Programming Systems Analysis Guide

IBM 7070 Series Four-Tape Autocoder

IBM

IBM

Programming Systems Analysis Guide

IBM 7070 Series Four-Tape Autocoder

Preface

This manual was prepared by Applied Programming to provide detailed information on the internal logic of the 7070 Series Four-Tape Autocoder assembly program. It is intended for technical personnel who are responsible for diagnosing the system operation or for adapting the programming system to special usage.

Certain knowledge is a prerequisite for the full utilization of this manual. The reader should have a basic knowledge of the 7070 Autocoder language to the extent that it is described in the *IBM 7070/7074 Four-Tape Autocoder* Reference Manual, Form C28-6102, and an understanding of the IBM 7070 Input/Output Control System which is an integral portion of Four-Tape Autocoder. Background on this system can be obtained from the *IBM 7070 Input/Output Control System* Bulletin (J28-6033) and the *IBM 7070 Input/Output Control System* Programming Systems Analysis Guide (C28-6119).

A program listing may be obtained by sending a magnetic tape (at least 400 feet in length) to the IBM 7070 Program Librarian at Data Processing Library Services, 590 Madison Avenue, New York 22, New York.

Contents

Four-Tape Autocoder is an assembly program that allows the user to write a program in Autocoder language and use substitution-type macros and subroutines. All programs written in Basic Autocoder can be assembled by Four-Tape Autocoder.

Macros and subroutines are contained on the system tape in a section called the "library." The system tape comes equipped with a library primarily composed of IOCS macros. The user may use this library and add or delete macros and subroutines. Information necessary for the user to write macros and subroutines is contained in the *IBM 7070/7074 Four-Tape Autocoder* Reference Manual, Form C28-6102.

Three different operations may be performed with the Four-Tape Autocoder system tape. These operations (runs) are:

1. A system run that updates the present Autocoder system tape
2. An original assembly that produces an object-output program from source entries
3. A reassembly that produces an updated object-output program from a previous listing of that program

The Four-Tape Autocoder system tape consists of:

1. System control, used to load the phases of the program into storage.
2. Four distinct phases that perform the assembly.
3. A library of subroutines and macros that is used during Phase 1.
4. A librarian that is used to update the system tape during a system run.

The system tape, as received, operates on a given configuration of machine conditions (permanent options). These permanent options may be temporarily changed by control cards during any given run of this system tape (temporary options). The permanent options may be changed to a new configuration on the system tape during a system run.

For all runs, system control, the librarian, and the macro and subroutine name table are loaded into storage. System control first reads any options that may alter permanent options for this run. It then reads a "run" card that instructs the program to make a particular type of run (system, assembly, or reassembly). If the run control card indicates a system run, system control causes the librarian section of the program to be executed. The librarian reads change cards and then alters the system tape to conform to the changes. These changes may include adding to, deleting from, and changing macro or subroutine skeleton instructions in the library. They also can alter instructions and data in any of the four phases, system control, or the librarian.

If either an original assembly or a reassembly run is indicated, Phase 1 is loaded. This phase overlays the librarian section of the program. The macro and subroutine name tables will be preserved in storage, however, for use in Phase 1. The phases of the program are then executed. At the end of each phase, except Phase 4, a branch is made to system control to bring in the next phase. At the end of Phase 4, messages are typed informing the operator of the nature and location of his output tapes. An end-of-job halt is then made.

Each phase performs specific functions in the conversion of an Autocoder symbolic-language source program to a 7070 machine-language object program.

Phase 1

1. Reads and processes each source-program input record.
2. Inserts macro-instructions and subroutines.
3. Develops a table of symbols which must be previously defined.
4. Produces 17-word output records (original 16 words plus a control word).

Phase 2

1. Expands each 17-word input record to a 26-word record.
2. Assigns storage.
3. Processes the operation code of each instruction.
4. Processes all actual addresses (including asterisks).
5. Processes all actual index words and electronic switches, recording them in availability tables.
6. Processes the symbolic table built in Phase 1.
7. Builds record blocks (on tape) of symbolic labels and their assigned storage locations.
8. Places symbols found in the operands of Autocoder statements in specific words of the 26-word record, and sets switches to indicate the type of further processing necessary for each.

AA

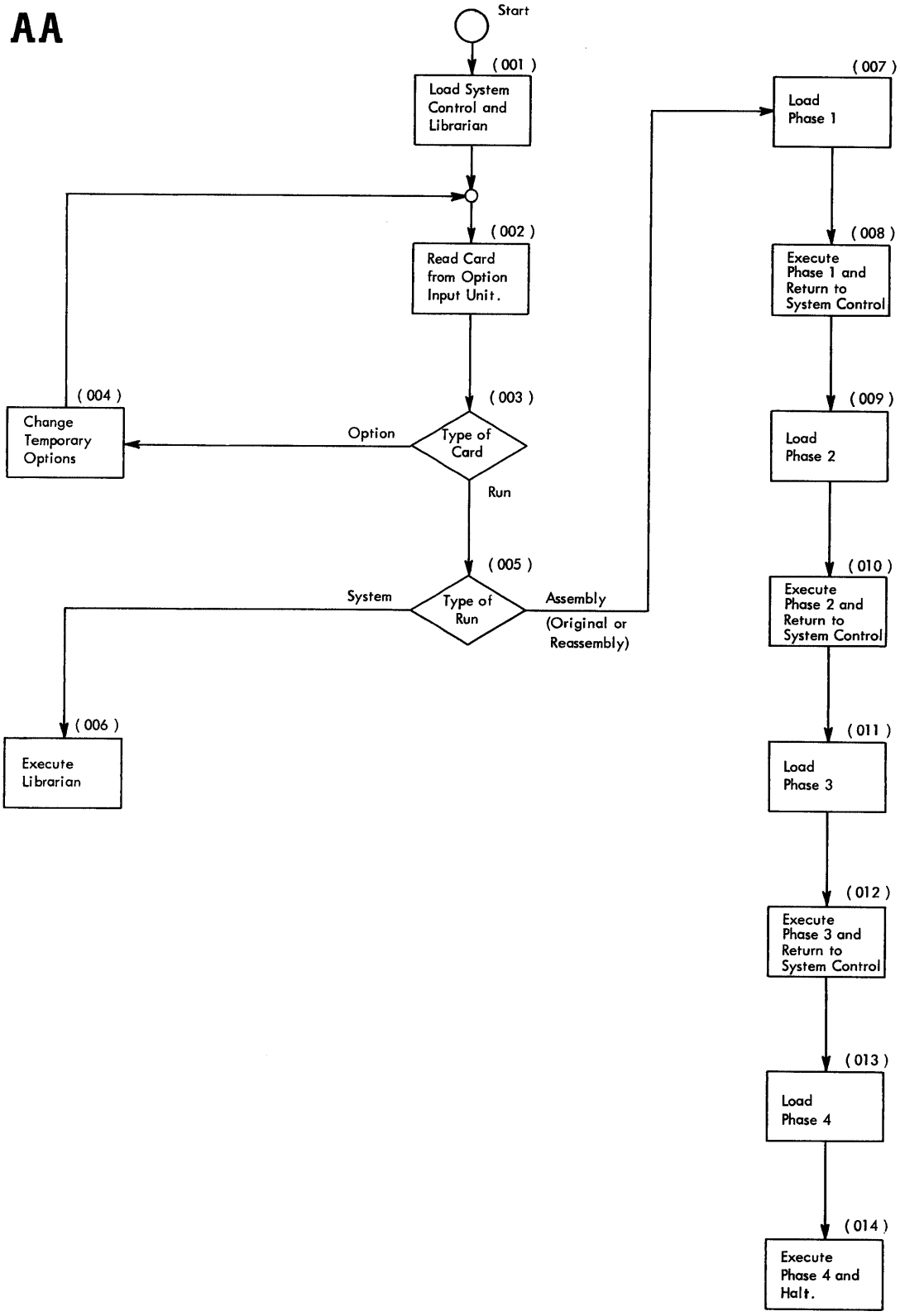


Chart AA. Organization of Four-Tape Autocoder

Phase 3

1. Builds a table of the symbolic labels from Phase 2.

2. Processes the symbols found in the expanded portions of the 26-word records.

3. Builds a table of symbolic index words and switches and their assigned locations.

Phase 4

1. Completes the assembly of certain instructions from the table of symbolic index words and switches.

2. Packs literals and inserts their assigned locations into the assembled instructions.

3. Prepares an edited listing tape.

4. Prepares an assembled program tape in condensed-card format.

Description of Chart AA

Chart AA is a general flow chart of the entire Four-Tape Autocoder program. Each block is numbered and is described as to its general function.

Block 1 represents the load program. It places in storage the system control and librarian programs and then branches to system control.

Blocks 002, 003, 004, 005, 007, 009, 011, and 013 represent system control. In block 002, system control reads an Autocoder control card from the option input unit as specified in the permanent options. Block 003 determines whether the card just read is a run control card or an option card. The detection of an option card causes a branch to block 004, which processes the card by altering the indicated standard option for the present machine run. From block 004, the program returns to block 002 to read the next card. When a run control card is read, block 005 determines which of the three available runs is indicated. A system run causes the execution of block 006. In an assembly run (either original or reassembly), system control loads Phase 1 (block 007).

Block 006 represents the processing of the librarian which was loaded during the execution of block 001. The librarian enables the programmer to add to, change, or delete any portion of the Four-Tape Autocoder system tape; i.e., system control, librarian, library, or any of the four phases.

