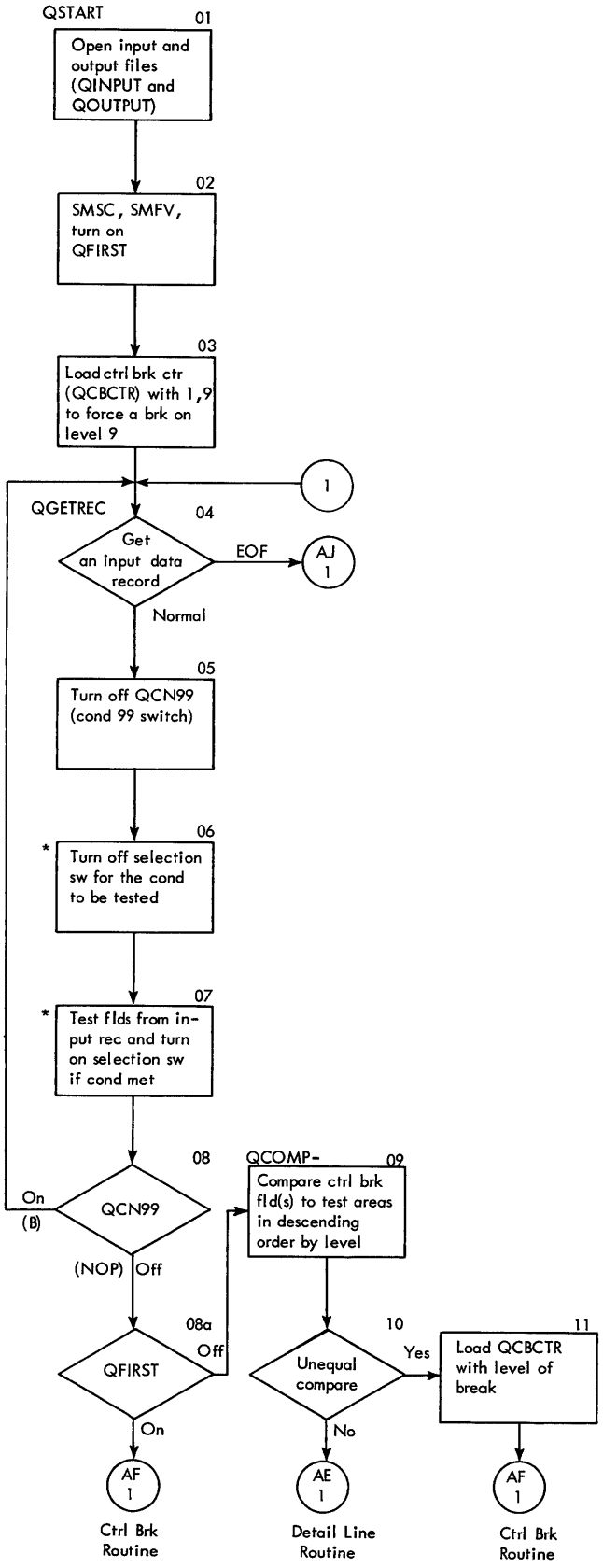


Chart AD. Generated Report Program, Initialization and Data Selection

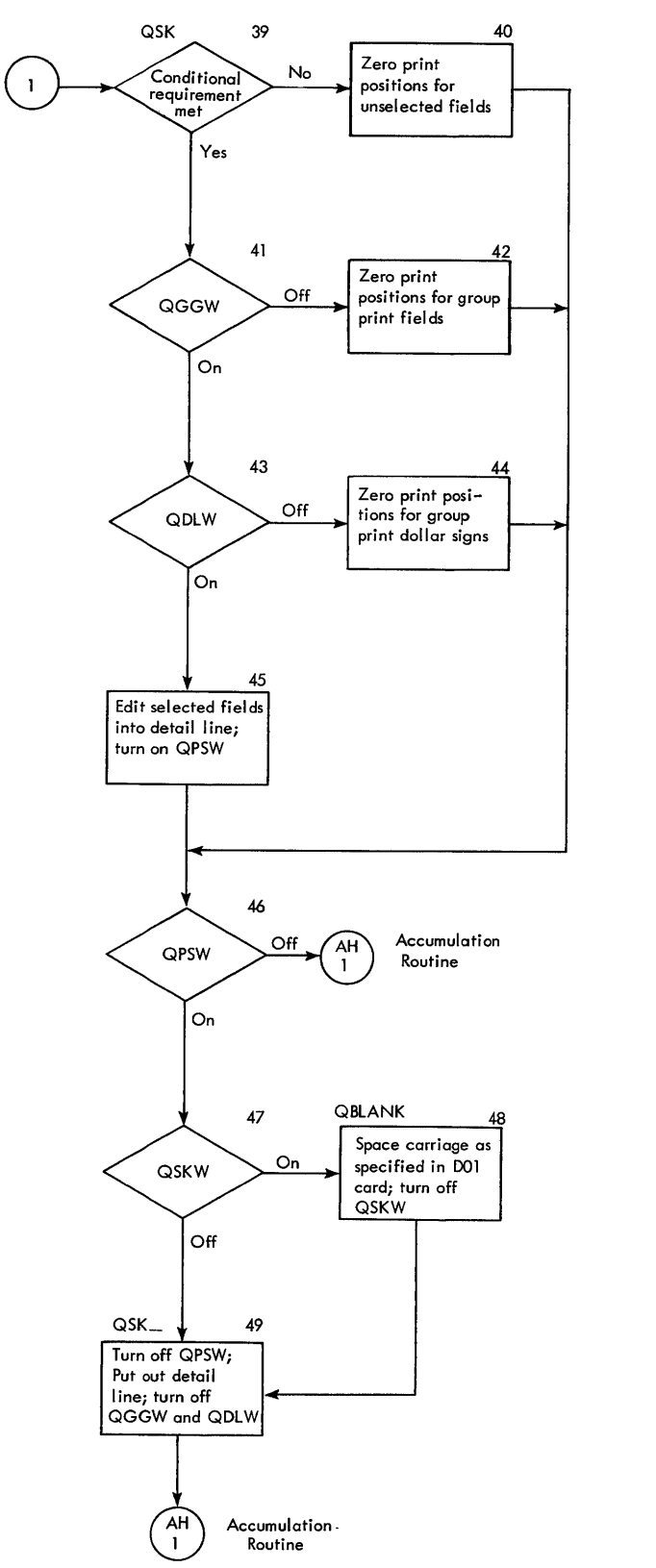
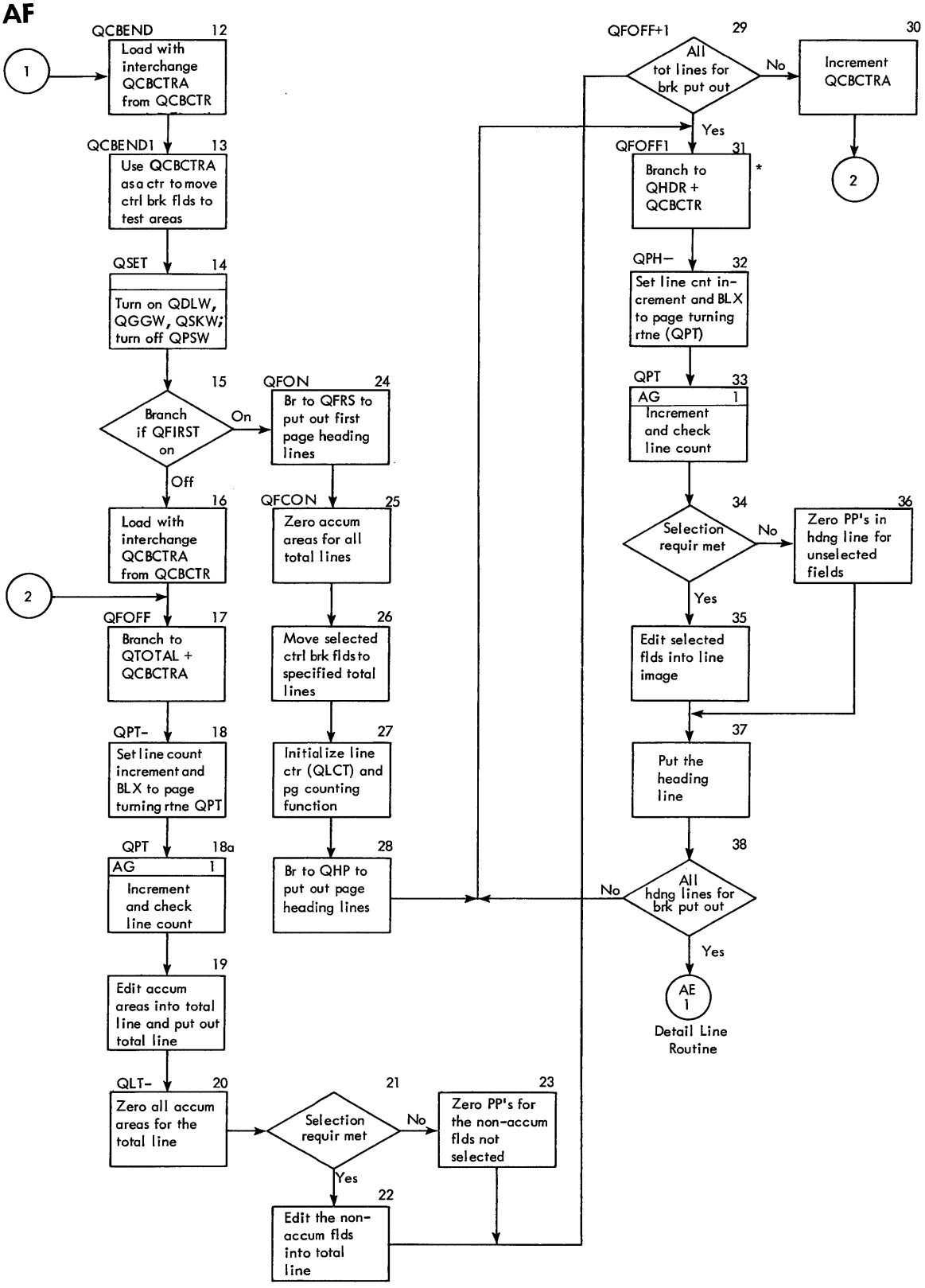


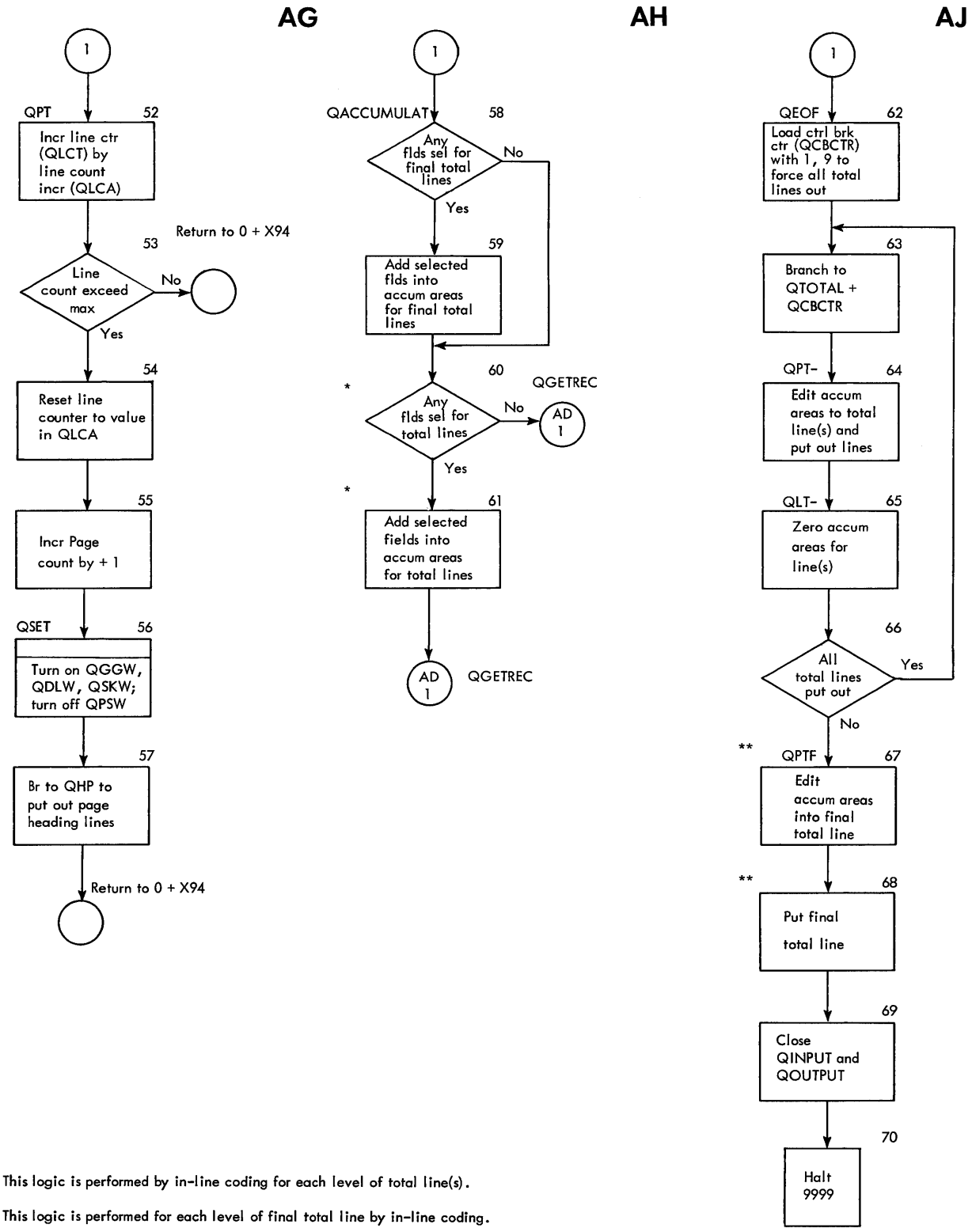
Chart AE. Generated Report Program, Detail Line Processing

AF



*Before the first heading line is put out, the carriage is advanced the number of spaces specified in CC 7 of the D01 card.

Chart AF. Generated Report Program, Control Break Processing



* This logic is performed by in-line coding for each level of total line(s).

** This logic is performed for each level of final total line by in-line coding.

Chart AG. Generated Report Program, Page Turning Subroutine

Chart AH. Generated Report Program, Accumulation Routine

Chart AJ. Generated Report Program, End-of-File Routine

Generated Presort Program

RPG will generate a presort program in addition to the report program if card column 31 of the type 1 card contains a Y. The function of the presort program is to read the input file and write an intermediate file which, to reduce sorting time, is composed only of those fields defined by type 3 cards.

The presort program begins at QSTART where the input and output files are opened (Chart AK). A data record is obtained from the input file and the named fields are moved to the output area. The new record is then written on the intermediate file and control returns to process the next record at QGETREC. When end-of-file occurs for the input file, the files are closed and the program halts and branches to 308 to load Sort 90.

Note that the named fields will not necessarily occupy adjacent digit positions, but may be separated by a maximum of ten digit positions. Thus, the original field definition in regard to digit positions within words is maintained.

Because the presort program creates an intermediate file for sorting in which the fields are relocated from the original input record, the report program will not prepare a correct report from the input file. If a presort program is generated, the report program will process the fields as they are defined in the intermediate file, not as in the original input file.

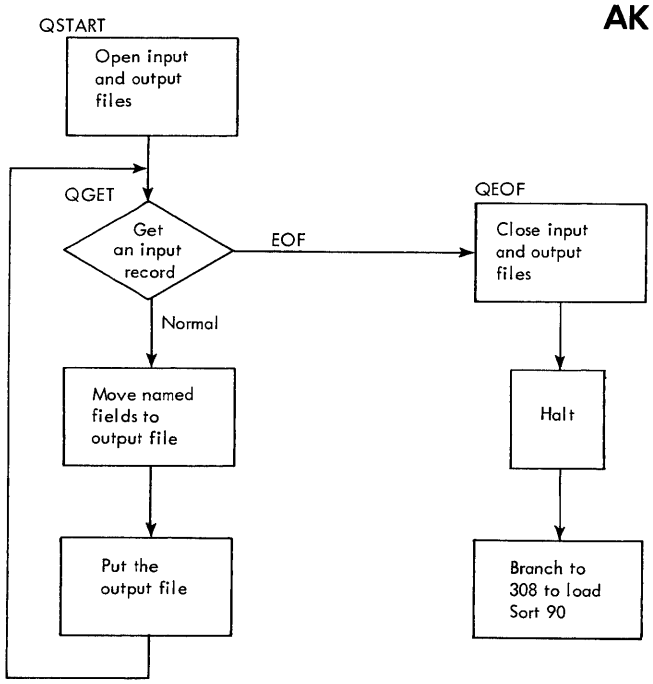


Chart AK. Presort Program

The 7070 Series Compiler System

The 7070 Series Compiler System consists of four major sections:

1. Autocoder is the compiler; it functions as a macro-expander and as an assembly program.
2. Fortran is a preprocessor to Autocoder; it feeds into the macro-expansion phase.
3. Commercial Translator is a preprocessor which produces symbolic Autocoder statements and thus feeds into the scan phase that immediately precedes the macro-expansion phase.
4. RPG is also a preprocessor which enters Autocoder at the scan phase.

The four sections are included on the 7070 Compiler Systems Tape. Provisions for updating the tape, a system control, and a generalized sorting program are also included.

Systems Control is the integrating factor of the Compiler Systems Tape. All sections use Systems Control to perform tape housekeeping on tapes external to the system (such as the symbolic input tape), input-output operations within the sections, and run initialization. Systems Control consists of ten sections: SYCL1 through SYC10. SYCL4-SYC10 are contained in the same coding block and repeated on the systems tape wherever required because in some cases two or three sections use common routines and all seven sections easily fit into one storage load.

SYCL1 contains IOQ, an input-output control system closely resembling IOCS, and STC, a related routine that controls the movement of the systems tape. IOQ only handles form 1 and form 2 records and stacks the tape operations on a first-come, first-served basis. When a request is made to move the systems tape, STC first checks IOQ to see that no other tape operations are waiting, and, when none are waiting, transfers control to IOQ. IOQ and STC occupy storage locations 160-940.

CMREC occupies locations 130-149 and contains all the operating options for a run. The communication record is primarily used by Autocoder but some fields are significant to RPG; for example:

TAPESET1 — the first two work tapes available.
TAPESET2 — the remaining three work tapes available.
HIGHMEMORY — the upper storage limit of the compilation machine.
RPGIND — 1, indicating a presort is to be generated; or 0, indicating only a report program is to be generated.
MAINUNIT — the input unit from which the specifications are read.

SYCL2 is used after the Autocoder section of an RPG run to perform end-of-job procedures.