Blocks 007, 009, 011, and 013 show system control loading Phases 1, 2, 3, and 4, respectively. Blocks 008, 010, 012 and 014 represent the processing of Phases 1, 2, 3, and 4, respectively. Each phase processes specific types of source entries or given portions of the entries. At the conclusion of Phase 4 (014), the assembly is complete and an end-of-job halt is made.

Input-Output Units

The input-output units used during Four-Tape Autocoder processing are shown in Figures 1, 2 and 3. These figures present a graphic flow of input and output for each phase and pass. Figures 1 and 2 represent assembly runs and are the same except for Phase 3. Figure 1 shows a single pass for Phase 3 and Figure 2, a multiple pass. Figure 3 illustrates the two-pass operation of a system run.

Inputs are shown above the label designating the phase or pass, with an arrow from the unit to the labeled area. Output units are shown below the phase or pass, with an arrow extending from the area to the unit. The output of one operation is used as the input to the next. Unlabeled units are not used.

The tape units are numbered at the top of each figure. Single-channel operation uses tape units 10, 11, 12 and 13. Two-channel operation uses tape units 20, 21, 12 and 13. The capital letters A-G at the lower right of the tapes refer to their formats.

In Figures 1 and 2, the input may be source programs from tape unit 12, the card reader on any synchronizer, or a previous listing tape from tape unit 13. The four-tape system tape is always on tape 10 or 20. The output of Phase 1 is on tape 11 or 21 which becomes the input for Phase 2. The output for Phase 2 is on tapes 12 and 13 which becomes the input to Phase 3.

Phase 3 may be a multiple-pass operation depending upon the number of labels in the symbol table on tape 13. Phase 3 analyzes the size of the symbol table and determines whether Phase 3 will be a single pass as shown in Figure 1 or a multiple pass as shown in Figure 2. The output of Phase 3 for a single-pass operation is on tape 11 or 21 so that, when two channels are used, the efficiency of two-channel processing may be obtained for processing Phases 3 and 4. Figure 2 shows all the output tape configurations for Phase 3 in the order in which they occur. The final output of Phase 3 depends upon the size of the symbol table and becomes the input to Phase 4. The output of Phase 4 varies with the input; this is not a problem, since a typed message at the end of Phase 4 will state which units contain the object program and the listing tape.

The output tape configuration of Phase 4 as shown in Figure 1 is always the same and is indicated by a typed message.

Figure 3 describes a system run. The system tape is placed on tape 10 or 20. The control and change cards are either on tape 12 or are read from the card reader through any synchronizer. The outputs of Phases 1 and 2 are always the same as shown in Figure 3.

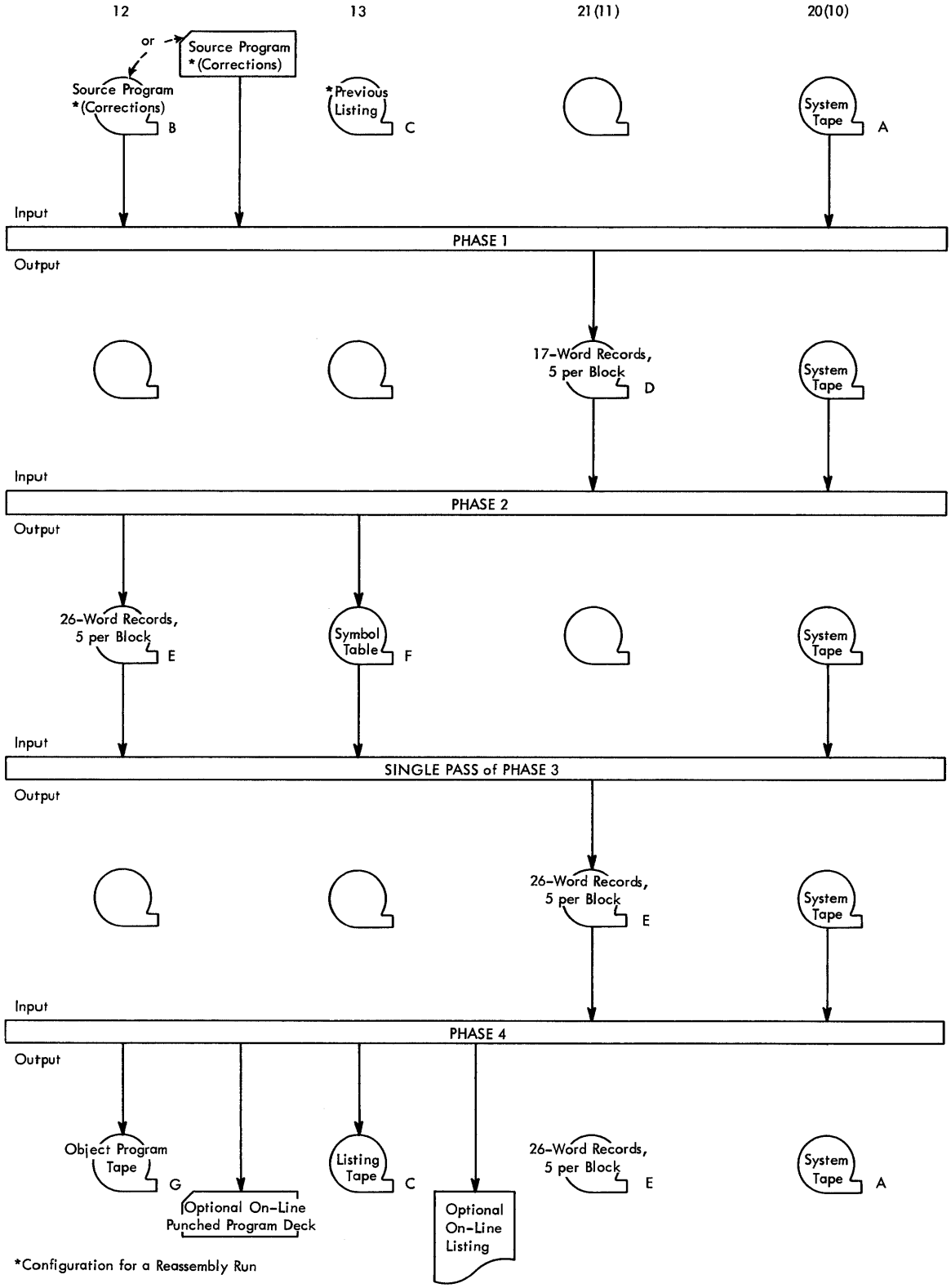


Figure 1. Input/Output for an Assembly Run, Single-Pass Phase 3

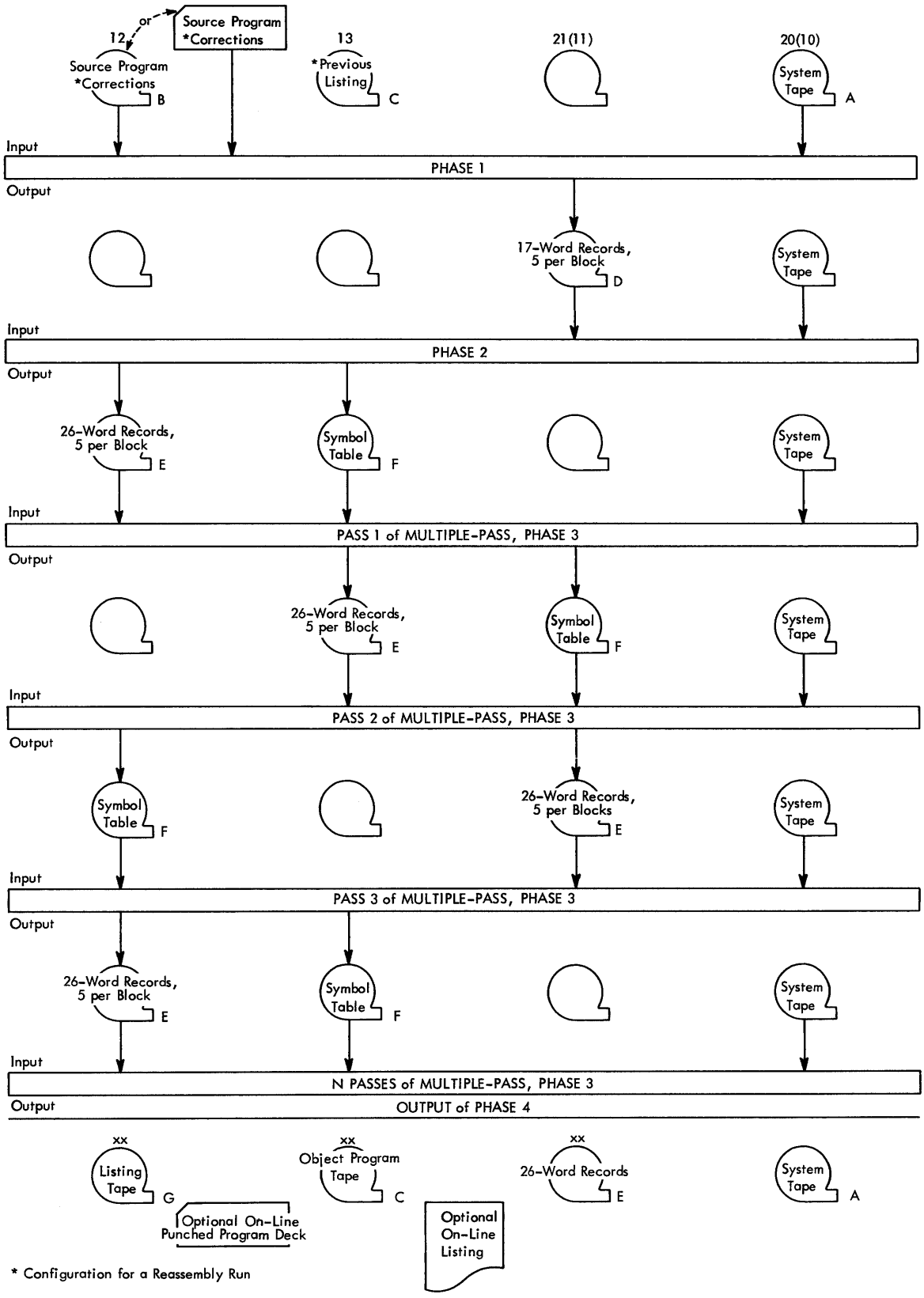


Figure 2. Input/Output for an Assembly Run, Multiple-Pass Phase 3

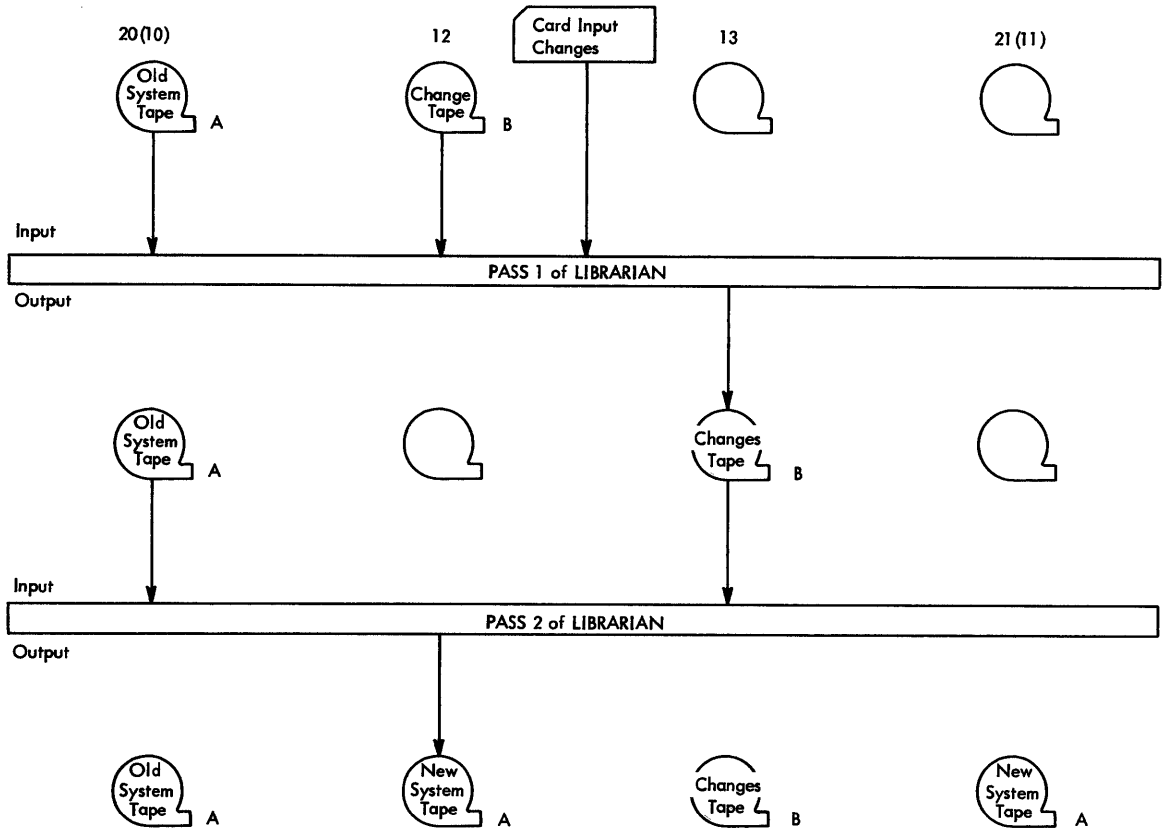


Figure 3. Input/Output for a System Run

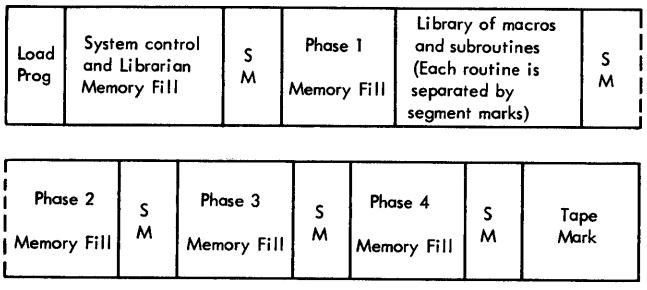
Tape Formats

System Tape: See diagram for format. The system tape includes:

Load Program: Five cards, eight numeric words each; one execute card to place system control in storage.

Library: 16-word alphameric records, five per block. Segment mark (SM) between macros.

Phase Loads: Numeric; length variable.



Source Program: 16-word alphameric records, unblocked.

Listing Tape: 23-word alphameric records, unblocked.

17-Word Record: Five records per block; 16 alphameric words (source program record) plus one numeric word (control information).

26-Word Expanded Record: Five records per block.

Words 1-16	alphameric
Words 17-19	numeric
Words 20-22	alphameric or numeric
Words 23-24	alphameric
Word 25	alphameric or numeric
Word 26	alphameric

Symbol Tape: 130-word records, each record containing a maximum of 43 symbols. Each symbol uses three words; the first two are alphameric and the third is numeric. The last word of each 130-word record contains zeros.

Object Program Tape: 16 alphameric words in condensed card format. Records are unblocked.

Operating Instructions for Four-Tape Autocoder

Each of the three types of runs possible with Four-Tape Autocoder requires slightly different preparations before a run can be started. Once the system has been set up for a particular type of run, the same console procedure may be used to start the run. Preparations

required for each type of run are described separately below, and are followed by one group of instructions pertaining to console operations.

The system tape that is available from the IBM 7070/7074 Program Librarian assumes a typical 7070 series system that might be used for assembling programs with Four-Tape Autocoder. This typical system is described by the following "permanent" options included as part of the system tape:

1. Option cards, control cards and the source program will be read from tape.
2. No on-line unit record machines will be used.
3. The data processing system to be used by the object program and the processor has 9990 words of core storage available.
4. None of the tapes (including the source program tape) will have labels.

If these permanent options describe the user's system, no option cards need be included when assembling with Four-Tape Autocoder. If the user's system differs from the assumed one, however, or if additional options are desired, the permanent options on the system tape may be changed or new options may be added through a system run; new options may also be added or substituted for a particular run by the use of option cards ahead of the run control card.

To simplify the description of the preparations for each type of run, option cards and control cards which may be read from either cards or tape will be explained in terms of cards although they may actually be on tape. For example, if the permanent option unit is a tape unit, the statement "place option cards in the permanent option unit" means that the option cards must be written on a tape which is then mounted on the permanent input unit.

Preparations for an Original Assembly

1. Mount the Four-Tape Autocoder system tape on tape unit 20 (channel 2, unit 0). If only one channel is available, mount the system tape on tape unit 10 (channel 1, unit 0). Set the tape unit to high density.
2. Mount two work tapes as follows:

	TAPE CHANNEL	TAPE UNIT
Two-Channel Operation	1	3
	2	1
One-Channel Operation	1	3
	1	1

3. Place the option cards, if any, in the permanent input unit. If an OPTIONUNIT option card is included, any option card which follows the OPTIONUNIT card must be placed in the unit specified by the OPTIONUNIT card.
4. Place an original run control card in the permanent unit; if an OPTIONUNIT option card is included in step 3, the run control card must be placed in the unit specified by the OPTIONUNIT card.
5. Prepare the source program to be assembled as follows:
   A. If the source program has been written on tape by a card-to-tape operation, mount the tape on tape unit 12 (channel 1, unit 2). Set the tape unit to low density. (The source program may occupy one full reel of tape.)
   B. If the source program is in cards, place the source program deck in the on-line card reader, insert the 7500 Utility Panel and set the card reader alteration switches to B,B,A,A. Mount a work tape on tape unit 12 (channel 1, unit 2).
6. If a permanent option on the system tape or an option card included in step 3 specifies on-line printing or punching, insert the 7400 Utility Panel in the printer or the 7550 Utility Panel in the punch, and set the alteration switches to A,A,A,A.
7. Continue as explained under "Console Procedure."

Preparations for a Reassembly

1. Mount the Four-Tape Autocoder system tape on tape unit 20 (channel 2, unit 0). If only one channel is available, mount the system tape on tape unit 10 (channel 1, unit 0). Set the tape unit to high density.
2. Mount a work tape on tape unit 21 (channel 2, unit 1). If only one channel is available, mount the work tape on tape unit 11 (channel 1, unit 1).
3. Mount the edited listing tape produced by an original assembly or a previous reassembly on tape unit 13 (channel 1, unit 3).
4. Place the option cards, if any, in the permanent input unit. If an OPTIONUNIT option card is included, any option cards which follow the OPTIONUNIT card must be placed in the unit specified by the OPTIONUNIT card.
5. Place a recompile run control card in the permanent input unit; if an OPTIONUNIT option card is included in step 4, the run control card must be placed in the unit specified by the OPTIONUNIT card.
6. Arrange the corrections to the source program in page number and line number sequence. (See "Reassembly Run" in Form C28-6102 for card format used for corrections.) Prepare the corrections for the reassembly run as follows:
   A. If the corrections have been written on tape by a card-to-tape operation, mount the tape on tape unit 12 (channel 1, unit 2). Set the tape unit to low density. (The corrections may occupy one full reel of tape.)