SYCL3 is called after SYCL1 and CMREC and reads any operating option cards in front of the COMPILE RUN REPORT GENERATOR card and sets these options into CMREC. SYCL3 also performs functions specified in the operating options which pertain to the Compiler Systems Tape in general, for example, checking work tape labels if they were specified. SYCL3 then calls RPG11, Phase 1 of RPG.

SYCL4 is called after the last phase of RPG to perform tape housekeeping functions and to call Autocoder. SYCL8 performs tape housekeeping functions after RPG11 and calls Phase 2 of RPG. SYCL5, SYCL6, SYCL7, SYCL9, and SYC10 are not used by RPG.

SYCL1 and CMREC are not destroyed on successive phase loadings; that is, locations 130-940 remain constant throughout any run with the Compiler Systems Tape. Thus, the starting location for a phase is never below 941.

The sorting program, Autosort, is available wherever needed on the Compiler Systems Tape. Autosort functions only as a sort and not as a means of updating a file; it will sort form 1 and form 2 records only, and no exits are provided for record modifications, insertion, or deletion. It is a two-phase program. The first phase is a sort pass which takes advantage of all natural sequences existing in the file to be sorted. The second phase makes enough merge passes to place the sorted file on one tape. Autosort references a communication block in locations 941-949 where information necessary to the sort is stored. This includes the input tape unit, work tape units, record form and blocking factor, and the control data fields. Upon completion of the sort, Autosort stores the sorted output tape unit in the communications block and calls the next phase. Autosort uses the field HIGHMEMRY2 in CMREC as the highest storage location available for the sort. Autosort comprises Phase 2 and Phase 4 of the RPG processor.

The format of the communication block is outlined in the Interphase and Autosort Communication Areas section of the Appendix.

The over-all relationship of the components of the Compiler Systems Tape is shown in Chart AL. For a more detailed discussion of the Compiler Systems Tape and its components, refer to *Programming Systems Analysis Guide, IBM 7070 Series Autocoder,* Form C28-6149, in the Autosort, CMREC, IOQ and STC sections.

AL

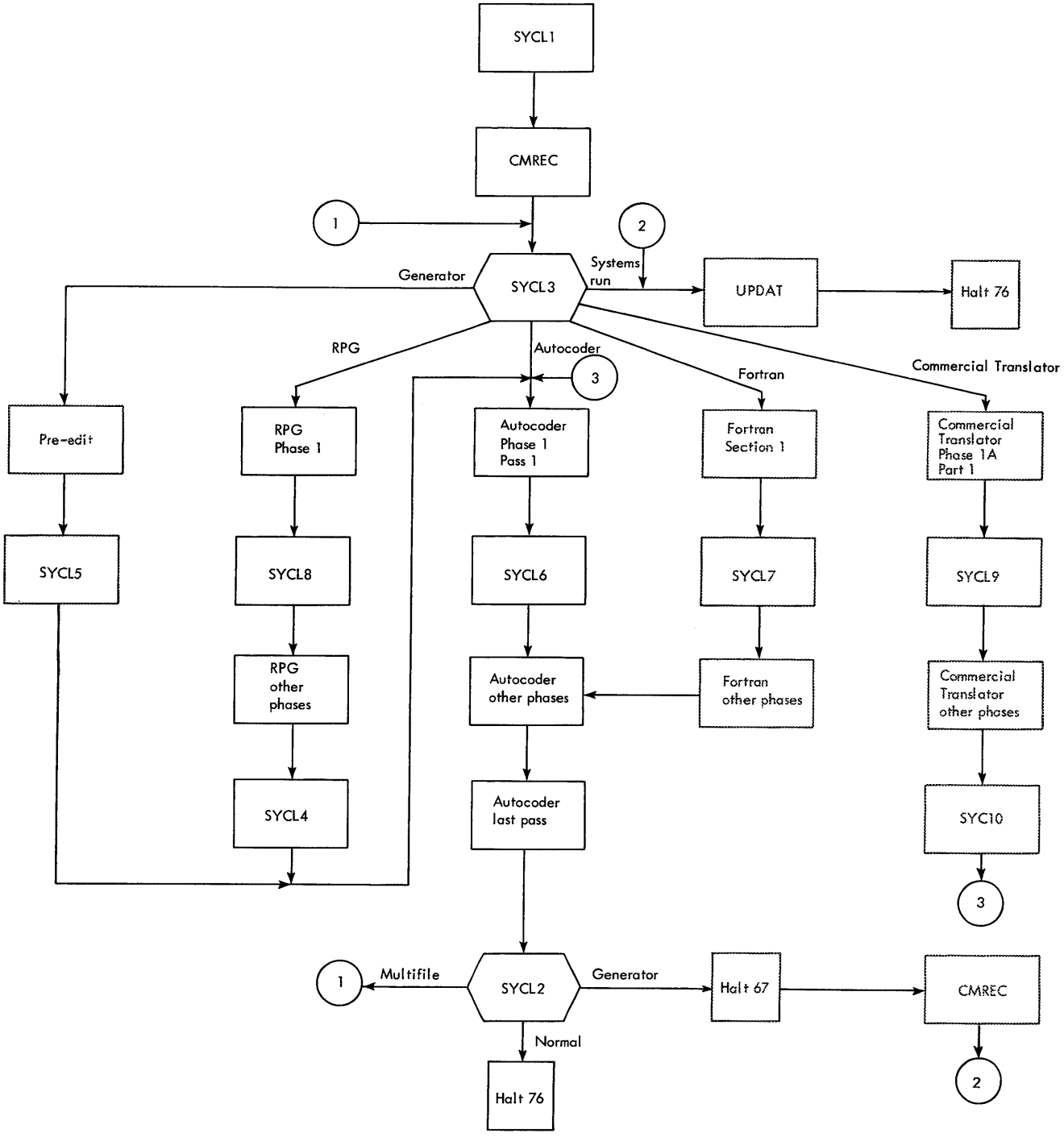


Chart AL. IBM 7070 Series Compiling System

RPG Processor, General Description

RPG utilizes a cross referencing technique to process report format lines. The format lines are defined by type 2 cards as containing areas (print positions) of literal and variable information. Data must be edited into variable fields in format lines. This is accomplished by defining data fields on type 3 cards and specifying the format lines and print positions in which these fields can be printed. By combining data from type 3 and type 2 cards, RPG can generate coding that will edit data into the variable fields in report format lines.

The generation of routines and declaratives from the type 1 card is independent of any other processing (except the maximum number of print positions per line, card columns 68-70; this value is used as the maximum number of characters scanned in format lines). The generation of data selection coding from the type 4 cards is also independent of other processing.

Routines, single instructions, and declaratives that must be generated for every report program are contained in the processor as literals under DC's. This includes the initialization of the report program and obtaining the data records, the end-of-file routine, and the DTF's for the input and output files. Macro-instructions and subroutines which can vary from report to report are built (or compiled) according to the needs of the report program. For example, the editing routines are built one instruction at a time and parameters are inserted where required. Symbolic labels are generated which often reflect such user's labels as the names of fields taken from type 3 cards. Occasionally, generated labels are superfluous because of a lack of communication among various routines within the processor.

Phases 1, 2, and 4 have no generation responsibility and Phase 7 generates only a page/line number for the statements of the report program. Phases 3, 5, and 6 are the generation phases.

Phase 1 scans each type of card with a separate scan. The four scans create four types of tape records and five tables:

Image-4 tape record is a type 4 card edited slightly.
Image-2 tape record is a format line with an assigned serial number.
Preform tape record describes a variable field from a type 2 card.
Line tape record describes the print position of a named field in a format line.
TFPTBL is an edited version of the type 1 card.
CBREAK is a table referencing all control break fields ordered by level.
SORT is a table referencing all sort control fields ordered by level.
SERIAL is a table listing all format line numbers in the order encountered.
NAMEFIELD (or NA) is a table defining each named field from type 3 cards.

The tape records are all 30 words long, although none but the image-2 records can contain that much information; the unused fields are padded with plus zeros. The tables can occupy a maximum of 630 storage locations but will usually be shorter because the SERIAL and NAMEFIELD tables are only as long as necessary.

Phase 2 is a sort which groups all references to a format line and, within any given reference, orders print positions in the lines.

Phase 3 generates DLINE's for a format line containing variable fields or DC's for literal lines. A DLINE is generated by scanning an image-2 record after finding a preform record describing a variable field in the line, and a line record describing where a named field is to appear in the line. If a preform-line record combination does not reference an image-2 record, a DC is created for the literal line. Editor records are created by Phase 3 by an information transfer from an image-2, preform, and line record combination. The editor records are used in Phase 6 to generate edit coding.

Phase 4 is the second sort. It orders the records so that complete subroutines may be generated at one time. This avoids a sort after generation, or extensive communication within the processor to keep track of which routine, or what part of a routine, has been generated.

Phase 5 generates a presort program if a sort has been specified in card column 31 of the type 1 card. It also edits the DC/DLINE records into card format by eliminating the first three words of every record. These three words were control information for the Phase 4 sort. The DTF's and I-O areas for all files are generated by Phase 5.

Phase 6 consists of three loads: Load 1 generates the report program from QSTART to the end of the QEOF routine and includes necessary declaratives such as the DA's that hold the information for control break comparisons. Load 2 generates all editing and put-out routines from QPTF to QACCUMULAT. It also generates an END CNTRL QSTART if load 3 is not called. Load 3 generates the accumulation routines and areas.

Phase 7 inserts a page/line number into the Autocoder card images and writes the declaratives and instructions of the generated report program onto one file.

Two sections of Systems Control are used by RPG: SYCL8 and SYCL4 perform tape housekeeping and initialization functions after Phase 2 and Phase 7, respectively. SYCL8 saves the source input if it is from tape and SYCL4 initializes Compiler Systems Tape operating options to accept the output of RPG as input to Autocoder. Autocoder is automatically called by SYCL4 with no operator intervention.

The general data flow in the RPG processor is shown in Figure 1.

Description of Phase	Phase	File A	File B	File C	File D	File E
Copy type 2 cards; create record describing each variable field from type 2 cards; copy type 4 cards; create records describing the locations where fields are to be printed in a print line; form table directly from the type 1; form NAMEFIELD table and sort and cntrl brk tbls	1	Output (all records are 30 words)	Not used	Not used	Original input (if from tape)	Not used
Autosort by internal serial number (NSC from NAMEFIELD table), RPP, and one-digit sort code	2	Input	Work	Work	Work	Work
Information transfer--Condense the cross referenced information from the tape records into editor records; create DC's or DLINE's describing the print lines; pass image-4 records	3	Input	Not used	Output (DC/ DLINES, editor, image-4 records)	Not used	Not used
Autosort (19-word records) by accumulation code (AI), format line number (CC 2-4 of type 2), RPP, and conditional requirement	4	Work	Work	Input	Work	Work
Generate presort program if CC31 of type 1 is Y; generate report program DA's and DTF's; truncate first three words of DC/DLINE records to yield 16-word Autocoder card images	5	Not used	Output (DA's and DTF's)	Input	Not used	Output (presort, if specified)
Generate report program initialization to control break moves; generate DA's for control break test area and for control break fields storage area	6 (1)	Output (report program instr)	Output (DA's for ctrl brk testing)	Input	Not used	Same
Generate control break move routines, detail line editing, and put-out routines; generate end control statement if no accumulation is required	6 (2)	Output (editing routines)	Output (additional declar)	Input	Not used	Same
Generate accumulation routines and a DA for accumulation areas; generate end control statement	6 (3)	Output (accumulation routines)	Output (DA for accumulation)	Input	Not used	Same
Merge the report program from files A and B onto file C; copy program onto file E if no sort; call systems control section 4 (SYCL4) which will call Autocoder	7	Input	Input	Output (merged files A and B); if sort, file E copied onto file C	Not used	Output (if no presort, file C copied onto file E)

Figure 1. RPG General Data Flow

Phase 1

The function of Phase 1 is to scan the report specifications cards and build internal tables and tape records from the information on these cards. The tables and tape records are illustrated in Figures 14 and 15. Because Phase 1 has no responsibility to interpret and process any information in these records or tables, their function is not explained here, but in the Appendix. The records and tables are processed; an explanation of their specific functions is also included.

Chart AM illustrates the over-all processing flow in Phase 1 in a general manner. Note, however, that the first specifications card read must be a type 1 or the processor will type an appropriate message and halt.

When Phase 1 completes its function, it calls SYCL8 which will in turn call Phase 2 of RPG. Phase 1 also stores the information necessary to the Phase 2 sort in a communications block so that when Phase 2 is entered, the sort assignment can be made.

Phase 1 Control

DESCRIPTION, FLOW CHART AN

At the symbolic location PHAS1CTRLA, the first report specifications card is obtained from the card reader or from the input tape unit (01). The generalized subroutine M1GETCARD is utilized by Phase 1 to obtain the input specifications cards from either the card reader or from the tape unit. If alteration switch 1 is off, the report specifications are considered to be contained on INITAUNIT, and if alteration switch 1 is on, on INITBUNIT. INITAUNIT and INITBUNIT are compiler systems operating options specified by the user. (See *IBM 7070/ 7074 Compiler Systems: Operating Procedure,* Form C28-6113). At symbolic location R1TRADZ1, card column 1 of the input specifications card is tested to determine if this is a type 1 card. If the first card is not a type 1 card, a message is typed stating the type 1 card is missing; the cards should be reloaded and

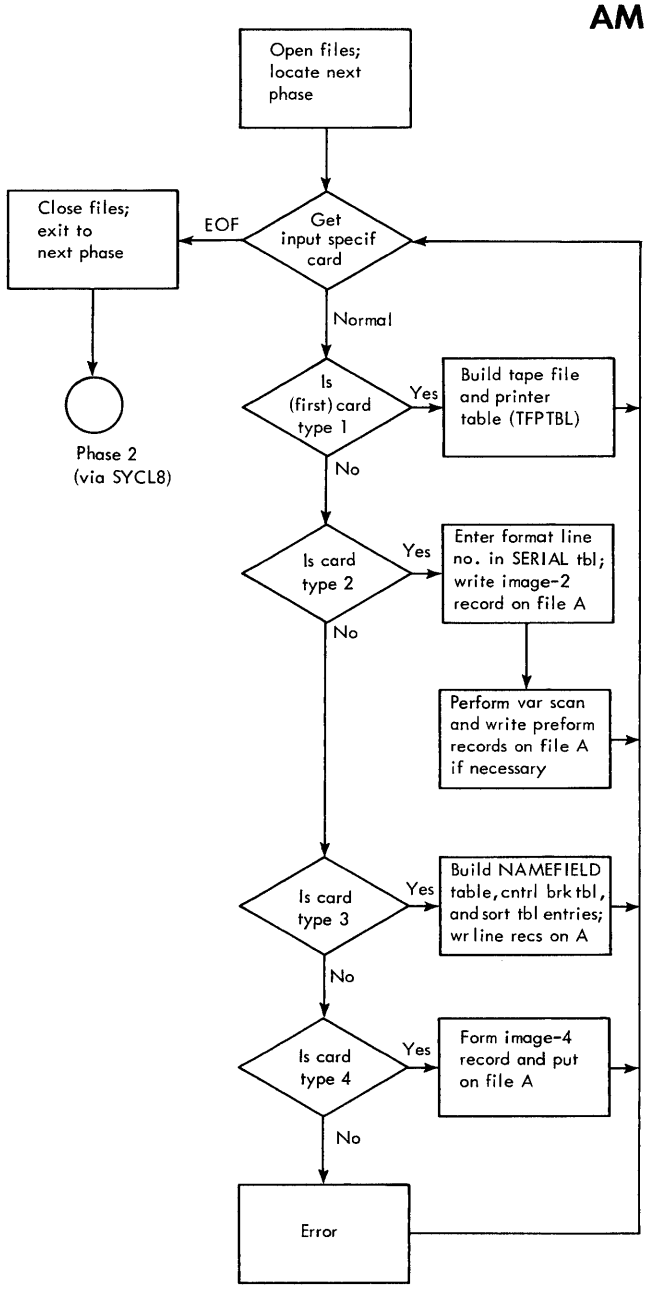


Chart AM. Over-all Processing Flow, Phase 1

START pressed to continue. In this case, control returns to PHAS1CTRLA. If the first card is a type 1 (02), the type 1 scan is entered at symbolic location TYPE1SCAN (04). In the type 1 scan, the tape file and printer table (TFPTBL) is created directly from the type 1 card with alphameric to numeric conversions where required.

At PHAS1CTRLB, the next report specifications card is obtained (05). The card is tested at R1TRAF2 to determine if another type 1 card is being processed (06), and, if so, a message is typed saying that the type 1 card has appeared in the middle of the deck and will be bypassed (07). Control then returns to PHAS1CTRLB to obtain the next specifications card. If the report specifications card is not a type 1 card, the end-of-file switch is tested at symbolic location NEXTTYPE2 (08). As the type 2 cards may have continuation cards, it is necessary, after obtaining a type 2 card, to test the succeeding card and determine if it is a continuation of the preceding card. If it is not, processing continues on the initial card and the succeeding card image will remain in storage. This method of obtaining the next card before processing the current input specifications card is used throughout Phase 1 control. (There is one exception: the initial check of the type 1 card.)

NOTE: A type 3 continuation is processed after the first card is processed. Each time a type 3 card is obtained, it is checked to determine if it is a continuation card before it is processed. If it is a continuation card, only card columns 26-70 will be scanned. When end-of-file condition is reached on the input file, the final specifications card is in storage and remains to be processed. In order that this record may be processed, an end-of-file switch is set on by the end-of-file routine and control returns to the main program to process the last record. If when the end-of-file switch is interrogated at NEXTTYPE2 it is on, the input and output files are closed and the five tables are moved to upper storage (09); control branches to load and execute the next coding block, SYCL8. If end-of-file has not occurred, a test is made to determine if the input card is a type 2 (10). If so, the type 2 card is moved into the work area (11) and the next card or tape record is obtained from the input file (12). A test is made of the succeeding card to determine if it is a type 2 continuation (13). If it is not, control passes to SCANFORMAT (Chart AO, Connector 1). If it is a type 2 continuation, card columns 10-69 are moved into the work area in locations adjacent to the first type 2 card (14). The next card or tape record is then obtained (15) and the card is tested to determine if it is a second type 2 continuation card (16). If it is not, control passes immediately to SCANFORMAT; if it is the second type 2 continuation,

card columns 10-24 are moved into locations in the work area adjacent to the preceding two cards. Since a maximum of 135 print positions is allowed, only columns 10-24 are moved to the work area. Card columns 10-24 of the second continuation card represent print positions 121-135. The next card or tape record is obtained (18) and control passes to SCANFORMAT to create an image-2 record and its associated preform records, if variable fields are present in the line (a variable field contains X's or Z's).