B. If the corrections are in cards, place the cards in the on-line card reader, insert the 7500 Utility Panel, and set the card-reader alteration switches to B,B,A,A. Mount a work tape on tape unit 12 (channel 1, unit 2).

7. If a permanent option on the system tape or an option card included in step 4 specifies on-line printing or punching, insert the 7400 Utility Panel in the printer or the 7550 Utility Panel into the punch, and set the alteration switches on the printer and punch to A,A,A,A.

8. Continue as explained under "Console Procedure."

Preparations for a System Run

1. Mount the Four-Tape Autocoder system tape on tape unit 20 (channel 2, unit 0). If only one channel is available, mount the system tape on tape unit 10 (channel 1, unit 0). Set the tape unit to high density.

2. Mount the two work tapes as follows:

	TAPE CHANNEL	TAPE UNIT
Two-Channel Operation	1	3
	2	1
One-Channel Operation	1	3
	1	1

3. Place the option cards, if any, in the permanent input unit. If an OPTIONUNIT option card is included, any option cards which follow the OPTIONUNIT card must be placed in the input unit specified by the OPTIONUNIT card.

4. Place a systems run control card in the permanent input unit; if an OPTIONUNIT option card is included in step 3, the run control card must be placed in the unit specified by the OPTIONUNIT card.

5. Group the changes to the system tape. (If no changes are used, duplicates of the system tape will be produced.)

Place changes to permanent options behind one ASIGN card.

Place all additions and changes to any items in the library behind an INSER card for those items. (DELET cards need not be grouped.)

Place changes to the Four-Tape Autocoder processor behind each UPDAT card which identifies the section of the processor that is to be changed.

The various groups of changes may be in any order; e. g., the ASIGN card followed by option changes may follow changes to the library. Card formats for the various changes to the system tape are described in the "System Control and Librarian" section. Prepare the changes for the system run as follows:

If the changes have been written on tape by a card-to-tape operation, mount the tape on tape unit 12 (channel 1, unit 2). Set the tape unit to low density. (The changes may occupy one full reel of tape.)

If the changes are in cards, place the cards in the on-line reader, insert the 7500 Utility Panel, and set the card reader alteration switches to A,A,A,A. When an on-line card reader is used for reading the changes, mount a work tape on tape unit 12 (channel 1, unit 2).

6. Continue as explained under "Console Procedure."

Console Procedure

1. Manually store the following three instructions, in which C is either a 1 or a 2 to indicate whether one or two channels are to be used for the run:

STORAGE LOCATION	INSTRUCTION TO BE STORED
0000	−8C01010004
0001	−0100030010
0002	+5100C10002

2. Set console switches as follows: Turn alteration switches 2, 3, and 4 off. Alteration switch 1, if on, causes a halt at the end of each phase. Turn all accumulator overflow switches off. Turn both unit record priority controls to OFF and N.

3. Depress the run key.

4. Depress the computer reset key.

5. Depress the start key.

Multiple stacking of several input-source programs and several object-output programs and listings may be accomplished if one, two, or three additional tape units are available. Alteration switches 2, 3 and 4 are used to control these tapes.

Several source programs may be read from tape 14 (with SOURCE DECK USE TAPE option) if alteration switch 4 is on. Each program must be followed by a tape mark and prefaced by an original run card. If the option LABELSIN EQUIP YES is used, then the first card will be a dummy card to act as a tape label. Tape 14 is never rewound.

Program listings are stacked on tape 15 if alteration switch 2 is on. Each listing is followed by a tape mark. Labels are not written and the tape is not rewound.

The condensed-card tape outputs are stacked on tape 22 if alteration switch 3 is on. Each condensed card deck file is followed by a tape mark. Labels are not written, and the tape is not rewound.

While the system control and librarian are but a single section of the system tape and, therefore, are loaded at the same time, they are two sections of the processor.

System control controls the operation of the entire system. To do so it remains in storage throughout an entire machine run. It processes all option cards used to alter any of the permanent options for a particular machine run and the run control card. For temporary changes, the option cards must be encountered prior to the run control card. From an analysis of the run control card, system control determines the type of run and either loads Phase 1 for an assembly run or initializes the librarian for a system run. During an assembly run each phase except the last, when fully executed, returns to system control which then loads the next phase.

The librarian is the program which processes system tape change cards, thereby allowing the system tape to be revised. The librarian also contains the table of the names of macros and subroutines that may be used by Phase 1. Consequently, this table remains in storage for use by Phase 1. The remainder of the librarian section is executed only during a system run and is overlaid by Phase 1 for an assembly run.

Description of Charts BA, BB, and BC

Block 050: The system control and librarian programs, in a single section of the system tape, are placed into storage by the load program.

Block 051, START: The permanent options (loaded with the program) are placed into index words 95 and 96. These index words contain the temporary options which may be modified at the beginning of the run, and control the operations throughout the run.

Block 052: An input record is read from the card reader or tape (according to the option). The normal exit from this block is to examine the record and take appropriate action. The other exit, EOF during a system run, starts the last pass of the run. If the EOF exit is used on an original assembly or reassembly run, an error type-out is made.

Block 053, ASW: This switch is initialized to NOP and will be set to a branch if a system run card is recognized. The switch setting to branch allows entry to the routines which examine the change cards and take appropriate action.

Block 054, RUNROUTN: The operation code is compared to RUN. A BE passes control to block 058.

Block 055, TESTOPTION: The operation code on the input record is compared to the option operation codes possible (USE, OBJECT, and EQUIP).

Block 056, OPERROR: The operation code is not legal. An error message, "INVALID OPTION CONTROL," is written.

Block 057: A BE on a legal operation code causes entry to individual routines (not shown) that examine the operand of the input record, alter it to numeric, if necessary, and change the given temporary option to that in the record.

Block 058: The first word of the label is compared to ORIGI, RECOM, and SYSTE. Recognition of a system run card (equal comparison to SYSTE) passes control to block 067. A message (7070 AUTOCODER #VVLLL *type run*) is typed indicating the run specified, where VV indicates the version number and LLL the level number of the system tape being used.

Block 059, ASSEMROUTN: Position 1 of index-word 96 (TEMPOPT2) is set for the type of run: zero for an original assembly, 1 for a reassembly. System control is then entered to read the first phase into storage.

Block 060, PHASELOAD: The final status word of the system tape (position 0) is set to 9. This forces use of the pivot at location 159 to enter the error-testing routine at interrupt. The system tape is then moved to its proper position for the phase load.

Block 061, FILL: The system tape is read into storage. (MACROTAPE is the symbolic location of the read RDW.)

Block 062: Location 893, position zero, is compared to the proper phase number (PHASE, positions 2-5). An unequal comparison passes control to POSERROR (063).

Block 063: The system tape is rewound and forward-spaced (PTSF) to the proper phase. Control then returns to block 061.

Block 064: The proper phase has been read into storage. A branch is made to the location indicated by the control word (location 893, positions 6-9) to start the phase.

Block 065: At the completion of each phase (except Phase 4) a pivot returns control to system control for continuation of the program.

Block 066: After completion of Phase 4 the end-of-job messages are typed and final halt 4444 is made.

Block 067: Initialization for the system run is accomplished by setting ASW (053) to a branch. The files to be used in the run are opened and a segment mark is written on the output file (change tape). Control then returns to block 052 via connector 2 to read the next record.

Block 068, LIBSPLIT: The operation code is examined to determine if this record is a header card. If the comparison is equal (ASIGN, DELET, etc.), a branch is made to individual routines to handle this header and its following detail cards. If a header is not recognized, a branch is made to 1+EXIT where the detail card will be handled properly because of a previous setting of EXIT from a header card.

Block 069, DELETROUTN: The card was recognized as a DELET card. The first word of the label is compared to blanks to determine if the card is a header card or a detail card. If the label is blank the card is a detail card during an update operation and control passes via 1+EXIT to the update routine for detail cards.

Blocks 070, 071, 072 and 073: These blocks represent the initial setting of EXIT for entrance to the individual routines for each type of operation. The BLX is to 0+EXIT to complete the operation for the previous header card.

Blocks 074, 075, and the DELET pivot represent the completion of an operation on recognition of a new header card. Entrance to these routines or pivots is via 0+EXIT and exit from these points is via a branch to 2+EXIT. The DELET pivot is merely a branch to 2+EXIT, since no further operation is necessary after a delete header card.

Block 074, CHECKCOUNT: This block is entered to complete an update or insert operation. The count of the detail card is compared with that from the header card. If the count does not agree, a message is written indicating this. The output file is released, writing the last block on tape, and a segment mark is written after this block.

Block 075, STOREOPT: This block is entered to complete the ASIGN operation. The temporary options, altered by the ASIGN detail cards, are set into the permanent options. These options will be the permanent options for future runs using this system tape.

Blocks 076-081: These blocks represent the initial operation for each header card. They are entered with a branch to 2+EXIT. The exit from these routines, in all cases, passes control to read a new input card (052) via connector BA2.

Block 076: A delete header has been recognized; the macro or subroutine name given in the label field of the header is removed from the macro table.

Block 077: This block is centered for an ASIGN header card. The temporary options (index words 95 and 96) are reset by loading them with the permanent options.

Block 078: This block is entered for an insert header card. The name in the label field of the header is looked up in the macro table. If it is not found, control passes to block 080.

Block 079, CLEAROLD: The macro name was found and is removed from the old macro table for an insert header card. The macro name would also be removed on recognition of an insert header card.

Block 080, INSERNOTIN: The name in the header label field is placed on the new macro table. Control is passed via connector BA2 to read a new card.

Block 081: This block is entered on an update operation to initialize for the detail cards. The system tape is positioned to the macro given in the header label.

Blocks 082-086 are entered for detail change cards. The entry is via a branch to 1 + EXIT and the exit is via connector BA2 to read a new card. The ASIGN pivot is entered via a branch to 1+EXIT and forces entry to block 055 via connector BA3 to process the option card. These option cards will become permanent options on the new system tape.

Block 082: This block is entered via a branch to 1 + EXIT on an update detail card. The PCLIN field in the system tape record is compared with that in the input record. An equal comparison passes control to GETSYS (085). A low comparison passes control to PUTSYS (084).

Block 083, MACROSKEL: This block is entered via a branch to 1+EXIT on an insert detail card, or a high comparison between the system-tape record and the input record (update detail card). The new change record is PUT to the output tape. The index word keeping count of the number of detail cards in each group is incremented by one. Control now passes via connector BA2 to read the new record.

Block 084, PUTSYS: The system tape record is PUT to the output file, since it is lower than the change record.

Block 085, GETSYS: A GET is issued to the system tape file to obtain the next record for comparison.

Block 086, LIBCTLERR: This block is entered if any detail cards follow a delete header card. Since this is improper sequence, an error message is written. Control is passed via connector BA2 to read a new card.

Block 087, STARTPASS2: This block is entered on EOF from either the card reader or the input-record tape. A BLX is made on EXIT to 0+EXIT to complete operation for the last header card. The new setting of EXIT allows return to the merge routine for pass 2 of a system run.

Block 088: After completion of the pending operation the return via 2+EXIT first closes the files (not shown). The macro table is sorted according to the double-digit code generated from an operand character in the update header card of the macro or subroutine. The character follows the word UPDAT or INSER and dictates the relative position of the macro or subroutine in the library. The files for tapes 12 and 13 are reversed by resetting the digits in the channel and tape unit fields in the DTF. The files are then opened (not shown).

Blocks 089, 090: A test is made to determine if any patches are to be placed in system control or librarian. If patches are present, control passes to system control to load the patches (090). If there are no patches, control passes to block 091.

Block 091: The load-program and execute records are written on tape. This is accomplished with consecutive priority-write operations interspersed by branches to * (itself). The branch to itself is altered to a NOP after the interrupt when the write operation is completed and found error-free. The load program and execute records are card images in storage.

Block 092: Segment marks are written on both output tapes. A BLX is made on SORTX, setting it to the first (next) item in the macro table. SORTX determines the macro to be written and allows detection of the end of the macro table. Control then passes to TEST (093).

Block 093, TEST: A comparison is made between 0+SORTX, position 1 and a zero. If the reference is to a macro or subroutine in the table, this character will be the units position of a double-digit letter (first letter of the macro name). For reference to phases, this character is zero and control passes to PROGRAM (096).

Block 094: The proper tape (controlled by digit position 3 of the position table) is positioned to the next item in the macro table. That routine is copied from the given tape to the new system tapes.

Block 095: The BIX on SORTX is stepping the macro table addresses. When the 2-5 portion of SORTX becomes greater than the 6-9 portion, all records have been written and control passes to block 099.

Block 096: This determines if any patches for the storage load of this phase are on the change tape. If patches are to be made, control passes to block 097.

Block 097: Patches (load card format) are read from the change tape and placed into their proper position in storage.

Block 098: The phase is written on both new system tapes.

Block 099: Tape marks are written on both new system tapes and the files are closed. The end-of-job messages are then written and the final halt is made.

Control and Change Card Summary

Three types of runs are possible for the Four-Tape Autocoder program: an original assembly run, a reassembly run and a system run. Specific control cards are processed by system control, which determines the type of run and the parameters of the run. These control cards are described in the IBM *7070/7074 Four-Tape Autocoder* Reference Manual, Form C28-6102.

The Control and Change Card Summary (Figure 4) is included for checking the validity of the punching of the cards.

Page Line 1 5	Label 6 15	Operation 16 20	Operand 21	Remarks
	ORIGINAL	RUN		Input source prg
	RECOMPILE	RUN		Reassembly
	SYSTEM	RUN		Update system tape
		ASIGN		Change permanent options
	SOURCEDECK	USE	TAPE	Tape only
	SOURCEDECK	USE	1,2,3	Reader sync
	OBJECTDECK	USE	TAPE	Tape only
	OBJECTDECK	USE	1,2, or 3	Punch sync
	REPORTLIST	USE	TAPE	Tape only
	REPORTLIST	USE	1,2, or 3	Printer sync
	OPTIONUNIT	USE	TAPE	Tape only
	OPTIONUNIT	USE	1,2,3	Reader sync
	MESSAGES	USE	TAPEONLY	Tape only
	MESSAGES	USE	TYPEWRITER	Typewriter
	MESSAGES	USE	1,2,3	Printer sync
	MEMORYSIZE	OBJCT	NNNN	NNNN ≤ 9990
	MEMORYSIZE	EQUIP	NNNN	NNNN ≤ 9990/≤ 5000
	LABELSIN	EQUIP	Yes or No	Input labels
	LABELSOUT	EQUIP	Yes or No	Work tape labels
	TAPDENSITY	EQUIP	High/Low	Density of input
AA26		DELET	AA43	AA26 through AA43 delet
AA57		DELET		AA57 deleted
	MACRONAME	INSER	M,n	n=1 to 9999 cards
	SUBROUTINENAME	INSER	S,n	n=1 to 9999 cards
	MACRONAME	DELET	M	Delete macro
	SUBROUTINENAME	DELET	S	Delete subroutine
	MACRONAME	UPDAT	M,n	n=1 to 9999
	SUBROUTINENAME	UPDAT	S,n	n=1 to 9999
	SYSTEMS	UPDAT		Update sys ctrl or lib
	PHASE 1	UPDAT		Update system tape, Phase 1
	PHASE 2	UPDAT		Update system tape, Phase 2
	PHASE 3	UPDAT		Update system tape, Phase 3
	PHASE 4	UPDAT		Update system tape, Phase 4