At symbolic location TYPE3OR4, it has been determined that the input specifications card is not a type 1 or a type 2. The card record is moved into a work area (19) and the next card or tape record obtained (20). A test is made to determine if the card in the work area is a type 3 card (21), and, if so, control branches to DICTSCAN (Chart AQ, Connector 1). If the card is not a type 3, a test is made to determine if it is a type 4 (22); if it is, control branches to TYPE4EDIT (Chart AQ, Connector 2). If the card was not a type 1, 2, 3 or 4, an error message will be typed saying that the input card contains an incorrect punch in card column 1 and control returns to NEXTTYPE2 to process the next specifications card.

The cards are processed by their respective scans. SCANFORMAT is the symbolic location where the type 2 scan begins; DICTSCAN is the symbolic location where the type 3 scan begins; TYPE4EDIT is the symbolic location where the type 4 scan begins.

Type 2 Scan

DESCRIPTION, FLOW CHARTS AO AND AP

At symbolic location SCANFORMAT, the format line number from card columns 2-4 of the first type 2 card is used as a search argument in a table lookup of the SERIAL table (24). If a format line number in the SERIAL table is equal to the current format line number, a message is typed saying that a duplicate format line has been specified (25) and control returns to NEXTTYPE 2 in Phase 1 control to process the next report specifications card. If the format line number is not already in the SERIAL table, it is entered into it. The SERIAL table thus becomes a list of format lines in the order in which they are received.

A serial number is taken from the index word referencing the SERIAL table and stored in the preform record area (26). At this point, the entire type 2 work area is prefaced by one word of control information for the Phase 2 sort and then put out on file A as an image-2 record. The preform sort code, 1, is stored into the preform record, word 0, position 6 (27). At symbolic location M1SFAA3, the scan of the type 2 card proceeds. The first character is obtained from the image-2 work area (28) and a test is made to determine if this character is blank (29). If it is blank, the rightmost print position counter (RPP) is incremented by 1, and the character counter (KARND) is incremented by 2 (30). The character counter is incremented by 2 because the characters are stored as aphameric double digits.

KARND cannot exceed 120, 240, 270 (60, 120 or 135 character print positions); the limit is determined by the number of continuation cards for this format line. KARND is tested to see if it exceeds the maximum number of print positions that can be contained on the type 2 card and its continuation cards (31). If it does, control returns to NEXTTYPE2 in Phase 1 control to process the next card. If KARND does not exceed the maximum number of characters, control returns to M1SFAA3 to obtain the next character and scan it.

If the character is not blank, a test is made to decide if the character is a variable, i.e., an X or a Z (32). If so, the decimal point indicator DECIND is turned off (33) and the variable scan entered at VSCAN (Chart AP). If the character was not an X or a Z, a test is made to see if it is a decimal point (34) and, if not, control returns to M1SFAA4, block 30, to update the counters and obtain the next character. If the character is a decimal point, DSCAN is entered, where RPP and KARND are incremented by 1 and 2, respectively (35). KARND is then tested to see if it exceeds the maximum number of characters that are in the image-2 work area (36), and, if so, control returns to NEXTTYPE2 to process the next specifications card. If KARND has not been incremented to a value greater than the maximum number of characters in the image-2 work area, the next character is obtained from the image-2 work area under control of KARND (37). A test is made of this character to determine if it is an X or a Z. If it is, control passes to block 40. If it is not an X or a Z, a test is made of the character to determine if it is a comma (39), and, if so, control returns to DSCAN to update the counters and to obtain the next character from the image-2 work area. Thus, if a decimal point is not followed by a variable (an X or Z) or a neutral character (the comma), it is not part of a variable field and control returns to M1SFAA4 to scan the next character. If it is the start of a variable field, the decimal point indicator is turned on and word 1 in the preform record area is set to +0 (40). Because a decimal point has been found at the beginning of a variable field, the LD field in the preform record area must be zeroed to reflect no digits to the left of the decimal point. Control then branches to VSCAN to scan the variable field, (Chart AP, Connector 1).

Variable Scan

DESCRIPTION, FLOW CHART AP

The variable scan is entered when the first character from a variable field is found or when a decimal point is found outside a variable field and it is determined that a variable field follows the decimal point. The variable scan is responsible for filling the remaining fields of the preform record.

At symbolic location VSCAN, the counter LD is initialized to +1 and the contents of the RPP counter are stored in the LPP field of the preform record (41). At the start of the variable scan, the RPP counter references the leftmost print position of the variable field. The W indicator, WIND, and the last character indicator, KIND, are set off (42). At M1SFAA6, the RPP counter and the character counter, KARND, are incremented by 1 and 2, respectively (43). KARND is then tested to determine if it exceeds the maximum number of characters in the image-2 work area (44). If not, it is used to obtain the next character from the image-2 work area (45). The character is subjected to a series of tests to determine which variable character it is. If the character is a blank (46), the variable field has been completely scanned and control passes to block 58 to exit from the routine. If the character is an X or a Z (47), the LD counter is incremented by 1 (48) and control returns to M1SFAA6 to update the counters and get the next character. If the character is a comma (49), control returns immediately to M1SFAA6. If the character is a W (50), the indicator, WIND, is set on (51) and control branches to M1SFAB1 to perform end-of-scan procedures for the variable field just completed.

If the character is not a decimal point (52), it is not a variable character and consequently the end of the variable field has been reached. Therefore, control branches to M1SFAB1 to perform end-of-scan procedures. If the character is a decimal point, the decimal point indicator, DECIND, is interrogated (53) and, if on, a message is typed saying a duplicate decimal point has been found in a variable field (54) and control returns to M1SFAA6 to update the counters and obtain the next variable character. If DECIND is off, it is turned on and the contents of the LD counter are stored in the LD field of the preform record (55). The LD counter is then reinitialized to 0 (56); from this point until the end of the variable scan, it counts the number of digits to the *right* of the decimal point.

If KARND exceeds the maximum number of characters contained in the image-2 work area (44), the last character indicator, KIND, is set on (57) and DECIND is interrogated (58). If DECIND is on, the contents of the LD counter are stored in the RD field of the preform record (59) because the LD counter contains the number of digits to the right of the decimal point. Control branches to M1SFAA9.

If the DECIND is off, the contents of the LD counter are stored in the LD field of the preform record (60). The contents of the RPP counter are then stored in the RPP field of the preform record and the record is put on file A (61). The W indicator is interrogated (62) and, if on, it is turned off (63). Control returns to VSCAN to scan the variable field which is immediately to the right of the field just scanned.

The first character of the new variable field has already been checked (the W that caused the exit to M1SFAB1) so that the variable scan can be entered from block 63 in the same way as from block 33 or block 40. If the W indicator is off, the last character indicator, KIND, is interrogated (64) and, if not on, control returns to M1SFAA4 to continue scanning the line, (Chart AO, Connector 2). If KIND is on, i.e., the last character has already been obtained from the image-2 work area, KIND is turned off (65), and control branches to NEXTTYPE2 in Phase 1 control to process the next input specifications card.

Type 3 Scan

DESCRIPTION, FLOW CHART AQ

The type 3 scan begins at symbolic location DICTSCAN where a test is made to determine if this is a continuation of the previous type 3 card (66). If so, control branches to LINESCAN, block 71. If not, a NAMEFIELD table entry is created for this field. Unlike type 2 continuation cards where the one or two continuation cards are processed with the first card, type 3 continuations are processed individually.

For each field named by a type 3 card, an entry is made into the NAMEFIELD table. An entry is four words long; it consists of the field name (words 1 and 2), the location of the field (word 3), the characteristic of the field as alphameric or numeric (sign position of word 4), and the number of decimal digits (word 4-0, 1). The remainder of the fourth word is unused unless Phase 5 generates a presort program; in that case, the fourth word will be used to define the new location of the field in the intermediate file. The third word also contains a two-digit serial number, the NSC, which is an index to the name of the field; that is, instead of carrying the full ten-character name in the associated tape records, the two-digit NSC is used to identify the field. In this way, a tape record associated with a type 3 card can readily be related to the field in the NAMEFIELD table through use of the NSC.

The name from this type 3 card is condensed by eliminating blanks (66a). It is then used as a search argument in a table lookup of the NAMEFIELD table

(67) and, if the name is already in the table, a message is typed stating that a duplicate name has been found and will be used only for editing purposes; the name will not be tested for selectional requirements, but will be edited into the specified print lines. If the name is not in the NAMEFIELD table (69), it is stored into the table; the starting and ending positions of the field and the number of digits after the decimal point are also stored into the table. The sign of the fourth word of the entry for this field in the table is made plus if the field is defined as numeric, and minus if the field is defined as alphameric. If a control break level has been assigned to the field, the NSC of this field is entered into the control break table at the specified level. If the field is to be a sort control field, the NSC is entered into the sort table at the level specified (70). At symbolic location LINESCAN, the line records are formed. The NSC from the NAMEFIELD table, the conditional requirement from the type 3 card, and the Phase 2 sort digit, 2, are set into the line record area (71). If the name of a field on this type 3 card is COUNT, DATED, or PAGE#, a special symbol code of 3, 4, or 5, respectively, is set into the line record area. If the field is to be accumulated into total lines, or if card column 26 is anything other than blank, a 0 is stored into the accumulation indicator specifying that accumulation is to be performed on this field (72).

The remainder of the type 3 card is processed in sets of seven card columns. A set will be composed of a format line number, a rightmost print position, and a group print or a dollar sign group print indicator (e.g., card columns 29-35). The first set is obtained from the work area and placed in accumulators 3, 2, and 1, respectively (73). If the format line number is not blank, the format line serial number is computed from the SERIAL table and set into the line record area (75). This is the serial number equivalent to the format line number and is not to be confused with the NSC from the NAMEFIELD table. This serial number relates the set (FLN-RPP-G/D) from the type 3 card to the proper report format line from type 2 cards. The rightmost print position and group print or dollar sign group print values from the type 3 card are set into the line record area and a line record is put on file A (76). If the field is to be group printed, the value stored in the line record group-print indicator is 7; if the dollar sign is to be group printed, the value stored is 4. If the format line number was blank at M1LSAA5+1, or after putting a line record at block 76, a test is made to determine if all the sets on the card have been checked (77). If they have not, control returns to M1LSAA5 to process the next set. After all sets from the card have been processed, control branches to NEXTTYPE2 in Phase 1 control to process the next report specifications card. When a type 3 continuation card is processed, the type 3 scan is entered at LINESCAN to process the sets (FLN-RPP-G/D) as well as the accumulation indicator (card column 25) and conditional requirement (card columns 26-27).

Type 4 Scan

DESCRIPTION, FLOW CHART AQ (CONTINUED)

The type 4 scan begins at symbolic location TYPE4EDIT and is entered in the same way as the other scans from Phase 1 control. The name is condensed (78) and the name is looked up in the NAMEFIELD table (79). If the name is in the table, control passes directly to block 81. If the name is not in the table, a message is typed stating that the selection name is not in the dictionary (80). If a name is not defined in the field dictionary, an incorrect assembly occurs. A LOGIC macro-instruction is compiled by RPG which tests for the conditions specified in this type 4 card. However, Report Program Generator will not be able to define the field since it has no knowledge of where in the input record this field is contained. When Autocoder assembles the report program, the field mentioned in the LOGIC macro-instruction will be undefined and an appropriate error message will occur in the listing.

The type 4 card is then edited from the work area into an output area (81) so that the fields are arranged in a more convenient manner in the image-4 record than in the original type 4 card. The image-4 record is prefaced by a sort control word for the Phase 2 sort. The sort control word contains a sort code of 3 and a format line serial number of 500 (82). The format line serial number of 500 is greater than any serial number which could be assigned to a format line. This will force the image-4 records to sort to the end of the file because they are not needed in Phase 3. A sequence number for image-4 records is set into the sort control word (83). The first sequence number is 002. The image-4 record is put on file A (84) and control returns to NEXTTYPE2 in Phase 1 control to process the next specifications card.

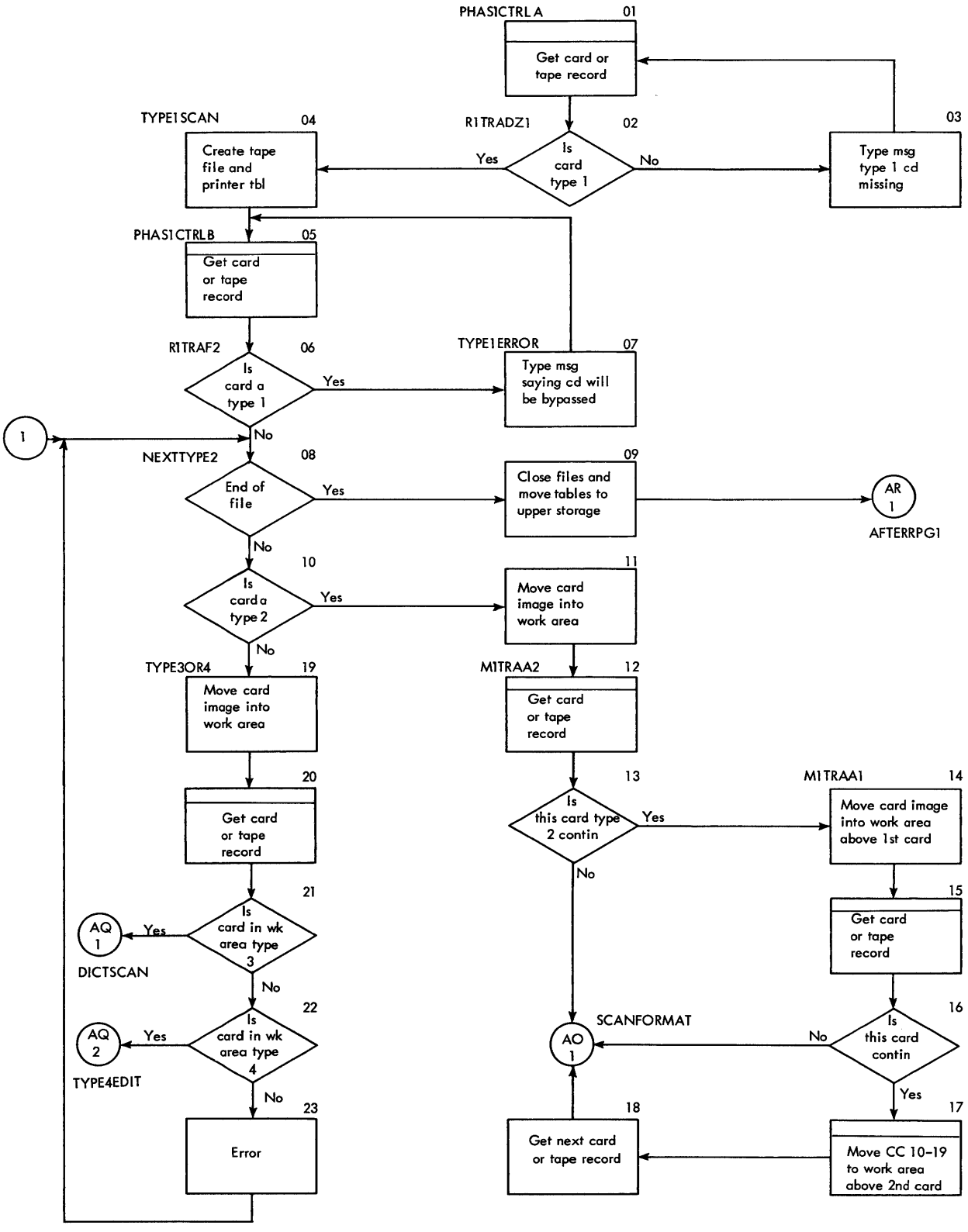


Chart AN. Phase 1 Control

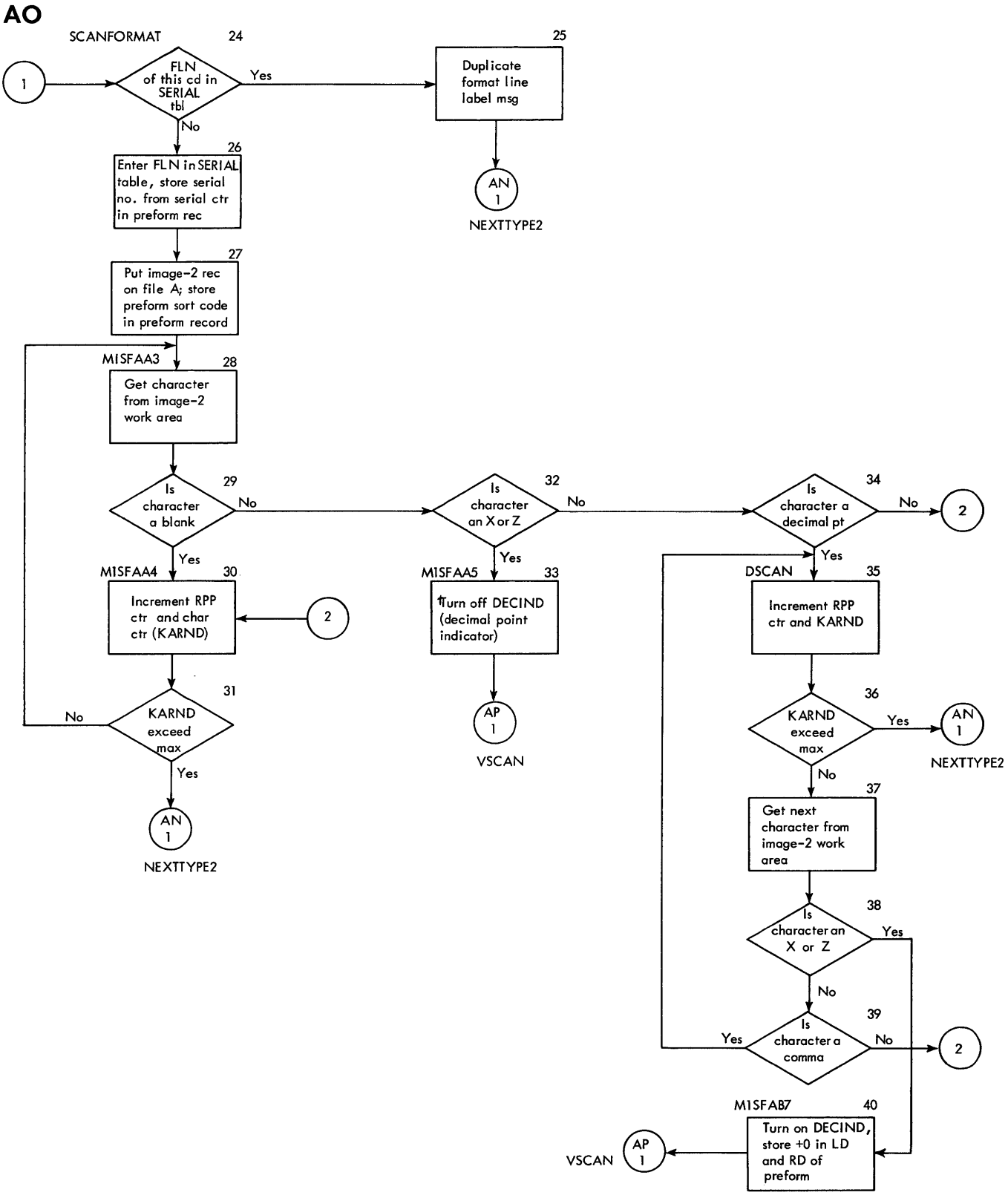


Chart AO. Type 2 Scan, Phase 1

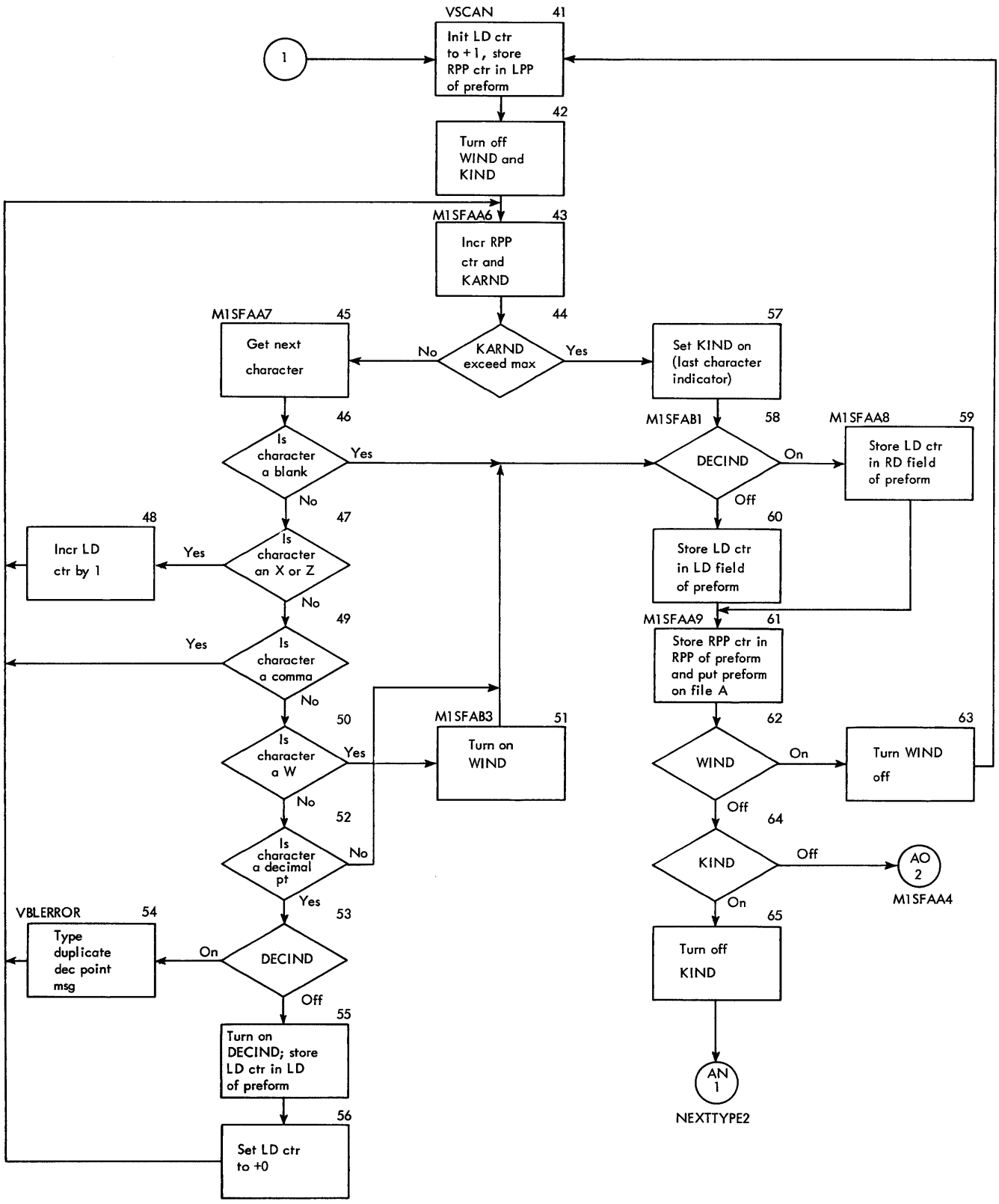


Chart AP. Variable Scan, Phase 1

AQ

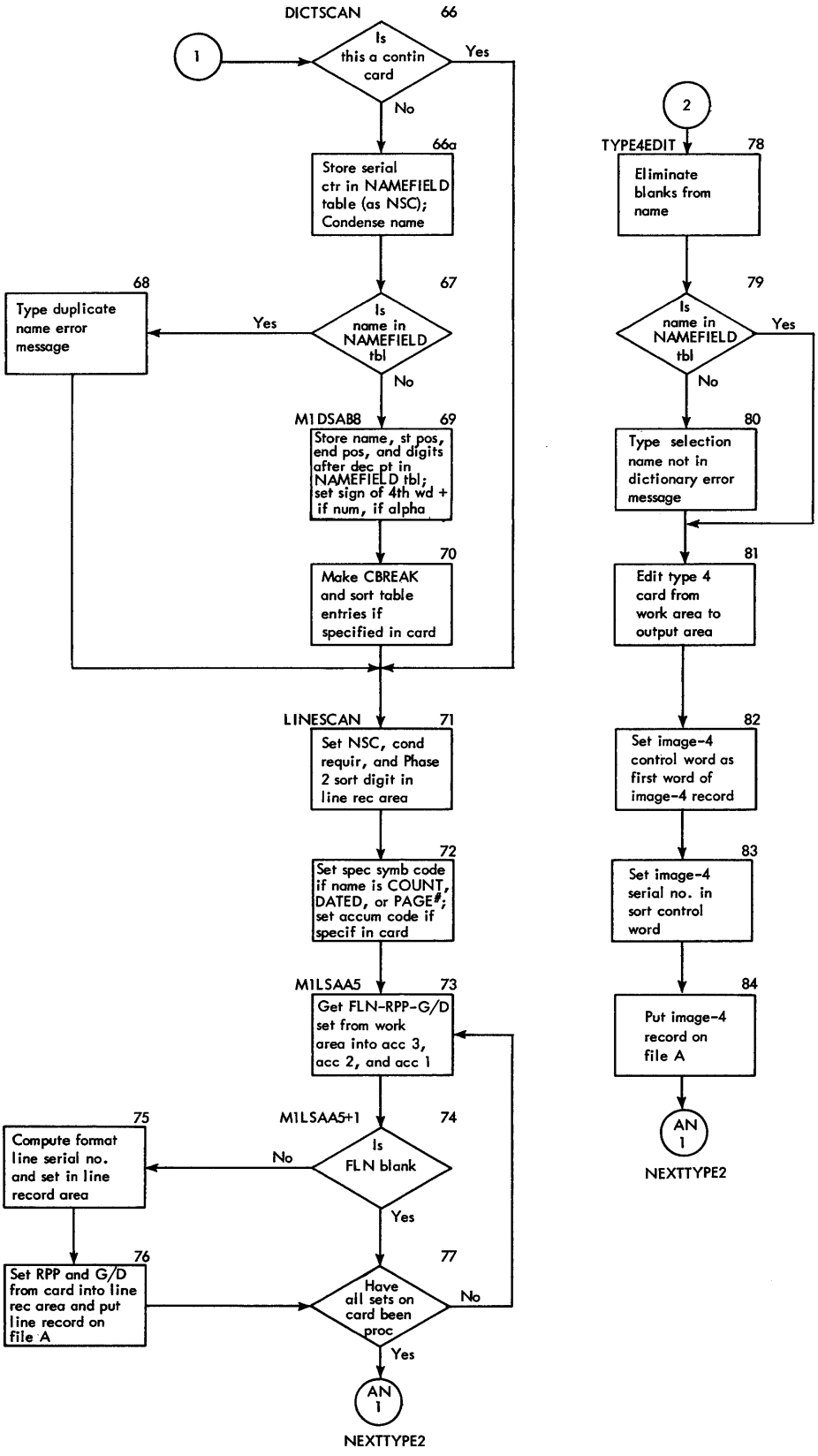


Chart AQ. Type 3 and Type 4 Scans, Phase 1

Systems Control, Section 8 (SYCL8)

DESCRIPTION, FLOW CHART AR

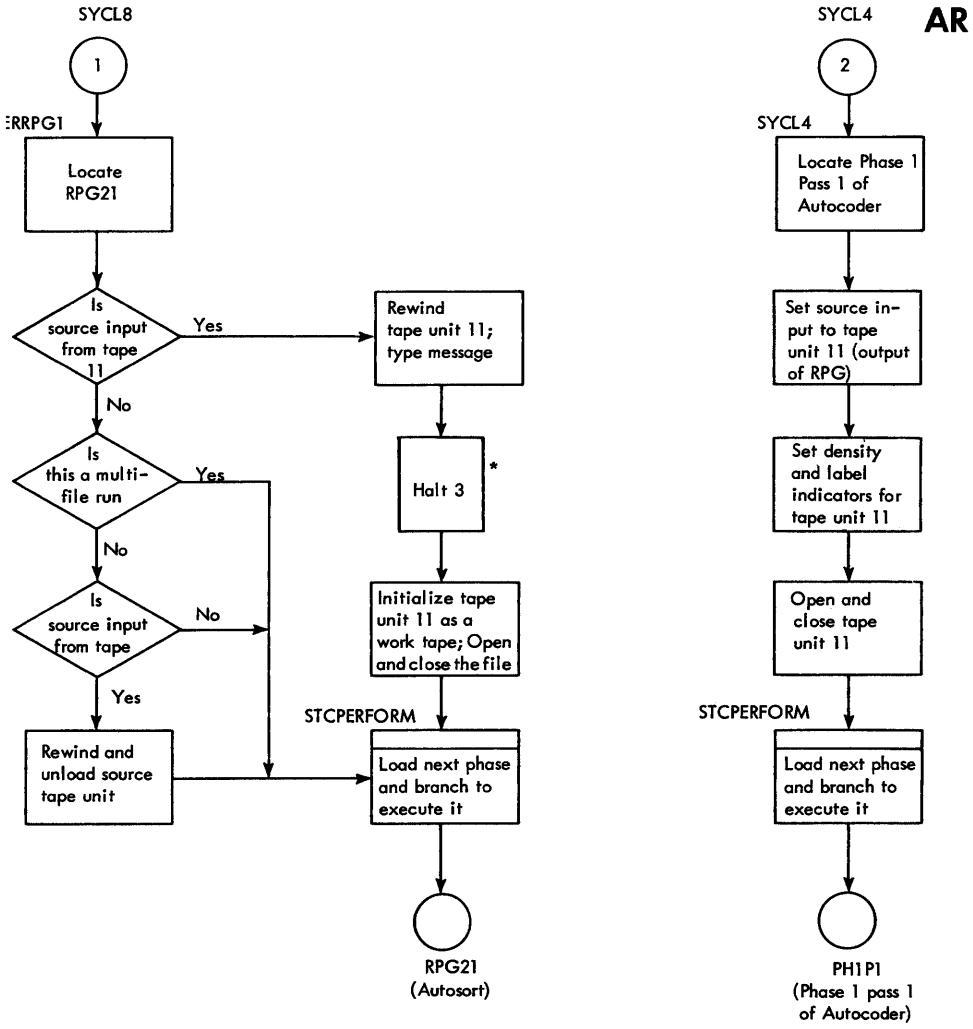
RPG utilizes two sections of Systems Control, SYCL8 and SYCL4. The function of both sections is tape housekeeping. SYCL8 is called after RPG11 and SYCL4 is called after RPG71.

At symbolic location AFTERRPG1, the search for RPG21 is initiated. A test of MAINUNIT, positions 4, 5 of location 136 in CMREC, is made to determine if the report specifications were read from tape unit 11. If they were, tape unit 11 is rewound, a message is typed, and Halt 3 follows.

If the source input is to be saved, the operator must mount a work tape on tape unit 11 and press START to continue. If the source input is not to be saved, the START key should be pressed and processing will continue. Tape unit 11 is initialized as a work tape. If the user has entered a labeled work tape, this information is set into a general purpose DTQ (the IOQ equivalent of an IOCS DTF).

Tape unit 11 is opened and closed using the general purpose DTQ. The file is opened and closed to save the user's label, if one was specified, and to write a compiler system label on the tape. (All work tapes used by RPG contain labels.) SYCL8 then branches to STCPERFORM to load and execute RPG21.

If the input specifications were not read from tape unit 11, a test is made to determine if this is a multifile run. If it is, tape unit 11 must be available because the source input cannot be read from tape unit 11 on a multifile run. Consequently, SYCL8 branches to STCPERFORM to load and execute RPG21. If it is not a multifile run, a test is made of MAINUNIT to see if the report specifications were read from the card reader. If they were, SYCL8 branches to STCPERFORM to load



* The operator must replace the tape on unit 11 to save source input; otherwise, just press START.

Chart AR. Systems Control, Section 4 and Section 8

and execute RPG21 because tape unit 11 is already available. If the source input was not read from the card reader, MAINUNIT, an extra tape unit not needed by subsequent phases, is rewound and unloaded. SYCL8 then branches to STCPERFORM to load and execute RPG21.

Phase 2

Function

A grid approach is used by RPG in the preparation of the report program from the input specifications. In Phase 1, records were formed from three types of specifications cards (types 2, 3, and 4). The information from these records is used in Phase 6 for the generation of the report program. The Phase 2 sort orders the Phase 1 records for the information transfer in Phase 3. The Phase 2 sort has three control fields:

1. The format line serial number that was assigned to the record, word 0 (0,2).
2. The rightmost print position, word 0 (3,5).
3. An assigned sort code, word 0 (6).

As each type 2 card is encountered in Phase 1, the format line number from this card is entered into the SERIAL table under the control of a serial counter. At any time, this counter contains the number of type 2 cards that have already been processed. This number becomes the serial number of the type 2 card being processed. In this way, the serial number corresponds to the format line number of the type 2 card. It is entered into the image-2 record and into the associated preform records during type 2 processing. When the line records are formed, a reference to a format line number in a type 3 card requires the serial number to be computed from the SERIAL table. This is done by looking up the format line number in the SERIAL table and then subtracting the beginning address of the table from the address of the format line number in the table. The remainder is the serial number corresponding to the format line number. This value is stored in the line record as the serial number.

The major control field groups all references to a serial number (therefore, to a format line number) together. The intermediate control field (the rightmost print position) is 000 for the image-2 record. This value causes the image-2 records to sort in front of their associated preform and line records. The preform and line records both contain a rightmost print position which was stored in the record by their respective scans. The minor sort field is a sort control digit which is 0 in the image-2 record, 1 in the preform record, and 2 in the line record. This causes the preform record to sort in front of the line records which reference the same format line number. The image-4 records formed by the type 4 scan in Phase 1 must be passed to Phase 6. For this purpose, the first sort control field is 500. This number is larger than any serial number that may be assigned because, if all possible combinations of format lines were specified, the highest serial number possible is 198. The second sort control field is an image-4 sequence number assigned during the type 4 edit in Phase 1. The minor control field is 3. Output from the Phase 2 sort is an image-2 record followed by preform and line records referring to that image-2, subsequent image-2 preform-line record combinations, and all the image-4 records. Because image-4 records are already in sequence, they are not actually sorted. The records sorted by Phase 2 are all 30 words long to accommodate an image-2 record describing 135 print positions. Because the other three kinds of tape records do not contain 30 words of information, they are padded to thirty-word length with plus zeros.

Autosort

Autosort is used to sort form 1 and 2 records only. The input and output to Autosort is in high density. Autosort utilizes a nine-word communication block located in positions 941-949. The phase that calls Autosort prepares the communication block with the sort parameters and the name of the phase to be called after completion of the sort. Autosort refers to HIGHMEMRY2 in CMREC, location 132 (6,9), to set the upper limit of storage allowed for the sort. RPG moves the tables formed in Phase 1 to upper storage and sets HIGHMEMRY2 to the value of the table area, minus one. The sort communication block contains, in 943 (4,5), the channel and unit of the file to be sorted; in 943 (6), the unit of the second tape (work tape) on the same channel; in 943 (7,9), the second channel and the two work tapes on that channel; in 944 (0,1), the input blocking factor; in 944 (2,3), the output blocking factor, and in 944 (4,5), the record length. Words 945-949 describe the control fields. (See the Appendix.) Following the sort, 943 (4,5) contains the channel and unit of the sorted tape to be used as input to the next phase. The tape unit which will contain the sorted file cannot be predicted or controlled in any way. Flow charts and a more detailed discussion of Autosort can be found in *Programming Systems Analysis Guide, IBM 7070 Series Autocoder,* Form C28-6149, in the Autosort section.

Phase 3

Phase 3 performs three functions:

1. Transfers information from image-2, preform, and line records into editor records.
2. Generates Autocoder DLINE's and DC's describing the report format lines.
3. Passes the image-4 records for Phase 6.

The information transfer of Phase 3 condenses information contained in an image-2 record and subsequent preform-line record combinations into editor records which will be used by Phase 6 to generate edit coding. The first record read in by Phase 3 is always an image-2 record. The image-2 record is followed by preform-line record combinations. If an image-2 record is not followed by a preform record, the format line contains all literal information because preform records describe the variable fields in a format line. In such a case, a special editor record, called a liteditor record, is created for this format line. When a preform record is not followed by a line record, no reference has been made in a type 3 card to the variable field described by the preform record. Consequently, no DLINE entry describing that variable field will be generated. An image-2 record is followed by a number of preform records equal to the number of variable fields in that record. A preform record is followed, in turn, by a line record each time the variable field is referred to by a type 3 card. An editor record is created for each preform-line record combination.

The editor record normally has an accumulation indicator (AI) of 7. Two editor records are created for an accumulated field in a total line; one with an AI of 2, and one with an AI of 9. An AI of 9 causes this editor record to sort to the end of the file where it is used by RPG63 for generation of the accumulation coding.

One of two types of liteditor records is for total lines, that is, for literal total lines. These records have an AI of 2. This value causes the liteditor records for total lines to sort among the editor records for accumulated fields which must be edited into total lines. The editing of fields into total lines is done in two subroutines for each level of lines. The QPT- subroutines edit accumulated totals into total lines for their respective levels and write the lines on the printer and/or on tape. They also put out the total lines which contain all literal information. The second set of subroutines, the QLT- routines, edit all non-accumulated total-line fields into their respective total lines. If more than one total line is specified on a control break level, the editor records for these lines must be together to generate the accumulate-edit coding for that level. For example, if a T21 line is all literal and the T22 line contains several accumulated fields, the editor records for the T22 line and the liteditor record for the T21 line must be processed at the same time to generate the QPT2 subroutine.