Figure 4. Control and Change Card Summary

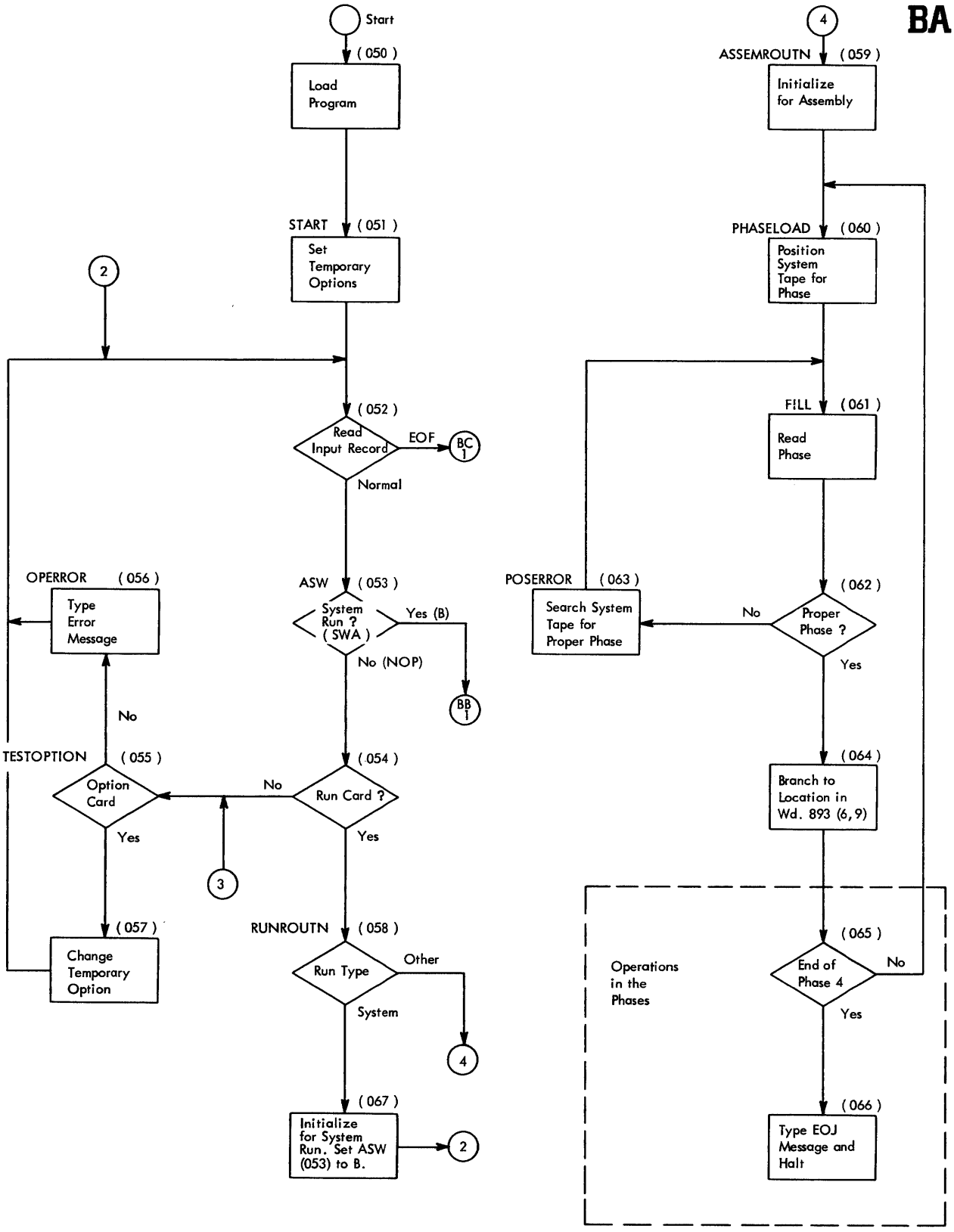


Chart BA. System Control and Librarian

BB

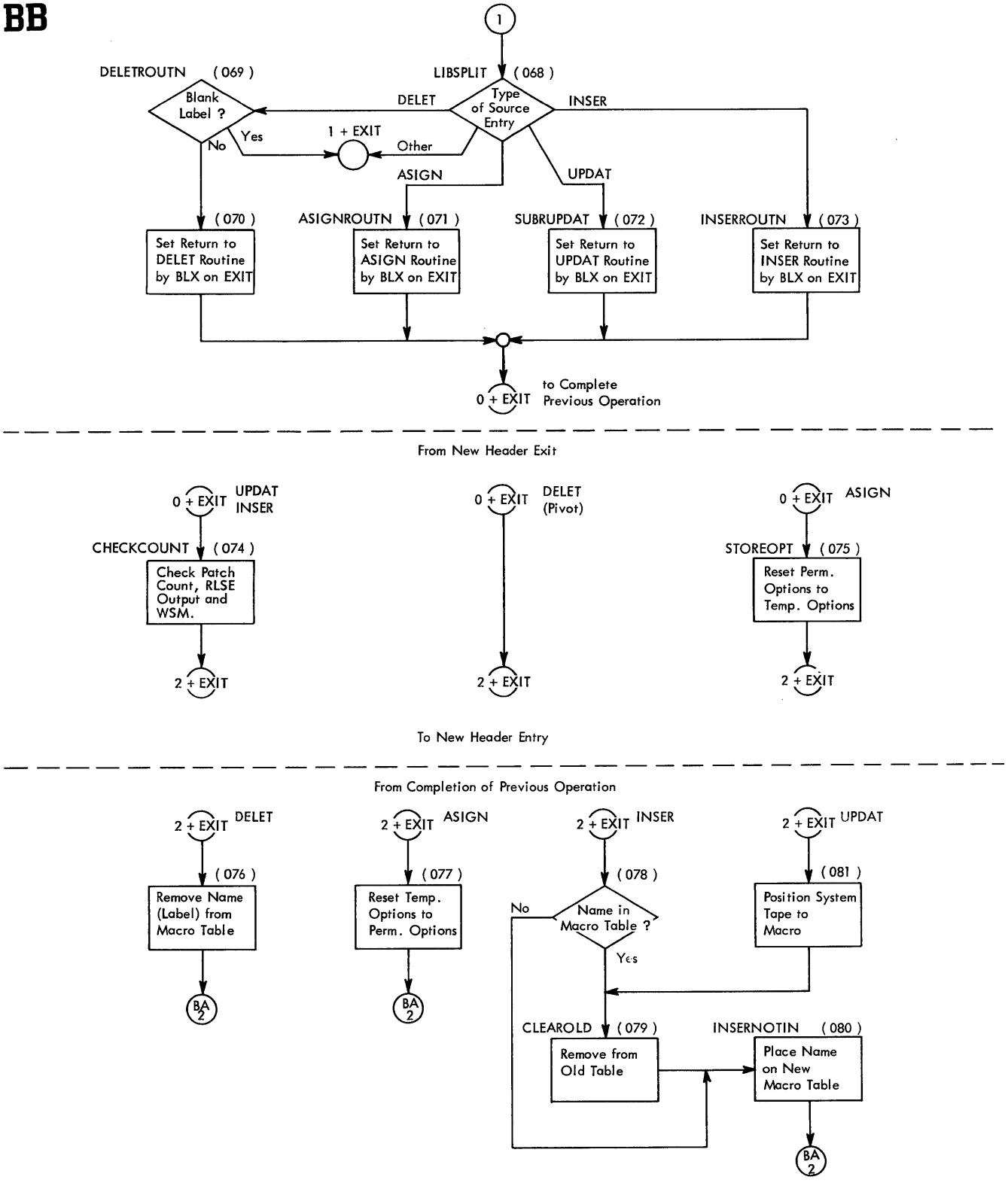


Chart BB. System Control and Librarian, Continued

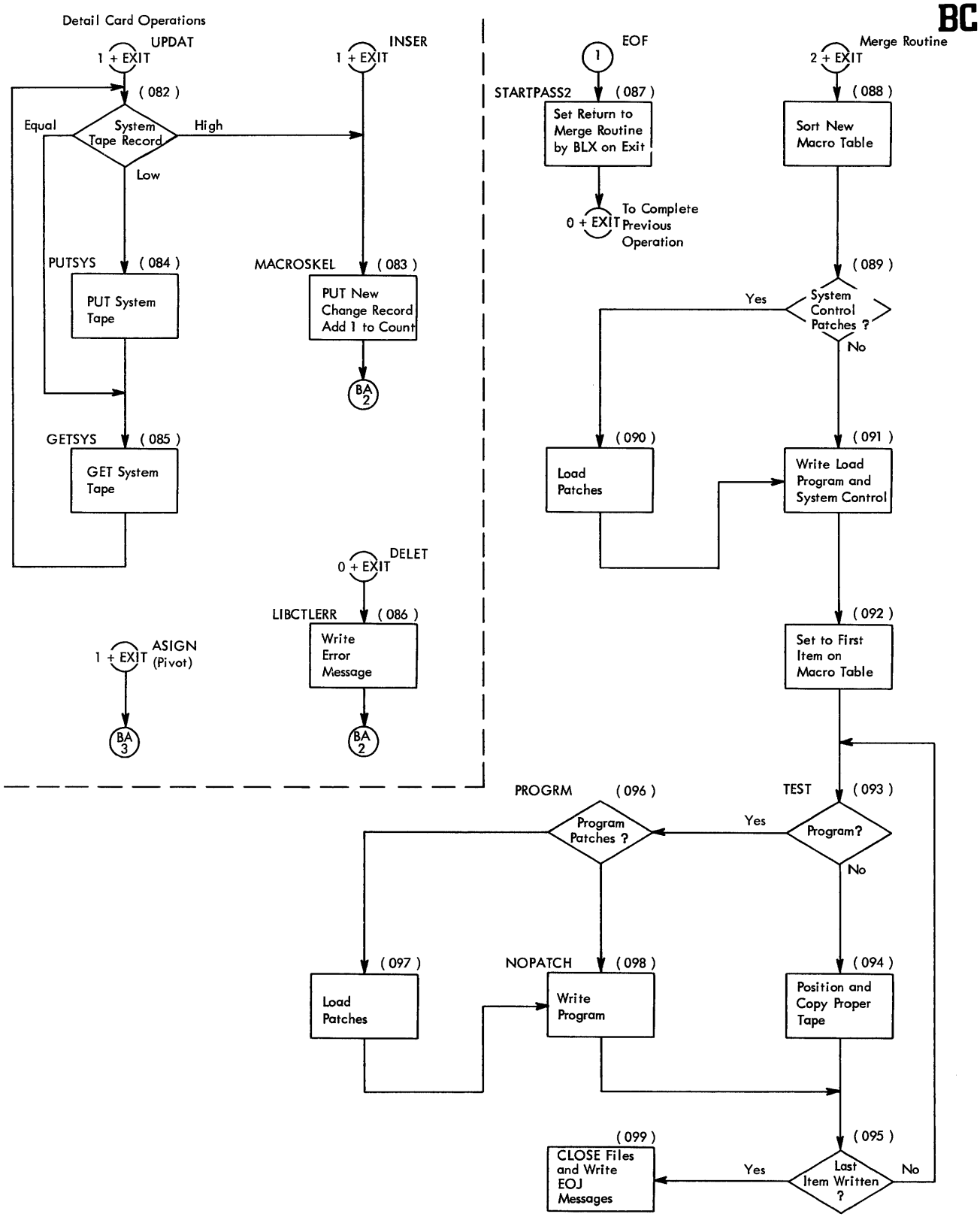


Chart BC. System Control and Librarian, Continued

Phase 1

Four-Tape Autocoder accepts two types of input records, source entries for an original assembly run and a listing tape from a previous assembly for a reassembly run. The major difference in processing between the original assembly and a reassembly is the different record-length inputs. Since a reassembly run uses the listing tape as input, it is necessary to edit each record. This is accomplished by:

1. Removing the carriage control character
2. Removing the generated entry "X"
3. Inserting the page number and identification (held from heading)
4. Removing the blanks between fields
5. Deleting any field after the comments
6. Deleting heading lines, switch and index-word tables, and literal tables

Deletions, additions, and updated records are also handled so that change records control the reassembly input. The original assembly and reassembly runs are identical in Phases 2, 3, and 4.

A control word is added to all output records in Phase 1. This word becomes word 17 of the 26-word record created in Phase 2. Some control information is placed into this word in positions 6, 7, 8, and 9 to differentiate certain types of entries in later processing.

All records are checked for sequence of page and line. If an out-of-sequence record is detected, an error message is typed.

Operation codes on input records, except for a comments record, are looked up in the following tables located in storage:

1. IOCS macro-instruction (IOTBL)
2. Macro-instructions or subroutines (MACTBL)
3. Special operations (SPECTBL) which require different processing

Instruction statements with symbolic machine operation codes are expanded to 17-word records and written on the output tape, five per block.

When a macro-instruction is encountered, the macro is located in the library, and the skeleton instructions are processed. This processing consists of replacing pseudo-labels and operands (e.g., □1, □B, etc.) with parameters from the operand of the macro-instruction. (Refer to Form C28-6102 for information on skeleton instructions.) IOCS macro-instructions are processed in the same manner, except that they usually need information from the DTF (e. g., FILEFORM, TIOAREAS, FILETYPE).