The other type of liteditor record is for heading lines, and contains an AI of 7. A liteditor record for a detail line would also have an AI of 7, but the application in which a literal detail line occurs is rare. Consequently, this application of the liteditor record is disregarded. The liteditor record is identical to the editor record except that the RPP field contains 999 instead of a print position.

To describe the report format lines, Autocoder DLINE's and DC's are generated by scanning the image-2 records. A DC is generated for report format lines which contain all literal constant information. A literal line is recognized by Phase 3 when two image-2 records occur in succession with no intervening preform records. A DLINE is generated for report format lines which contain variable fields. The image-2 records are scanned from the last character of the variable field, described by the LPP field of the previous preform record, to the variable field described by the current preform record. An unlabeled DLINE subsequent entry is generated for this constant information whether it is blank or contains alphameric information. The variable field is then scanned and a labeled DLINE entry is generated for the variable field. The image-2 record is scanned through the number of characters specified in the type 1 card (columns 68-70) which specifies the maximum number of print positions that can be used in a print line. For example, if 90 is the number of specified print positions, a counter is loaded with the value 90; after 90 characters have been scanned, the rest of the image-2 record is disregarded.

The image-4 records are used by RPG61 to generate the selection coding, that is, the LOGIC and SETSW macro-instructions necessary to perform the selection requirements specified in type 4 cards. The image-4 records must be passed on by Phase 3 to Phase 6. The control fields are padded for the Phase 4 sort to force them immediately behind the DLINE and DC records that are processed in Phase 5. Control word 1 remains the same as it was in Phase 1; that is, word 0 (0,2) contains 500, word 0 (3,5) contains a sequence number for the image-4 records, word 0 (6) contains a digit value of 3 and word 0 (7,9) contains 000. The second control word contains a major control field, word 1 (4), with the value of 1. This control field is the accumulation indicator; it forces the image-4 records to follow immediately the DC and DLINE records that contain 0 in this position. The remainder of word 1 of the record contains zeros. Word 2 of the image-4 record contains plus zeros. Words 3 through 15 of the new image-4 record are identical to words 1 through 13 of the old image-4 record. Words 16 through 18 contain plus zeros. After these editing functions are performed on the image-4 records, control branches to execute RPG41, Autosort.

The general data flow in Phase 3 is illustrated in Chart AS.

AS

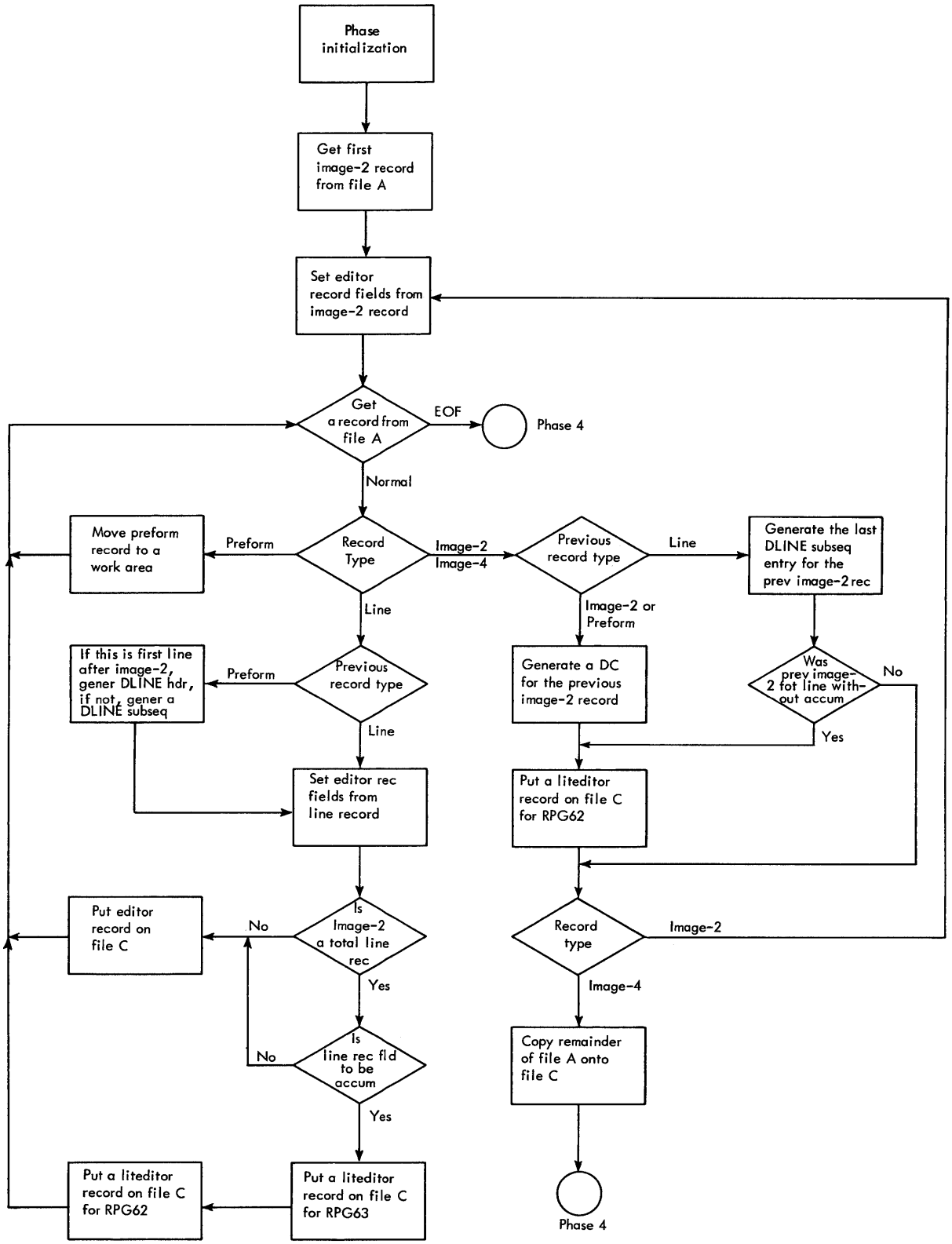


Chart AS. Over-all Processing Flow, Phase 3

Phase 3 Logic

DESCRIPTION, FLOW CHARTS AT AND AU

At R3START (01), the DLINE indicator is set off and files A and C are opened. The first record from file A is obtained and moved to the image-2 work area (02). Following the Autosort of Phase 2, this first record will be an image-2 record. At R3TRAA1 (03), the accumulate-total indicator (ATIND) is set on. The fields in the editor record that are taken from the image-2 record are set in the editor record area at this time. The print code from the image-2 record indicates if this line is to be printed on-line, off-line, or both. The print code is set into PHASECOMM2 if this print code has not appeared previously (04). The next record from file A is obtained and moved to a work area (05). When the end-of-file condition occurs, control passes to block 39, symbolic location RDEOF (Chart AU). Under normal conditions, a test is made to determine if the record in the work area is an image-2 (06) and, if so, the image-4 record indicator, IND4, is set off (09). If the record in the work area is an image-4 (07), IND4 is set on (08). In either case, control passes to R3TRAB4 to test the DLINE indicator, DLIND. If DLIND is off at this time, the record in the image-2 work area contains completely literal information so a DC is generated for this report format line (11) and control passes to R3TRAB6 (15). DLIND was set on when the first subsequent entry for a DLINE was generated after a preform-line record combination was found.

Because the current record is either an image-4 or another image-2 record, there are no variable fields (i.e., no preform records) for the record in the image-2 work area. Consequently, a DC can be generated describing the literal record in the image-2 work area. If the current input record is an image-4, only image-4 records remain to be processed. The presence of the first image-4 record means the same thing to the image-2 routines as does another image-2 record, namely: Finish the generation of the DLINE (or generate a DC) for the record in the image-2 work area, and start to process the current input record. If DLIND is on, it is turned off (12) and the last entry of the DLINE being generated for the record in the image-2 work area is compiled (13). The accumulate-total indicater, ATIND, is tested (14). ATIND is on only if the image-2 record in the image-2 work area is a total line which contains no accumulated fields. In such a case, a liteditor with an accumulation indicator of 2 is put on file C to identify this total line to RPG62 (15). The liteditor for a total line indicates to RPG62 that no accumulated fields are to be edited into that total line (i.e., no EDMOV macro-instructions are required in the QPT- routine for this format line in the report program). A liteditor record is also put out if a line contains only literal information. This liteditor record contains an accumulation indicator of 7. The image-4 indicator is tested (16). If it is not on, the record obtained from file A is an image-2 and, therefore, is moved to the image-2 work area (17). Control returns to R3TRAA1 (03) to process the new image-2 record. If IND4 is on, the record obtained from file A is an image-4 record and control branches to R3TRAB8. The first three words of each record written by Phase 3 contain control information for Phase 4 Autosort. For image-4 records, word 0 remains as it was on entry to Phase 3. Words 1 and 2 are plus zeros except position 4, word 1; this digit position is the major control field of the Phase 4 sort and has a value of 1 in image-4 records. Words 1-13 of the input image-4 record are moved to words 3-15 of the output area. The control information is placed in front of the image-4 record and the record put on file C (19). The next image-4 record is obtained from file A (20) and control returns to R3TRAB8 to continue processing the image-4 records until end-of-file is reached on file A. On end-of-file condition, control passes to RDEOF (block 39, Chart AU).

When the record was obtained from file A and moved to a work area, a test was made to determine if this record was an image-2 or an image-4. As it was neither of these, a test is made of the record to determine if it is a preform record (21). If it is, the record is moved to the preform work area and the preform indicator, PSW, is set off (22). A preform record may be followed by more than one line record, for example, when two fields are to be accumulated into a single print area. If the PSW is off, it indicates the preform record has not been processed. Thus, subsequent line records will not cause reprocessing of the preform record information. Control then returns to R3TRAA2 to obtain and process the next record. If the record is not a preform, a test is made to determine if the record in the work area is a line record (23). If not, an error message is typed stating that a specifications card cannot be identified (24); control returns to R3TRAA2 (block 05). If the record is a line record, control passes to R3TRAA8 (block 25, Chart AU).

PSW is an electronic switch that, if it is off, indicates a preform record has been read but not processed. If PSW is off (25), it is set on (26). If PSW is on, control branches to M3TRZZ1 (31). DLIND is tested to determine if this preform-line record combination is the first to be processed for the current image-2 record (27). If so, DLIND will be off; it is then turned on and the header line of the DLINE is generated (28). If DLIND is on, or after the DLINE header is generated, the preform record fields, from the preform record which set PSW off, are moved to the editor record area (29). The

image-2 record is scanned from the end of the previous variable field to the end of the variable field described by the current preform record; two DLINE subsequent entries are generated, one for the constant area between the two variable fields, and one for the current variable field.

At symbolic location M3TRZZ1 (31), the line record fields to be included in the editor record are transferred to the editor record area. A test is then made of the format line number that has been stored in the editor record area from the image-2 record (32). If it is not a total line, a non-accumulation value of 7 is set in the accumulation indicator of the editor record (33) and ATIND is set off (37). The editor record is put on file C (38) and control returns to R3TRAA2 to process the next record.

If the current image-2 is a total line (indicated by the format line number of the editor record), the AI of the line record is tested (34). The AI of a line record can have one of two possible values, 0 or 7. If the value is 7, indicating no accumulation, it is stored in the AI of the editor record (35) and the editor is put on file C (38). Control returns to R3TRAA2 to obtain and process the next record. If the value of the AI of the line record is 0, the field defined by the line record is to be accumulated. A 2 is set in the AI field of the editor record and an editor record is put on file C. This editor record is used by RPG62 for the generation of coding to edit the accumulated field into the total line. A 9 is then set in the AI field of the editor record (36) and ATIND is set off because a total-accumulate field has been processed (37). The second editor record is put on file C (38) and control returns to R3TRAA2. The second editor record is used by RPG63 to generate the accumulation coding for this field.

End-of-File Routine

When end-of-file is reached, (blocks 05 or 20, Chart AT), control branches to the end-of-file routine. If end-of-file occurs at R3TRAA2 (05), no image-4 records are to be processed, that is, no data selection requirements have been specified on type 4 cards. At RDEOF (Chart AU, block 39), a test is made of the image-4 indicator, IND4, and, if it is off, end-of-file was reached without reading an image-4 record. As there have been no image-4 records, the image-2 record routine has not been completed for the record in the image-2 work area. The presence of the first image-4 record is a signal to Phase 3 that no other image-2, preform, or line records remain to be processed.

Final housekeeping for the last image-2 record is executed either when the first image-4 is encountered or when end-of-file occurs without reading an image-4 record. The last entry for the final DLINE is generated (40). This is an entry of all blanks and/or constant literal information to make this DLINE the same length as all other DLINE's. The accumulate-total indicator, ATIND, is tested (41) and, if off, control passes to M3RDEOF. If ATIND is on, indicating the last image-2 record is a total line into which no fields are to be accumulated, a liteditor record is put on file C (42). M3RDEOF is entered after putting the liteditor record on file C, or on reaching end-of-file during image-4 processing, or if ATIND is off. At M3RDEOF, the input and output files are closed and control branches to execute Phase 4, Autosort.

In the initialization routine of Phase 3, the locations of the sort control fields for the Phase 4 sort were set into the Autosort communications area. The records written by Phase 3 are form 1 records.

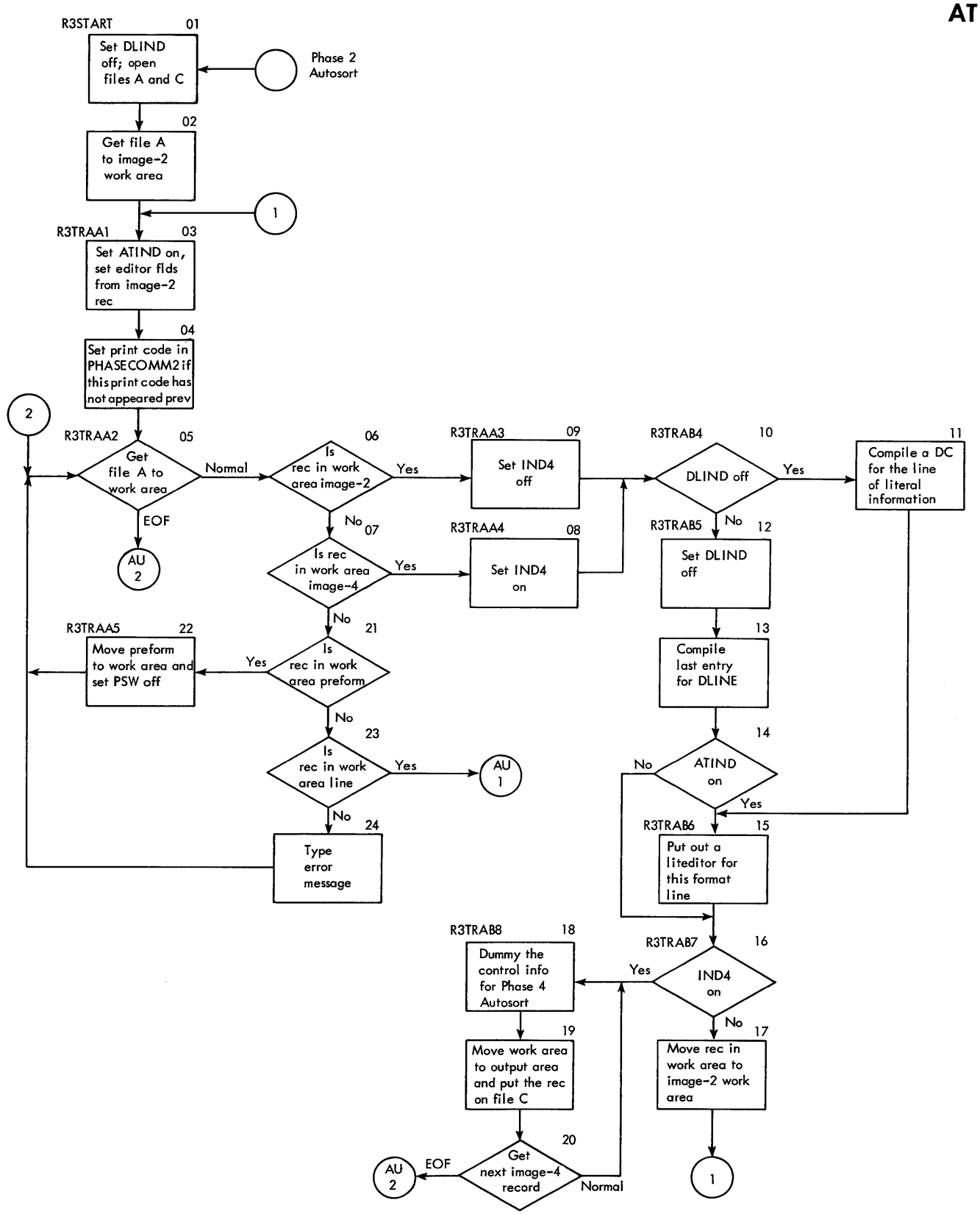


Chart AT. Phase 3

AU

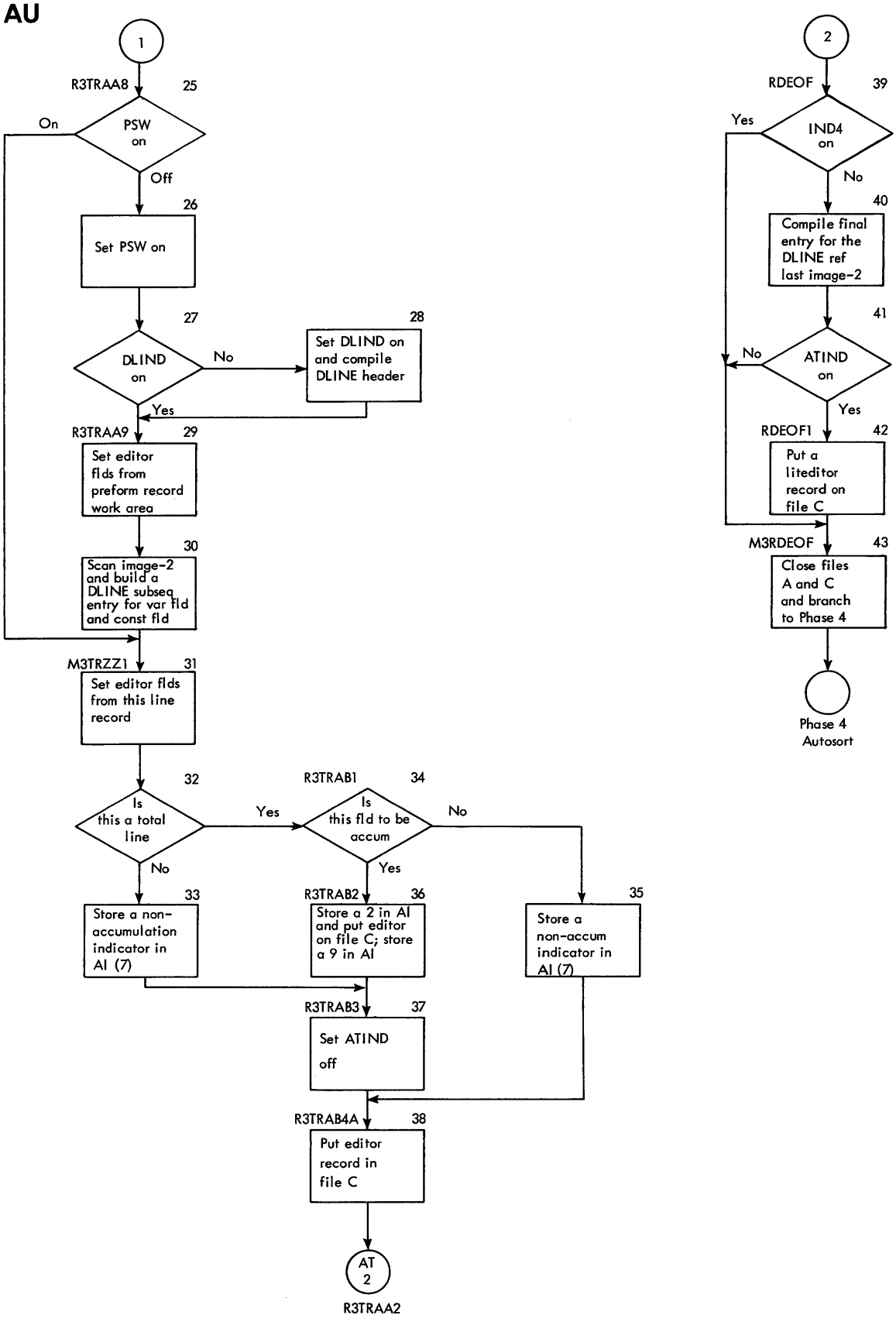


Chart AU. Phase 3 (continued)