IOCS subroutines, unlike normal subroutines, are processed immediately after the DIOCS macro. The DIOCS macro equates (EQU) the first three parameters and includes (INCL) the remaining parameters as subroutines. These subroutines are one-for-one skeleton instructions that are inserted into the program.

Normal subroutines are processed similarly except that when the CALL operation is detected, the subroutine name (operand of the CALL operation) is placed on the subroutine name table (SUBNAM). Subsequent subroutine names are also placed in the table. If a CALL operand is duplicated, as additional subroutine names are placed in the table, only the first operand will be stored. The subroutines are not processed until a LITORIGIN, DIOCS or END control card is recognized or end of file (EOF) is reached.

If DA's, DRDW's (single operand), and EQU's have symbolic operands, they are entered into the symbol-request table. The symbols are assigned locations in Phase 2. This operation saves searching the symbol tape to get equivalent addresses in Phase 2.

As each DTF is processed, information that will define IOCS macros later is stored into the DTF table (DTFTBL). After the last DTF is processed, the file-scheduler macros are located utilizing information from the DTFTBL. One file scheduler is then processed for each DTF.

The following detailed explanation of the processing of Phase 1 refers to Chart CA.

Block 101, INITP1: Initialization of Phase 1 is accomplished by setting electronic switches to zero, setting up index words, calculating the macro table RDW, modifying the DTF's for the proper tape density options, and opening the tape files.

Block 102, PROOPLESS1: Entrance to this block is made from initialization and from the IOCS end-of-segment procedure in system control. Since there is a segment mark between macros in the library, EOS signals to Phase 1 that the macro has been completely processed and turns the MACROS switch off. Other switches may then control operations, or Phase 1 will set up new controls for the next operation.

Block 103, READ: The MACROS switch, if on, forces a GET from the system tape to process the next skeleton record.

Block 104, READ+1 (one word after the READ label): If the scheduler switch (SKEDSW) is on, a BES is made to the scheduler control section (starting at block 133). This switch is on while scheduler skeleton instructions

are being processed, but remains unused until the scheduler macro is completed and the MACROS switch is turned off.

Block 105, READ+2: If the insert-subroutine switch (INSSUB) is on, a BES is made to NXTSUB (block 108) which determines if more subroutines are to be processed.

Block 106, READ+3: The IOCS subroutine switch (DIOCSS) will be turned on when the DIOCS operation is recognized. After the DIOCS macro is completed, a branch is made on this switch. This BES can pass control to NXTSUB−2, where subroutine operation will be initiated.

Block 107: The insert subroutine switch (INSSUB) is turned on to initialize for the processing of IOCS subroutines. When this switch is on, it insures the immediate generation of subroutines, rather than waiting for LITORIGIN or END CNTRL. All subroutines listed in the subroutine table will be processed.

Block 108, NXTSUB: The MACROS switch is turned on, thereby forcing a GET from the library on the next read operation. All skeleton instructions are read and processed under control of this switch until the end of the macro is recognized (EOS).

Block 109: After determining that subroutines are to be processed, a BCX is made on SNTX3. This index word contains the address of the next subroutine name and the stop address of the SUBNAM table. If more subroutine names are in the table, a branch is made to locate the next subroutine in the library (connector 4).

Block 110, NOSUB: As the last subroutine is processed, the INSSUB and MACROS switches are turned off. This insures a GET from the input tape (next record to be processed) on the next read operation.

Block 111: A BSF is made on the DIOCSS switch. This branches to PROOP (process operation) to initiate a GET on the input tape and process the next record. If this switch (DIOCSS) is off, the litorigin control is moved from the hold area and a branch is made to BACK via ENDSW. The routine at BACK writes the litorigin control record on the output tape.

Block 112: A BES is made on ENDSW. If this switch is on, an end control or end of file on the input tape was recognized. The branch is made to the end-of-phase routine.

Block 113, EOJEOJ: Close procedures start. The RDW of the symbol-request table is calculated. The files except for the system tape are closed. An end-of-phase message is typed and a branch is made to system control. This brings in Phase 2 and starts operation for that phase.

Block 114: A BES is made on EOJSW. If this switch is on, an end of file has been recognized on the input file and a branch is made to initialize the end-of-phase procedures.

Block 115: An input record is read. The page and line number is sequence-checked, and if an out-of-sequence record is detected, an error message is typed. If end of file is recognized by IOCS, the end-of-file exit (taken from the DTF) is used. This branches to the end-of-phase initialization procedure at block 116.

Block 116: After recognition of end of file, a branch is made (by IOCS) to turn on EOJSW. This switch causes a branch to the end-of-phase initialization after writing the last record in the work area.

Block 117, PROOPLUS2: A comparison is made between the first character in the label and an asterisk. If it is equal (comment card) a branch is made, by passing the record testing procedures to the write operation at block 130.

Block 118: A look-up is made in the IOCS macro table (IOTBL) on the record's operation code. This is necessary since IOCS macros require special handling. For instance, a GET macro-instruction would require knowledge of the file characteristics to put out the proper GET macro. If the macro-instruction name is found in the table, a branch is made to block 119 to turn on the IOCSS switch.

Block 119: The IOCSS switch is turned on to differentiate between IOCS macros and other macros.

Block 120: If the macro-instruction is not of IOCS, the macro name is looked up in the macro table (MACTBL) and, if found, a branch is made to block 121 to process this macro.

Block 121, NOMORE: The MACROS switch is turned on to force the GET on the library the next time the READ routine is entered. This reads the first and succeeding skeleton instructions until the macro is completely read and processed.

Block 122: A BSF is made on the IOCSS switch to initiate the proper IOCS macro procedure.

Block 123, BLDHDR: The operation code is compared to DIOCS. If it is equal, a BE is made to DIOCS at block 124.

Block 124, DIOCS: The DIOCSS switch is turned on to initialize processing of the IOCS subroutines after the DIOCS macro has been completed. This macro inserts the EQU's for the IOCS index words and the INCL's for the subroutine names. The parameters in the DIOCS operand are placed into the operands of the EQU's and INCL's.

Block 125: The macro operation code and operands are scanned and analyzed to determine the name of the IOCS macro required, and the headers for this IOCS macro are set up.

Block 126, CKSPEC: Since the operation code has not been found in the IOCS macro (IOTBL) or macro table (MACTBL) a look-up-equal is made in the special opera-

tion table (SPECTBL). These operations require special processing. The second word of each word-group in the SPECTBL is a branch to the address of the routine used for the individual operations. Therefore, after the LE, a branch to 1+X98 is made. When the operation is a DTF, a branch is made to the start of the DTF procedures at block 131. If CNTRL, it branches to test the CNTRL card label. When the operation code is INCL, a branch is made to block 138 to include this operand in the subroutine table (SUBTBL). If the operation code is DA, EQU, or DRDW, the branch is made to process its operand at block 129.

Block 127, CNTRL: The first word of the CNTRL card is compared to LITOR. If they are equal, a branch is made to start processing of the subroutines at block 107. If they are not equal, the first word of the label is compared to "ENDbb." If they are equal, a branch is made to start end procedures at block 128.

Block 128, ENDCD: An ESN is made on ENDSW, allowing an exit to end-of-phase procedures after completion of subroutine processing. A branch is then made to subroutine initialization at block 107 via connector 5.

Block 129: The operand of the DA, DRDW, or EQU is tested to determine if it is symbolic. If it is, this symbol is entered on the symbol request table. This table will be used in Phase 2.

Block 130, BACK: The record is PUT to the output file. IOCS blocks the records and, when a block is filled (five records), writes it on the output tape.

Block 131, DTF: Having recognized the first entry of a file's DTF in block 126, the entry is PUT to the output file. A DC for this DTF is then generated and written on the output file.

Block 132: The DTF records are obtained under control of a table of entries (TFCTL). These entries contain factors which determine if the operands of the DTF records are necessary to generate the type of scheduler needed for the file. All information that will contribute to generation of later IOCS macros is stored into specific positions of the DTF table (DTFTBL).

A BIX on the RDW of TFCTL determines when the last entry has been processed. After all DTF's of a file are processed, the necessary EQU's for that file are generated. The operation code of the next record is then tested. If the operation code is DTF, a branch is made to generate the next headers at block 131. If the operation code is not DTF, a branch is made to generate a file scheduler for each file under control of blocks 133 and 134.

Block 133: A BCX on the DTF table RDW is made which indicates that there are more schedulers to be generated (block 134) when the branch is made. The start address in this RDW is stepped after the finish of each scheduler macro. If the last scheduler has been generated, SKEDSW is turned off and a branch is made to continue operation via connector 2.

Block 134, GNNXTS: The SKEDSW is turned on to insure a return to the scheduler routine after each scheduler macro is completed. The scheduler's name is generated from information in the DTF table (DTFTBL), and the headers are written out. The MACROS switch is turned on, and a branch is made via connector 4 to find the proper macro.

Block 135, MACPRO: The location of the necessary macro is computed, and the system tape is spaced to this macro.

Block 136, MACRO: A GET is made, thereby making available a skeleton instruction from the library. The label, operation, and operand of each of these skeleton instructions are scanned. Whenever a lozenge is detected, the next character is checked. If the next character is 1 through 9, the parameter (from the macro-instruction operand) for this number replaces the lozenge and number. If the parameter is missing from the operand, an error message is written.