Phase 4

Function

Phase 4 of RPG is the second sort in the processor. Autosort is used again exactly as it was for Phase 2. On completion of the sort, Autosort calls RPG51. The sort is used to order the editor records before generation of the edit and accumulation coding. The sort control fields are arranged so that routines can be generated as entities. Also, the report program can be generated without another sort; that is, after Phase 4, the DC/DLINE, image-4, editor, and liteditor records are in the correct order to generate the report program in sequence of execution. RPG51 makes the DC/DLINE records into Autocoder card image records. The image-4 records are used to generate the selection coding (the SETSW and LOGIC macro-instructions) in RPG61. The editor and liteditor records govern the generation of edit and accumulation coding in RPG62 and RPG63, respectively. This is the sequence in which these routines appear in the generated report program (except the DC/DLINE's which are declarative entries and are not significant to the sequence of instructions).

Description of Control Fields

The Phase 4 sort has four control fields:

1. Accumulation indicator, word 1 (2).
2. Format line number, word 0 (0,5).
3. Group print indicator, word 1 (5).
4. Conditional requirement, double digit, word 2 (0,3).

The first three control fields in the DC/DLINE records are plus zero. The fourth control field contains a DC/DLINE serial number. This arrangement sorts the DC/DLINE records to the front of file C where they are used by RPG51. Because the image-4 records are the input to RPG61, the control fields for the image-4's are padded to cause them to sort immediately behind the DC/DLINE records. The accumulation indicator is 1, a value less than for any other record type except the DC/DLINE's. The value of the accumulation indicator has significance, in the editor and liteditor records, to the generation of edit and/or accumulation coding. The value 1 in the primary control field forces the image-4 records immediately behind the DC/DLINE records. The format line number was created in Phase 1; it contains 500 and an image-4 sequence number. The two minor control fields are set to plus zero.

The first routines generated by RPG62 are: (1) the editing of accumulated fields into total lines, and (2) the writing of these lines. The accumulation indicators for the records used to generate the accumulation edit coding (QPT- routines) contain the value 2. All total lines must have at least one record with an accumulation indicator of 2. If no fields are to be accumulated in a total line, a liteditor record with an AI of 2 was created for that total line. This liteditor record indicates that no accumulation edit coding is to be generated for that line, but that put-out coding must be generated. The accumulation indicator value that governs all other edit and put-out coding is 7. The records with an AI of 9 sort to the end of file C and are used by RPG63 to generate the accumulation coding. The editor and liteditor records contain a valid (nonpadded) intermediate sort control field, the format line number. Sorting on format line number groups all references to a format line allowing the edit coding or accumulate-edit and put-out coding to be generated in line. The value of the group print indicator is 0 if a field is not to be group printed, 4 if the dollar sign is to be group printed, 7 if the field is to be group printed. The minor sort control field groups all references to a conditional requirement together; thus, only one test of a conditional requirement is necessary for each format line, unless a field or dollar sign is to be group printed in the line. If it is, the conditional requirement is tested once for each grouping. Phase 4 sorts the records in the order shown (Figure 2).

RPG51 processes the DC/DLINE records, and, upon recognizing the first image-4 record, exits to RPG61. Similarly, when RPG61 recognizes the first editor or liteditor record, it exits to RPG62. RPG62 in turn exits to RPG63, after a liteditor record with an AI of 9 is found.

For an explanation of how Autosort functions in relation to RPG, see "Phase 2." The input to Phase 4 is on the same tape unit as the input to Phase 2. The record length is 19 words and the blocking factor is ten. The records are form 1. The name of the following phase, RPG51, is stored in the Autosort communication block so that it will be called upon completion of the sort.

Record	AI	Format Line No.		GI	Cond Req	Used By
DC/DLINE	0	000	000	0	0000	RPG51
Image-4	1	500	Seq no.	0	0000	RPG61
Editor and liteditor	2	T		*	***	RPG61
Editor and liteditor	7	**		*	***	RPG62
Liteditor	9	T		*	***	RPG63

* 0 = no group printing, 4 = group print $, 7 = group print field
** D, H, or T
*** Any conditional requirement specified in a type 4 card

Figure 2. Phase 4, Sort Control Fields

Phase 5

Phase 5 of the processor has three responsibilities:

1. Generate a presort program.
2. Generate the DTF's and I-O areas for the report program.
3. Edit the DC/DLINE records into Autocoder card images.

The primary function of Phase 5 is the generation of a presort program if one is required. A presort program will be generated if card column 31 of the type 1 card contains a Y. At object time, the presort program passes the input file from which the report is to be prepared and creates a shorter, intermediate file. The intermediate file is then sorted (by Sort 90) and is used as input to the report program. The presort program is generated entirely by Phase 5 whereas the report program is generated by Phase 5 and the three loads of Phase 6.

The DC/DLINE records were generated as Autocoder card images preceded by three words of sort control information for Phase 4. The editing function performed by Phase 5 deletes the three words of control information and writes the DC/DLINE records on file B as Autocoder card image records. The DC/DLINE records are the only tape records processed by Phase 5. Generation of the presort program, the DTF's, and I-O areas for the report program is performed by analyzing the tables created in Phase 1 and four additional tables created in Phase 5. Figure 3 illustrates the routines generated by Phase 5. If card column 31 of the type 1 card contains an N, a presort program is not generated and Phase 5 becomes a short phase.

Routines or Declaratives Generated	Tables Used	Phase Which Created Tables	Output File
Presort Program			
Sort parameters	SORT and TFPTBL	RPG11	E
DIOCS	- - -	- - -	E
Input data file DTF	TFPTBL	RPG11	E
Label DC for input file	TFPTBL	RPG11	E
Intermediate file DTF; label DC for inter file	TFPTBL	RPG11	E
Input area DA for data file. Intermediate file output DA	NAMEFIELD AND TFPTBL	RPG11	E
Presort program instructions(i.e., the OPEN, PUT, and CLOSE instructions)	NAMEFIELD COMPRESS MOVE	RPG11 RPG11 RPG11	E E E
Report Program			
DTF for input file to report program	TFPTBL	RPG11	B
DTF for report program output file	NAMEFIELD AND TFPTBL	RPG11	B
Input area DA for report program	- - -	- - -	B
Output area DA for report program	- - -	- - -	B

Figure 3. Generation Responsibilities, Phase 5

DESCRIPTION, FLOW CHART AV

Phase 5 first tests the RPGIND in CMREC (01); a 1 indicates a sort has been specified and a presort program is required. The presort program is relatively simple and the format is very rigid. In order to generate the MOVE macro-instructions which move the named fields from the input file to the intermediate file, the field definitions must be ordered. On the type 3 cards, the fields need not be specified in the order in which they occur in the tape record. Thus, the fields must first be sorted on their beginning addresses. Then full words which separate fields can be eliminated. If the relative field definition of two adjacent fields does not overlap, the fields will end and start in the same word in the intermediate file.

During the type 3 scan in Phase 1, the NAMEFIELD table was built that described the beginning and ending positions of a field in the input record, the name of the field, the number of positions after the decimal point, and the characteristic of the field as numeric or alphameric. (Note: Phase 5 refers to this table as the NA table although it is the NAMEFIELD table created by Phase 1.) A list of the beginning and end positions of each named field is constructed and then sorted by beginning positions (02). A small internal sort performs this function (not Autosort which is used as Phases 2 and 4). From this sorted list, the COMPRESS table is built (03) in which the named fields maintain their relative field definition. For example, assume that three entries in the NAMEFIELD table are:

```
A P R O D U
A C T     N O
+ 0 1 0 0 0 4 0 0 1 1
+ 0 0

A S A L E S
A M A N
+ 0 2 0 0 9 8 0 1 1 7
- 0 0

A C U S T O
A M E R
+ 0 3 0 0 5 8 0 0 7 9
- 0 0
```

The third word of each entry is listed:

+01 0004 0011
+02 0098 0117
+03 0058 0079

the list is sorted on the beginning position of each field:

+01 0004 0011
+03 0058 0079
+02 0098 0117

and the COMPRESS table is built:

+01 0004 0011
+03 0018 0039
+02 0048 0067

NOTE: The relative field definitions of each field have been maintained. The new field definition will be used to generate the input DA for the report program. The MOVE table is built from the original field definitions in the NAMEFIELD table, defining fields to the nearest full word (04). Because an instruction with the format MOVE Q01, Q02, Q03 TO QA01 is to be generated, the original field definitions used as operands would distort the field definitions as MOVE places the first character of Q02 adjacent to the last character of Q01. Consequently, the MOVE operands are built from field definitions in the MOVE table. At the same time the MOVE table is being built, a count of the number of digit positions being moved is kept in the SUM table and will be used in defining the record length in the intermediate file.

The COMPRESS table does not exist as a unique table in storage, but is composed of the fourth word of each entry in the NAMEFIELD table. The MOVE and SUM tables are built in the output areas of Phase 5 and are destroyed before the end of the phase.

The MOVE table would be:

+0000000017
+0000580079
+0000900117

The SUM table would be:

+0000000067

The MOVE macro-instructions generated will have a maximum of four operands. As many MOVE's as necessary to move all the named fields in the input file to the intermediate file will be generated. The SORT90 parameters are generated from a DC in which they are contained as constants, and from the TFPTBL which defines input, output, and work tape units (05).

The DIOCS for the presort program is generated (05) in the form:

DIOCS,,,CHANN,OPEN2,EOR1,,IGEN4

This entry causes generation of IOCS with the following options:

1. The three index words required by IOCS are to be assigned by Autocoder (IOCSIXF, IOCSIXG, IOCSIXH).
2. Routines are needed for N channels: one or two.
3. The OPEN routine is retained in storage for use whenever needed.
4. The label area address for each file is contained in its respective DTF.
5. The checkpoint routine is not required.
6. SPOOL routines cannot be run and the tape error routine will not check for illegal double digit characters.

The input file and output file DTF's are generated (06) and the input and output areas for the presort are generated (07). The output DA (the output areas for the intermediate file) is constructed so that it is approximately 200 words long. The value of 200 words per block is chosen as an optimum record length for Sort 90. The presort program is generated from DC's and from the MOVE table (08).

If a sort was not specified, the remaining functions would be the only ones performed. The input DTF (09) and the output DTF (10) for the report program are generated. The input DA (input areas for the sorted intermediate file) is generated (11) from an analysis of the NAMEFIELD table. If a presort has been generated, the fourth word of each entry in the table is used to define the named fields; if a presort has not been generated, the third word of each entry is used to define the named fields in the input DA. The output DA (areas for the off-line output of the report program) is generated (12).

The first three words of each DC/DLINE record are deleted and the record written immediately onto file B without further processing. The DC/DLINE is then a true Autocoder card image. After all the DC/DLINE records have been edited, control passes to Phase 6, load 1 (Chart AW).

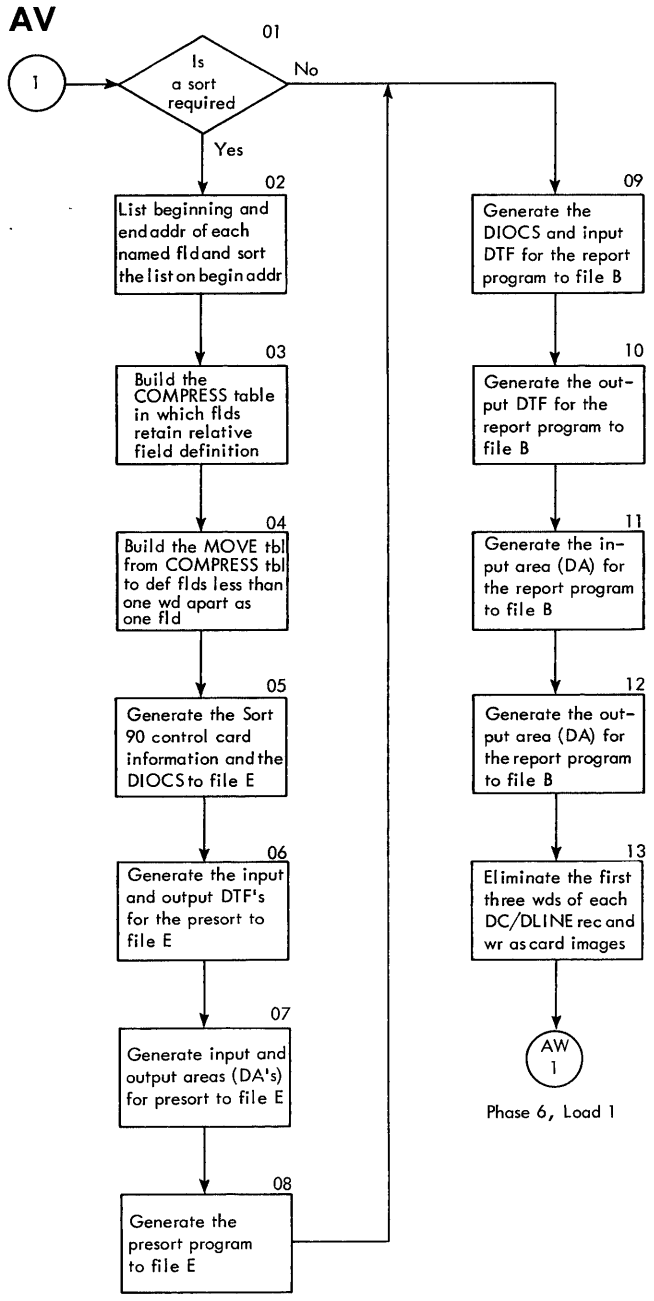


Chart AV. Phase 5

Phase 6

Phase 6 of RPG is the primary generation phase. Only the DC/DLINE's, the input-output areas, and the DTF's for the report program are generated outside Phase 6. The report program and any declaratives necessary (such as a DA for the accumulation areas) are generated by Phase 6.

Phase 6 consists of three loads, RPG61, RPG62, and RPG63. Phase 5 and the three loads of Phase 6 can be considered as one phase with four loads because the input-output areas, work areas, and DTF's for each load are identical and are located in the same storage locations; also, several utility subroutines are used by every load. The tape files are not closed between loads until end-of-file occurs on input file C in Phase 6.

The generation responsibilities of the three loads of Phase 6 are illustrated by the table in Figure 4. Flow Chart AW illustrates the over-all logic of Phase 6.

RPG61

The type 4 cards were edited into image-4 records by Phase 1; the image-4 records were slightly rearranged by Phase 3. As there is only one input file to Phases 5 and 6, when Phase 5 reads an image-4 record, RPG61 is called. When RPG61 is entered, the information necessary to generate the first LOGIC macro-instruction for selection is in storage. A table, CONDTABLE, is created and maintained by RPG61 to record the presence of each resultant conditional requirement specified in the type 4 cards (columns 54-55). When a new conditional requirement is encountered, it is entered into CONDTABLE and a SETSW macro-instruction is generated which turns the switch off before it is tested in the report program.

A LOGIC macro-instruction is generated for each image-4 record. The control break test routines are then generated by examining the CBREAK table. A DA that contains the control break test areas is generated and put onto file B. If a control break level is determined by more than one field from the input record, an area is generated into which the fields are moved before they are tested for a control break. The actual control break routine is generated from a series of DC's that contain the control break instructions as constants.

The total and heading line control branches are two lists of nine branch instructions following the labeled instructions: QTOTAL B QSK and QHDR B QSK. Each of the nine branch instructions corresponds to a control break level. If a heading or total line for a control break level is present in the SERIAL table, a B QPT- or B QHP- is generated at the proper location in the list

of branches for that level. If no total or heading line is present on a control break level, a B 0+x94 instruction is generated. For example, if eight control levels are specified, and total lines are to be printed on five levels and heading lines on seven levels, the following instructions would be generated (assuming no total lines are to be printed on levels 2, 4, and 5, and no heading lines on level 2):

```
QTOTAL B QSK                QHDR B QSK
       B QPT1                      B QHP1
       B 0+x94                     B 0+x94
       B QPT3                      B QHP3
       B 0+x94                     B QHP4
       B 0+x94                     B QHP5
       B QPT6                      B QHP6
       B QPT7                      B QHP7
       B QPT8                      B QHP8
       B 0+x94                     B 0+x94
```

The control break move branches (starting at QCBMOV in the report program), which govern the moving of control break fields to the control break test areas after a control break, have the same format as the total and heading line control branches. They are generated by examining the CBREAK table; if a control break is present at level 2, for example, a B QMOV2 is generated, and if no control break is present at level 2, a B 0+x94 is generated. The actual move routines (QMOV1 through QMOV9) are also generated from an analysis of the CBREAK table.

The final routine generated by RPG61 is the end-of-file routine. RPG62 is called when the first editor record is read from file C.

RPG62

RPG62 generates all edit and put-out coding from editor and liteditor records. The first edit routines generated are for the total lines. These routines, labeled QPT1 through QPT9, move all accumulated fields into their respective total lines and put the total lines out on tape, on-line, or both, as specified in card column 6 of each total line type 2 card.

The detail line editing proceeds with an examination of the first detail line. If card column 7 contains a digit value, that number of blank lines will be put out in the generated report program after a control break (i.e., after all total and heading lines for the control break have been put out). A blank line, QSKPA, is generated to file B with a spacing code adequate to "skip" the correct number of lines. If the number of lines to be skipped is greater than three, a second line, QSKPB, is generated. A maximum of three

Routine Generated	Symbolic Location in Report Program From	To	Tape Records Used	Phase *	Table Used	Phase *	Output File	Phase
Program initialization	QSTART	QGETREC+2					A	RPG61
Data selection	QGETREC+2	QCN99+1	image-4	1	CONDTABLE	61	A	RPG61
Define selection switches							B	RPG61
Ctrl brk testing routine	QCOMP-	QCBEND			CBREAK	1	A	RPG61
Ctrl brk declaratives	QCBAREA	QTES-			CBREAK		B	RPG61
Ctrl brk routine	QCBEND	QTOTAL					A	RPG61
Total line control	QTOTAL	QHDR			SERIAL	1	A	RPG61
Heading line control	QHDR	QCBMOV			SERIAL	1	A	RPG61
Move ctrl brk flds	QCBMOV	QMOV1			CBREAK	1	A	RPG61
	QMOV1	QEOF			CBREAK	1	A	RPG61
Ctrl brk test areas	QTES1	QTES-			CBREAK		B	RPG61
End-of-file routine	QEOF	QEOF+11					A	RPG61
Edit final tot lines	QPTF	QPT1	Liteditor & editor	3			A	RPG62
Edit tot lines	QPT1	QSK	Liteditor & editor	3			A	RPG62
Edit detail lines	QSK	QFRS	Liteditor & editor	3			A	RPG62
Blank lines for skipping	QSKPA	QSKPC	D01 editor	3			B	RPG62
Edit one-time headings	QFRS	QHP	Liteditor & editor	3			A	RPG62
Edit heading lines	QHP	QLTF	Liteditor & editor	3			A	RPG62
Total line initialization	QLTF	QFCON	Editor	3			A	RPG62
Prog initialization	QFCON	QSET					A	RPG62
Define PAGE area	QPAGE						B	RPG62
Init prog switches	QSET	QPT					A	RPG62
Line counting routine	QPT	QACCUMULAT					A	RPG62
Accumulation routine	QACCUMULAT	QLT-ACC01	Liteditor	3			A	RPG63
Accumulation areas	QACAREA						B	RPG63
End ctrl stmnt **	END						A	RPG63

* Phase which created the tape record or table.
** This statement will be generated by RPG62 if no accumulation is specified.

Figure 4. Generation Responsibilities, Phase 6

lines, each with triple spacing, will be generated to "skip" the specified number of lines.

The detail line fields to be printed in a line under the control of the same conditional requirement are generated as operands of the same EDMOV macro-instruction; if there are more than five such fields, a second EDMOV macro-instruction is generated, and so on. In this way, unless fields are to be group printed, only one test of a conditional requirement need be made before the fields associated with that requirement are edited into a line. Because one of the sort fields of the editor records was the conditional requirement, a test of a conditional requirement is generated when a new requirement is encountered and when a new format line is processed.

After all editor records have been processed, or when end-of-file occurs on file C, the QLT- routines are generated from an analysis of the SERIAL table. If one or more total lines are present on a control break level, coding is generated to zero the accumulation areas for that level of lines. The first-time routine, QFCON, is generated to initialize the accumulation areas for total lines; if line counting has been specified in card columns 66 and 67 of the type 1 card, the value assigned in these columns is set into a generated statement to initialize the line counter. The switch initialization routine, QSET, is generated and the page-turning routine, QPT, is generated if line counting has been specified. Control then passes to RPG63 if end-of-file on file C has *not* been reached. If end-of-file on file C has been reached, an END CNTRL QSTART statement is generated. Control in this case passes to RPG71.

RPG63

RPG63 generates the accumulation routines from the accumulation liteditor records. An ARITH macro-instruction is generated for every field to be accumulated and for every line in which these fields will be printed. If, for example, a field can appear in four levels of total lines and if the field is to be accumulated, four ARITH macro-instructions will be generated that accumulate the field in a different area for each total line. The definitions of accumulation areas are generated to file B by RPG63.

When end-of-file occurs on file C, an END CNTRL QSTART statement is generated to file A. All files are closed and RPG63 exits to RPG71.

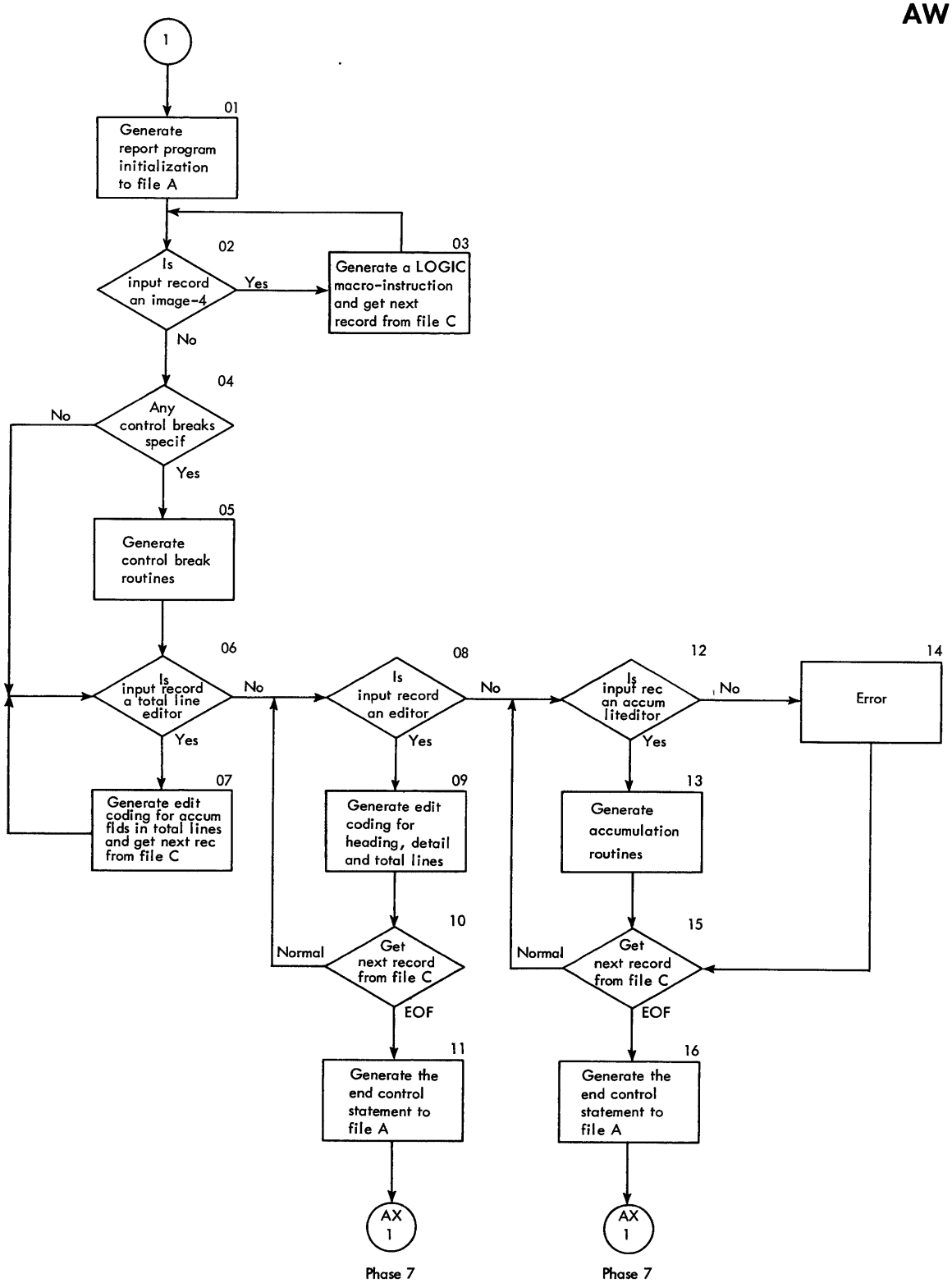


Chart AW. Phase 6

Phase 7

Phase 7 is the simplest and shortest phase of RPG. Its function is to write the generated report program, which is on files A and B, onto one file and exit to SYCL4 to call Autocoder. RPG71 is approximately two hundred instructions long and, when the I-O areas and constant areas are included, occupies about 1,100 machine locations.

When RPG71 is entered, file A contains the generated report program in the order in which it will be executed. File B contains all the declaratives for the generated report program (i.e., the DA's, DC's, DLINE's, DTF's and DRDW's). A logical order is not required; the DTF's, however, are in sequential order as dictated by IOCS. File E contains the complete presort program, instructions, and declaratives, in high density, if a sort was specified.

If a presort program has not been generated, file A and then file B are written onto file E. Simultaneously, the report program is written onto file C. File E is the input to Autocoder. The copy of the report program on file C is in low density and is rewound and unloaded. This copy may be punched and modifications may be inserted. If the copy is listed (printed), this

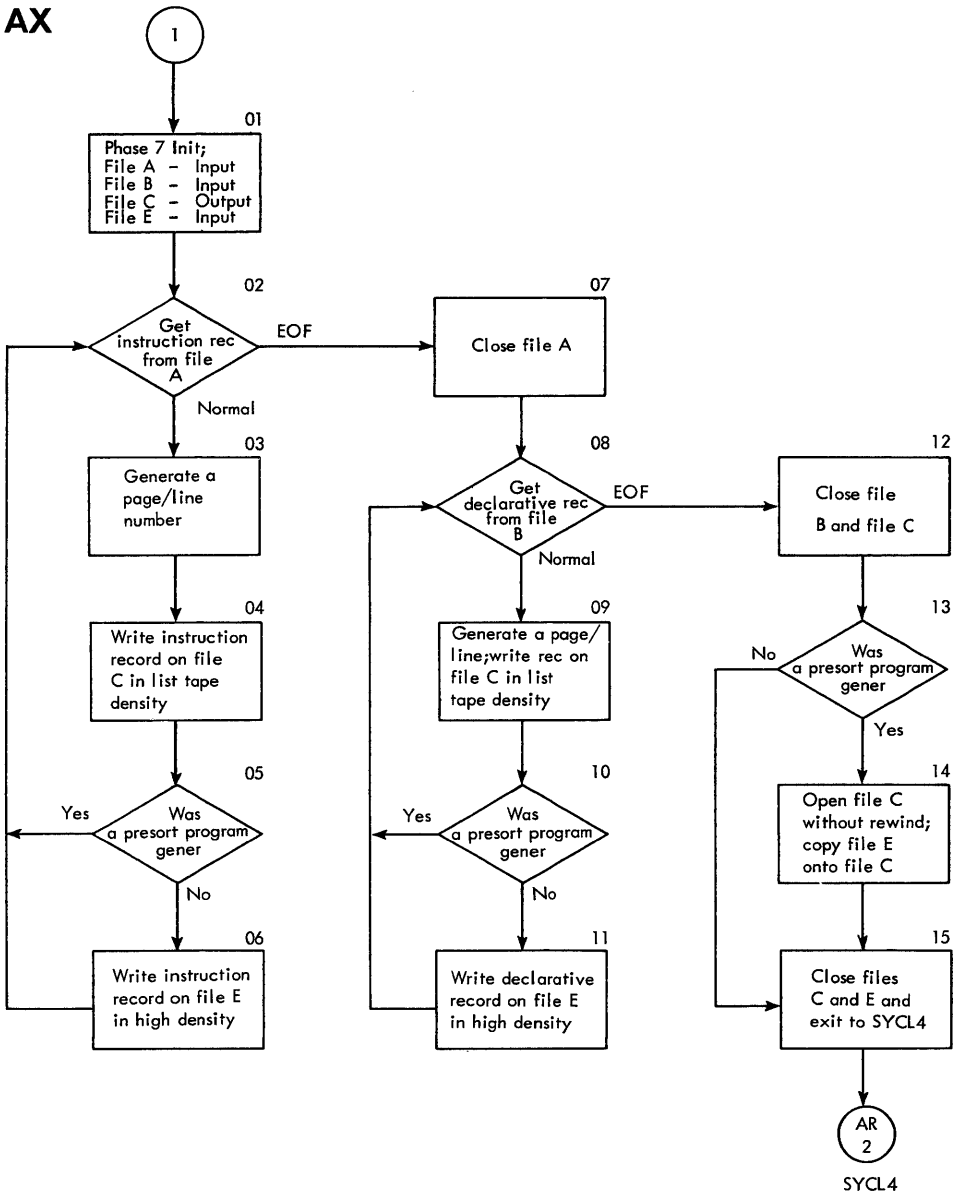


Chart AX. Phase 7

listing is a valuable aid in debugging the report specifications.

If a presort program has been generated by Phase 5, it is on file E when RPG71 is entered. The presort program is the input to the Autocoder section of this run. Consequently, files A and B are written onto file C in low density, and file C is closed. The CLOSE routine writes a tape mark on file C and does not rewind it. File C is then opened, without rewinding, and file E is copied onto file C in low density. File C must be used on a subsequent Autocoder run to compile the generated report program. The presort program on file E is compiled by Autocoder at this time.

Following RPG71, the two output files are on tape units 11 and 12. Tape unit 11 contains file E, the input to Autocoder. Tape unit 12 contains the copy of the report program and the copy of the presort program, if a sort was specified.

Chart AX illustrates the manner in which the function of Phase 7 is accomplished. One subroutine not in the main line of coding is used. PAGELINE inserts a sequential page/line number in the card record output area that corresponds to card columns 1-4. Phase 7 calls SYCL4, and passes control to it.

Systems Control, Section 4 (SYCL4)

SYCL4 is executed after RPG71 has processed and produced the output of RPG. SYCL4 is a section of Systems Control; it is contained in the same coding block as sections 4-10 (SYCL4-SYC10). These sections of Systems Control are contained in the same coding block for two reasons: first, all the sections are small and are easily contained in one block; second, although the entrance to each section is unique and depends on the preceding phase, the sections often use common subroutines. The function of SYCL4 is to initialize the output from RPG71 as input to the Autocoder portion of this run.

At symbolic location SYCL4 a search is initiated for Phase 1, pass 1 (PH1P1) of Autocoder. The contents of MAINUNIT, word 137 (4,5), are stored in a temporary location in CMREC. MAINUNIT is then set with the value 11 which is the high density output tape from RPG. MAINUNIT is restored to its original value after Phase 1, pass 1 of Autocoder. The density and label indicators for tape unit 11 are set into a general purpose DTQ and the file is opened using this DTQ. Control then branches to STCPERFORM to load and execute PH1P1. (See Chart AR.)

Appendix

Interphase and Autosort Communication Areas

Five words that are not destroyed upon successive phase loadings are available in 155-159. RPG uses four of these words:

PHASECOMM1: (0, 9) contains the beginning location of the tables which are moved to upper storage at the conclusion of each phase.

PHASECOMM2: (0, 4) unused
(5, 9) a digit 1, 2, or 3 indicating to RPG62 that a blank HPl line must be generated which will print on-line (1), off-line (2), or both (3), and will contain a carriage control character of 1.

PHASECOMM3: (0, 0) a 1 indicates a detail line has been specified
(1, 3) unused
(4, 5) a count of the number of format lines specified
(6, 9) unused.

PHASECOMM4: (0, 9) unused.

PHASECOMM5: (0, 9) program identification from card columns 76-80 of the type 1 card.

The Autosort communications block is set up by the phase which calls it. Note that the name of the phase following Autosort is set into the communication block and is called by Autosort. The format of the block is as follows:

NEXTLOAD	941	(0, 9)	name of phase following
	942	(0, 9)	Autosort
OPTIONS	943	(0)	not used by RPG
		(1, 3)	unused
INCHAN		(4)	channel of input tape (contains the channel of sorted tape after the sort)
INDRONE		(5)	unit of input tape (contains the unit of sorted tape after the sort)
INDRTWO		(6)	unit of second tape on input channel
OUTCHAN		(7)	second channel
OUTDRONE		(8)	unit of first tape on second channel
OUTDRTWO		(9)	unit of second tape on second channel
INBLOCK	944	(0, 1)	input blocking factor
OUTBLOCK		(2, 3)	output blocking factor
RECLTH		(4, 5)	input record length in words
		(6, 9)	unused
	945	(0, 1)	unused
CTLFLDONE		(2, 5)	location of first control field
CTLFLDTWO		(6, 9)	location of second control field
	946	(0, 1)	unused
CTLFLDTHRE		(2, 5)	third control field (0000 for Phase 2)
CTLFLDFOUR		(6, 9)	fourth control field (0000 for Phase 2)
	947	(0, 9)	these words are available for the additional control fields but are unused by RPG
	948	(0, 9)	
	949	(0, 9	

Storage Maps

The following section contains storage maps (Figures 5-13) for all the phases of RPG and for the generated report and presort programs.

Tables

Figures 14-16 illustrate the major tables used in RPG.

Location	Function
QSTART to QGETREC	Program initialization
QGETREC to QCOMP-*	Get data record and perform data selection
QCOMP- to QCBEND	Control break testing
QCBEND to QTOTAL	Control break routine
QTOTAL to QHDR	Total line control branches
QHDR to QCBMOV	Heading line control branches
QCBMOV to QMOV1	Ctrl branches to move new ctrl brk fields to test areas
QMOV1 to QEOF	Move new control break fields to control break test areas
QEOF to QPTF	End-of-file routine
QPTF to QPT1	Edit and put out final total lines
QPT1 to QSK	Edit accum areas into total lines and put out
QSK to QFRS	Edit selected fields into detail lines and put out
QFRS to QHP	Edit and put out report heading lines
QHP to QLTF	Edit selected fields into page heading lines and put out
QLTF to QLT1	Initialize accum areas for final total lines to zero
QLT1 to QFCON	Zero accum areas and edit non-accum fields into total lines
QFCON to QSET	Initialize page control and line control and branch to all QLT- routines
QSET to QPT	Set program control switches QPSW, QGGW, QDLW, and QSKW
QPT to QACCUMULAT	Increment line control and check for maximum; reset page control and line control
QACCUMULAT to QLT--ACC01*	Accumulation routine
IOC.MASK to IOC.GLAST	IOCS
TAPEFILEA to SCHEDINFD+2	DTF's for IOCS
QDC4 to QHF1	Label DC's and DA's for I-O areas (for IOCS)
QHF1 to QTF9*	Line images (DC's and DLINE's)
	Various declaratives required for the running program
QACAREA	Accum areas (one DA with subsequent entries)

* The numeric portions of these labels depend on the report specifications

Figure 5. Generated Report Program, Storage Map

Location	Function
369 to 400	Generated parameters for Sort 90
401 to IOC.GLAST	IOCS
TAPEFILEA to QDC1	DTF's for input file and output file (intermediate file)
QDC1 to QDA1	DC containing input header information
QDA1to QDA2	Input file DA
QDA2 to QSTART	Intermediate file (output) DA
QSTART to QGET	Open files
QGET to QEOF	Get input record; move fields to output file; put the output file
QEOF to QEOF+6	Close input and output files; halt; branch to load program (308)
	Generated subroutines and literals

Figure 6. Generated Presort Program, Storage Map

Location	Function
	CMREC, STC, IOQ, Autosort communication, DTQ's (files A and D), I-O areas
RPGSC1	Locate next memory load; assign tape unit numbers for Phase 1
PHAS1CNTRL	Open files; read and identify first card
PHAS1CTRLB	Read and identify next card
M1GETCARD	Subroutine to get next card from tape or reader
EOFILED & ENDOFFILE	End-of-file routines for files A and D
TYPE1SCAN	Build TFPTBL
NEXTTYPE2	Check input card for type 2 and, if so, check for continuation cards
ORDERR	Error message type out routines (including the messages)
SCANFORMAT	Put out a preform record if the type 2 card does contain variables (i.e., X or Z)
VSCAN	Scan the next field on a type 2 card for a variable
DSCAN	Scan digits after decimal point for a variable
FETCH	Subroutine to low order the next character in accumulator
GETTYPE3	Check input card for type 3
TYPE3OR4	Identify input card as type 3 or type 4
DICTSCAN	Type 3 scan
LINESCAN	Form line record for each RPP specified for a field in a type 3
GETSET	Subroutine to obtain format line no. (FLN) and RPP from type 3
CBERROR1	Error message routines for type 3 card errors (e.g., control break errors)
PUTIMAGE	Subroutine to put line record on file A
PUTPREFORM	Subroutine to put preform on file A
TYPE4EDIT	Create image-4 record and put on file A
ILEGREL	Error branch points for type 4 cards
R1TEMP	DC's and DA's for zeroing areas; adcons for moving flds from type 3 to acc 1 and 2
R1MESS1	DC's for error messages for type 4
R1MSG1	DRDW's defining error messages
RPGP1	Phase 1 initialization (zero table areas)
IMAGE	Work areas and tables
RPGSYCTL	Initialization for Phase 2; move tables to upper storage
IOQOPEN	OPEN subroutine
END	End control statement followed by literals

Figure 7. Phase 1, Storage Map

Location	Function
	CMREC, STC, IOQ, Autosort communication block
TAPEFILEAA	DTQ's and I-O areas for Phase 3
RPGSC3	Initialization of tape units and Autosort communication block for Phase 4
R3START	Initialization of work areas; open files
R3TRAA1-4	Get first image-2 record from file A
R3TRAA1	Set editor fields from image-2 record
R3TRAA2	Get a file A record and classify as a preform, line, image-2, or image-4
R3TRAA9	Set editor fields from the preform record
M3TRZZ1	Set editor fields from the line record
R3TRAB2+2	Put a liteditor record on file C for RPG62
R3TRAB4A	Put an editor (or liteditor) record on file C
ERROR3	Error message and type routines
R3TRAA5	Further processing of preform, image-2, or image-4 records
LITEDITOR	Create a liteditor record for RPG62 (and RPG63 if the line is a total line)
DLINESCAN & RIGHTSCAN	Scan the image-2 record and create a DLINE subsequent entry first for the variable field and then for the constant field between this and the last variable field
FETCH	Subroutine to fetch a character for the scans
WCOLLECT	Create a DLINE subsequent entry
COMPBETLIM	Create an unlabeled entry for a constant area in a DLINE
PUTDLINE	Subroutine which puts a DC/DLINE record on file C and blanks the work area
LITDLINE	Generate a DC for an all-literal format line
DLINEHEAD	Generate a DLINE header
DLINELAST	Generate the last entry of a DLINE
ERROR1	Error message and type routines
PUTEDITOR	Subroutine which puts an editor record on file C
R3TRAB8	Image-4 copy routine
R3RDW1	RDW's for zeroing editor record area, moving flds to work areas, and blanking work areas
AZERO	Constants, work areas, and error messages
RPGP3	Phase 3 initialization
SETPGPRT	Subroutine to set PHASECOMM2 with the print codes specified in report format lines
IOQOPEN	OPEN subroutine
END	End control statement followed by literals

Figure 8. Phase 3, Storage Map

Location	Function
	CMREC, STC, IOQ, Autosort communication block
TAPEFILEAA	DTQ's and I-O areas
BSORT1	Bring down tables; test for two programs
BSRT1	Make list of every third word of NA table
BSRT2-2	Sort list of names on beginning address
COMP	Compress list maintaining relative field definition
BMOVE	Build a table grouping fields which are not more than one word apart
SOMERDW	Work areas for presort program compilation
BSUBROUT	Declarative statement output routines for report program
BLABELOUT	Procedure statement output subroutine for presort program
BZERMAC	Zero declarative statement area
GO	Generation of control cards for Sort 90
BDTFSUB1	Generation of DTF for input file to presort program
B5DTF15	Generation of label information DC for input file to presort program
BNEXT+1	Generation of intermediate file DTF (output of presort program)
BHEADER	Generation of input file DA for presort program
B2DA	Generation of intermediate file DA for presort program
BOPEN	Generation of presort program (QSTART to QGET)
BREPEAT	Generation of MOVE macro-instructions
BPUT	Generation of PUT QOUTPUT macro-instruction and final housekeeping instructions
	Generation of DTF for intermediate file (input to report program)
BDTFSUB2	Generation of DTF for output file (report program)
BDC2BR	Generation of label information DC for intermediate file (presort output)
B5DTF40	Generation of report program input DA
BIF	Generation of report program output DA
BFI	Deletion of initial three words of the DC/DLINE records
ORCTL	Constant areas and error messages
BTAPEA	Constants which define generated presort program
STACK	Subroutines used in generation of MOVE macro-instructions for presort program
THREE30	Constants and literals
RPGPH5TEST	Initialization for Phase 5
IOQOPEN	OPEN subroutine
SPOT	Work areas and tables which overlay RPGPH5TEST and IOQOPEN
END	End control statement followed by literals

Figure 9. Phase 5, Storage Map

Location	Function
	CMREC, STC, IOQ, Autosort communication block
TAPEFILEAA	DTQ's, I-O areas, and image-4 work area
MG1START	Generation of initialization of report program
M61AA8-5	Generation of control break test routines
M61AA1	Generation of control break routine
M61AB2	Generation of total line control branches
M61AB3+1	Generation of heading line control branches
M61AC1-1	Generation of control break move routines
M61AC8-5	Generation of control break test area
M61AC7-6	Generation of EOF routine
GENERB	Output routines to put out generation on file A or B
M61STACKDA	Generation of data selection routine
DTLOUTPUT	Generation of DSW to define selection switches
STACK	Subroutines used in generation of macro-instructions
THREE30	Constants used in building the macro-instructions
SEQNO	Temporary storage areas, error messages, and DRDW's
NOCB	Generation of QCBEND to QEOF if no control break specified
STARTEST	RPG61 initialization
EOFILEC	EOF routine
SETSWCN	Constant area for generation of SETSW instructions
LABEL	Work areas and table areas
END	End control statement followed by literals

Figure 10. Phase 6, Load 1, Storage Map

Location	Function
	CMREC, STC, IOQ, Autosort communication block
TAPEFILEAA	DTQ's, I-O areas, and work area for editor records
BJOHN	Work areas and DRDW's
TFPTBL	Tape file and printer table and NAMEFIELD table (NA)
BEQURDW	Assorted statements under individual DC's to be generated into the report program
BINCLAB	Preliminary housekeeping
BENTER1	Phase 6, load 2 initialization
BSTART	Subroutine which obtains an editor record from file C
BMIKE	Generation of editing of accumulated fields into total lines and writing the lines
BPOE	Generation of editing of fields into detail lines and writing the lines
BKARL	Generation of editing of fields into HF- lines and writing the lines
BEVER	Generation of editing of fields into heading lines and writing of the lines
BPR10	Generation of 7400 control information for lines to to be put out on-line
BPR11	Generation of blank HP1 line if no HP- lines were specified
BZOOT	Generation of editing of non-accumulated variable fields into total lines
BPEND	Generation of the QFCON routine (to initialize total lines)
BPR32	Generation of switch initialization (QSET routine)
BPR39C	Generation of page turning routine (QPT routine)
EOFILEC	End-of-file routine used when RPG63 is not needed
BP1	Generation of QSK routine (put out blank lines before first detail line)
BP1B	Generation of editing of fields into detail lines and writing of the lines
RETURN2	Routines to fetch characters, put them in the output area, and write the record
THREE30	Constants used in generation
END	End control statement followed by literals

Figure 11. Phase 6, Load 2, Storage Map

Location	Function
	CMREC, STC, IOQ, Autosort communication block
TAPEFILEAA	DTQ's, I-O areas and work area for editor records
M6ACCSTART	Initialization for generation of accumulation routine; generation of accumulation area DA header
M6ACCAA1	Generation of first statement of the accumulation routine for a total line
LASTCOMP	Generation of subsequent entry for accumulation DA to define one accumulation area
MOREB	Generation of subsequent entry for accumulation DA to define all areas for one line
M6ACCAA2	Generation of test of conditional requirement made before accumulation
M6ACCAA4	Generation of ARITH macro-instruction
STACKDA	Place generated coding in output area
PUTCOMP	Input-output subroutines
LABEL	Work area where ARITH macro-instructions are built
ORIGINDC	DC describing the statement ORIGIN CNTRL*-1 for overlaying generated NOP's
COMPERR	Error message and type routine for a compiler error
GETRDW	DA's for work areas and DC's for constant areas
NAMEFIELD	NAMEFIELD table
STARTEST	Phase 6, load 3 initialization
FINISH	End-of-file procedures
END	End control statement followed by literals

Figure 12. Phase 6, Load 3, Storage Map

Location	Function
	CMREC, STC, IOQ, Autosort communication block
TAPEFILEAA	DTQ's and I-O areas
RPGSC7	Initialization for Phase 7
GETFILEA	Obtain file A records and write onto file E and/or file C
EOFILEA	Close file A
GETFILEB	Obtain file B records and write onto file E and/or file C
EOFILEB	Close file B
COPYFILEE	Copy file E onto file C if a presort program was generated
PH7EXIT	Close remaining files, branch to SYCL4 to call Autocoder
PAGELINE	Insert a page/line into the Autocoder cards before writing on file E and/or file C
PAGE	DC containing page numbers
CARDRDW3	Work area
IOQOPEN	OPEN routine
END	End control statement followed by literals

Figure 13. Phase 7, Storage Map

(NA) NAMEFIELD Table -- Four-word entries, up to 99 entries

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	A	First five characters of name									
1	A	Last five characters of name									
2	+	NSC -- Serial no. of entry		Start position of field				End position of field			
3	+/-	Digits after dec point		(These positions will be used by Phase 5 if presort) *							

SERIAL Table -- One-word entries, up to 198 entries

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	A	Not used				Format line number from CC 2-4 of type 2 cards					

SORT Table -- One-word entries, up to 9 entries

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	+	W	W	W	D	L	L	Not used		NSC of fld ** in NA table	

CBREAK Table -- One-word entries, up to 9 entries

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	+	Up to five NSC's for those fields that comprise the control break field for level 1, followed by level 2, etc									

TFPTBL -- One fifteen-word entry

Word No.	Sign	0	1	2	3	4	5	6	7	8	9	
0	A	First five characters of input label identification										for input file
1	A	Last five characters of input label identification										for input file
2	+	Not used						Chan	Unit 1	Unit 2	Dens	for input file
3	+	Rec type	Max rec length					Max blocking factor				for input file
4	+	Not used	Start position					End position				***
5	+	Not used						Chan	Unit 1	Unit 2	Dens	****
6	+	Not used					Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	for Sort 90
7	+	Not used					Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	for Sort 90
8	A	First five characters of output label identification										for report output file
9	A	Last five characters of output label identification										for report output file
10	A	Not used				Retention cycle						for report output file
11	+	Not used							Chan	Unit 1	Unit 2	for report output file
12	+	Not used						Printer number				
13	+	Not used								Lines per page		
14	+	Not used							Max char per line			

* + indicates numeric field, - indicates alphameric field
** WWW is the word number within which the control field begins
D is the starting digit within the first word
LL is the number of digits in the field
*** Used for form 3 records only
**** This file is the output from the presort, the input to Sort 90 and to the report program.

Figure 14. Tables Created by Phase 1

Preform Record (Refers to type 2)

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	+	Format line serial no.			RPP			1**	LPP		
1	+	LD		RD		0	0	0	0	0	0
2-29	+	0	0	0	0	0	0	0	0	0	0

Line Record (Refers to type 3)

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	+	Format line serial no.			RPP			2**	*	Accum Indic	Group Print
1	+	Cond requir (double digit)				0	0	0	0	NSC of fld in NA table	
2-29	+	0	0	0	0	0	0	0	0	0	0

Image-4 Record (Refers to type 4)

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	+	500			Image-4 seq no.			3**	0	0	0
1	A	First five characters of name of field									
2	A	Last five characters of name of field									
3	A	Not used								Relation	
4	A	First five characters of 1st range									
5	A	Last five characters of 1st range									
6	A	First five characters of 2nd range									
7	A	Last five characters of 2nd range									
8	A	Not used				First cond requirement					
9	A	Not used				Second cond requirement					
10	A	Not used				Third cond requirement					
11	A	Not used				Fourth cond requirement					
12	A	Not used				Fifth cond requirement					
13	A	Not used						Resultant condition			
14-29	+	0	0	0	0	0	0	0	0	0	0

Image-2 Record (Refers to type 2)

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	+	Format line serial no.			0	0	0	0**	0	0	0
1-28	A	Text of format line (incl continuation cards)									
29	A	Not used									

* Special symbol code: COUNT = 3, DATED = 4, PAGE # = 5
** Sort code

Figure 15. Tape Records Created by Phase 1

EDITOR Record

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	A	Format line number[0]						Space code[0]		Carr ctrl * code 0	
1	+	RPP[1]			Spec symb[2]	AI[2]	GI[2]	RPP of dollar sign if present			Print code[0]
2	+	Cond requir (double digit)[2]				NSC[2]		Digits left of dec pt		Digits right of dec pt	
3-18	+	0	0	0	0	0	0	0	0	0	0

DLINE (or DC) Record

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	+	0	0	0	0	0	0	0	0	0	0
1	+	0	0	0	0	0	0	0	0	0	0
2	+	DLINE or DC serial number			0	0	0	0	0	0	0
3-18	A	Autocoder DLINE or DC in card image format (Autocoder continuation cards(s) is in the DLINE record if necessary)									

IMAGE-4 Record

Word No.	Sign	0	1	2	3	4	5	6	7	8	9
0	+	500			Image-4 sequence no.			3	0	0	0
1	+	0	0	0	0	1	0	0	0	0	0
2	+	0	0	0	0	0	0	0	0	0	0
3	A	First five characters of name of field									
4	A	Last five characters of name of field									
5	A	Not used								Relation	
6	A	First five characters of 1st range									
7	A	Last five characters of 1st range									
8	A	First five characters of 2nd range									
9	A	Last five characters of 2nd range									
10	A	Not used				First cond requirement					
11	A	Not used				Second cond requirement					
12	A	Not used				Third cond requirement					
13	A	Not used				Fourth cond requirement					
14	A	Not used				Fifth cond requirement					
15	A	Not used						Resultant cond			
16-18	+	0	0	0	0	0	0	0	0	0	0

* The superscripts refer to the sort code of the record from which the field was taken:
0 = Image-2, 1 = Preform, 2 = Line

Figure 16. Tape Records Created or Passed by Phase 3

Glossary

ACCUMULATION: Totaling.

CONTROL BREAK: A change in the information contained in a field designated as a control break field in two successive records.

CONTROL BREAK LEVEL: A means of establishing the significance of control break fields in relation to each other. In RPG, level 1 is the least significant (or minor) level.

EDITOR RECORD: A record created by the information transfer of Phase 3 and used by Phase 6 to generate edit coding.

FORMAT LINE NUMBER: The three characters in card columns 2-4 of a type 2 card.

GROUP PRINTING: The technique of printing fields only in the first body line following total and/or heading lines. In RPG, dollar signs may be group printed as well as whole fields.

IMAGE-2 RECORD: A record created by Phase 1 from type 2 cards. It is identical to the format line described on a type 2 card(s) except that the first word of the record contains internal identifiers.

IMAGE-4 RECORD: A record created by Phase 1 which contains the information from a type 4 card in a slightly edited form.

LINE RECORD: A record created by Phase 1 to show the print positions where a variable field is to be printed in a format line.

LITEDITOR RECORD: An editor record with an RPP field of 999. It indicates a literal format line to Phase 6.

PREFORM RECORD: A record created by Phase 1 to show the location of a variable field in a print line.

TYPE 1 CARD: The tape File and Printer Specifications card.

TYPE 2 CARD: A Report Format card.

TYPE 3 CARD: A Field Dictionary card.

TYPE 4 CARD: A Data Selection Requirements card.

Abbreviations

accum	accumulation
acc	accumulator
addr	address
AI	accumulation indicator
br	branch
brk	break
carr	carriage
CC	card column
cd	card
char	character
cnt	count
cond	condition(al)
contin	continuation
ctr	counter
ctrl	control
const	constant
dec	decimal
def	define
declar	declarative
dig	digit
EOF	end-of-file
fld	field
FLN	format line number
gener	generate
GI	group print indicator
hdng	heading
incl	including
incr	increment
indic	indicator
init	initialization
instr	instruction
LD	integer digits
LPP	leftmost print position
max	maximum
msg	message
NA	NAMEFIELD table
no.	number
num	numeric
NSC	namefield serial code
pg	page
pg/ln	page-line number
pos	position
PP	print position
prev	previously
proc	process
prog	program
pt	point
rec	record
ref	referencing
rel	relative
rept	report
requir	requirement
RD	decimal digits
RPG	Report Program Generator
RPP	rightmost print position
rtne	routine
sel	selection
spec	special
specif	specified
st	start
subseq	subsequent
sw	switch
symb	symbol
tbl	table
tot	total
var	variable
wd	word
wk	work
wr	write

IBM 7070 Rept. Prog. Gen. Printed in U.S.A. C28-6192

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York