If the character following the lozenge is A through I, the operand is searched and, if the parameter is found, it is put into place. If it is not found, the skeleton instruction will *not* be PUT to the output tape. If the character was neither 1-9 nor A-I, a symbol is generated and put in place of the lozenge and letter. This symbol has the format MACRO*nnnny*, *nnnn* being the macro number in the library and *y* being the second digit of the double-digit character following the lozenge.

The completed skeleton record is then PUT to the output tape if it was not deleted. Control passes to the subroutine-name detection operation.

Block 137: The operation code is compared to INCL. If they are equal, a branch is made to test the subroutine name.

Block 138, INCLRT: The operand of the INCL record is looked up in the macro table (MACTBL). If it is not found, a branch is made to write an error message.

Block 139: A message is written indicating that a subroutine has been used which is not in the library.

Block 140: Since the subroutine name to be included is located in the macro table, it is added to the subroutine name table (SUBNAM) and the RDW of this table is adjusted to provide for the addition. Control passes via connector 8 to BACK where the INCL record is written on the output file.

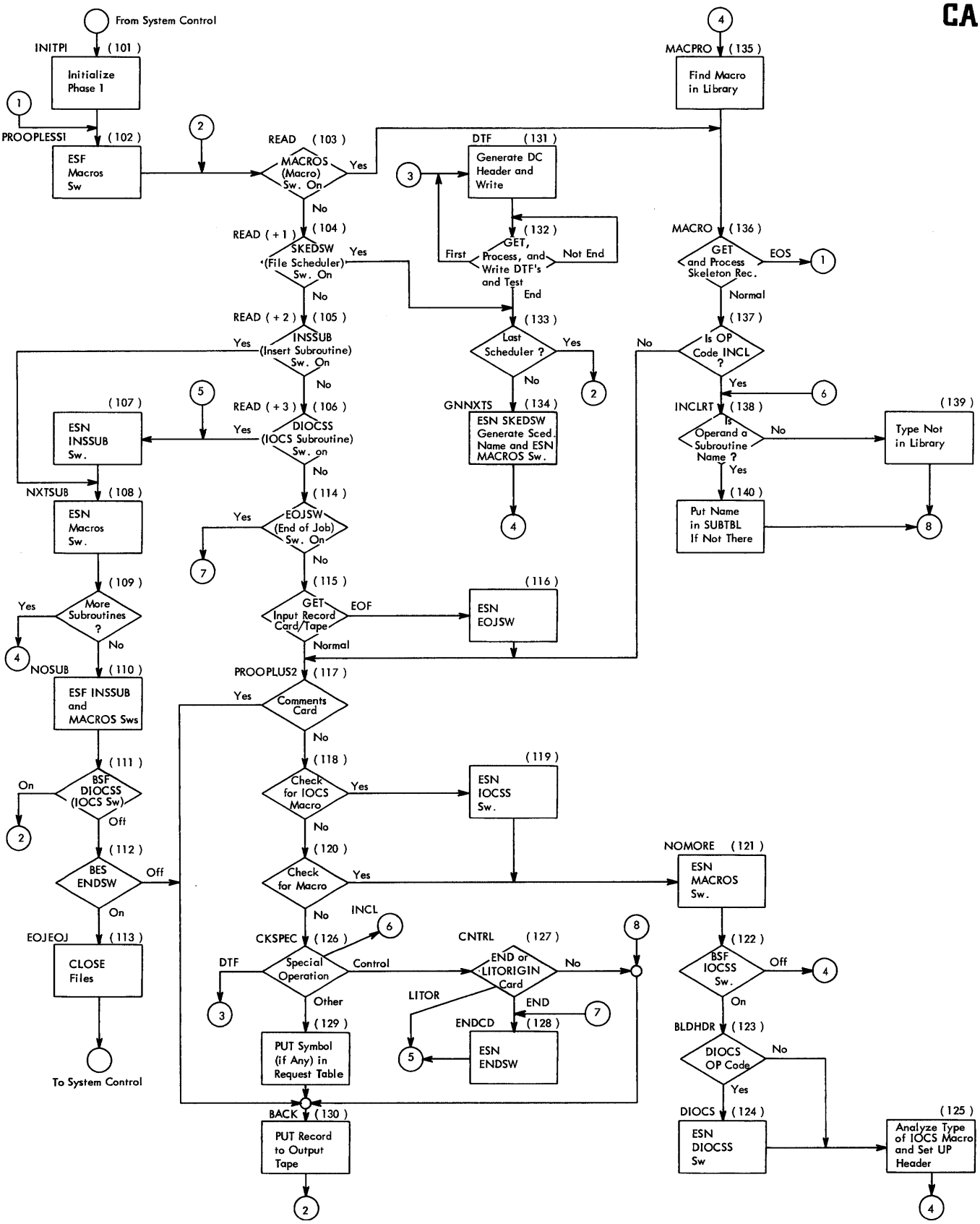


Chart CA. Phase 1

Phase 2

The processing in Phase 2 includes the following operations:

1. Storage locations are assigned to all statements that produce machine instructions.
2. Storage locations are computed and assigned for declarative statements.
3. Blocks of symbols and their equivalent addresses are written on tape for subsequent processing.
4. The operation code is processed for imperative instructions.
5. The symbol request table is processed and used in obtaining equivalent addresses of symbols which must be previously defined; i.e., the label of a statement earlier in program sequence.
6. Statements with actual addresses, including asterisks, are processed.
7. A 26-word expanded record is developed for each statement.
8. The operands of the statements are examined and the symbolic fields are placed into specific words of the expanded (26-word) record.
9. Switches are set that will control subsequent processing for incompletely processed statements.
10. Actual index words and electronic switches are processed and reserved in availability tables.

Much of the processing in Phase 2 is controlled by a code word for each operation. These code words are located, for a specific operation code, by a look-up in the operation-code table. The control digits in the code word are placed into index words, allowing indexing on branches to dictate statement processing. Processing is further determined by scanning the operand and analyzing the characters found. For instance, a plus sign after an address would indicate need for processing an index word or address arithmetic. The operands are processed by a group of subroutines. Entry to and from the subroutines is made by a BLX on specific index words.

The detailed description of Phase 2 refers to Charts DA and DB.

Block 201, INIT: Initialization of the phase is accomplished by zeroing out electronic switches and index words, setting the availability table of index words and switches to ones, modifying the DTF's for the proper tape density option, and opening the files to be used.

Block 202, GETREC: A BLX is made to GET a record. The record is then moved to the work area. If EOF is recognized by IOCS, all records have been processed and a branch is made from the EOF exit to the end-of-file routine, block 236. This address is obtained by IOCS from the input DTF.

Block 203: The type code (word 17, digit 9) of the input record is compared to 6. If it is equal, processing is bypassed, and a branch is made to block 235 to PUT the record to the output file; e.g., macro headers or INCL cards need no further processing.

Block 204: The contents of the location assignment counter (LOCTR, positions 2-5) are placed into the expanded record, word 18, positions 6-9.

Block 205, PROOP: A LEH and a LE are processed with the operation mnemonic against the operation-code table. The double look-up is used to reduce average look-up time. The first look-up (incremented by 28) finds the approximate location. The LE has its RDW altered to this small area and finds the exact operation if it is in the table. If it is not found, a branch is made to NOOPER (no operation code) at block 206.

Block 206, NOOPER: The operation code has not been found. Therefore, an error message is written and the operation code is changed to NOP. A branch is then made to the operand processing routines starting at block 218.

Block 207: The operation has been found and is tested to determine if it is a declarative entry (DA, DC, DRDW, etc.) or a control. In these cases, the word after the mnemonic in the operation-code table is an unconditional branch to a different routine for each type of entry. This branch address is used to enter a specific routine for processing.

Block 208, PSEUDO: Since each declarative entry uses a separate main routine for processing, a description of each routine follows.

CNTRL, CNTR: The label is processed as a normal label. (For greater detail, see block 234.) The first word of the label is then compared to the constants BRANC, ENDbb, LITOR, and ORIGI. Branches on equal are then made to separate routines for each type of control. If the label is not equal to any of the legitimate control labels, a label error message is written.

BRANCH: A branch instruction is set into the assembled instruction field of the expanded record (word 19), and the record is written on the output tape.

END: A branch instruction is set with the address in the operand or 0325 for blank operands, and the record is written on the output tape. Control then passes to the end-of-file procedure.

LITOR: If the first word of the control label is LITOR, the location counter is placed in the litorigin table and the litorigin counter (LORCTR) is stepped one. The record is then written on the output tape.

ORIGIN: The processing is determined by the type of operand. If the high counter (HICTR) is lower in value than the location counter (LOCTR) and the last origin control did not have ",s" in its operand, HICTR is set to LOCTR. The operand is processed and, if it is blank, the location counter is set to the high counter. If it is numeric, the actual location is set into the location counter. If it is symbolic, the symbol is looked up in the symbol table. If it is not found in the table and there was a symbol request table overflow, it is looked up on the symbol tape. If the symbol was not found, an error message, "NOT PREVIOUSLY DEFINED," is written. Its equivalent is then set into the location counter, the record is written on the output tape, and the next record is read and processed.

DA: The number of RDW's to be generated (first item in operand) is picked up and stored in index word *N*. The label is scanned, and, if blank, is assigned the page and line number and a number from the NOLABELCTR. After this counter is used, it is incremented by one to insure assignment of distinctive labels. The proper number of RDW's are generated and given their proper signs (defined by the second entry in the operand). The entries after the DA are then examined for a blank operation code. If they are blank, they are processed as fields of the DA. A test is made of the second field-defining number of the operands. The highest number is held, and will indicate the number of storage words to be reserved. Labels, when detected, are given assigned locations and placed in the symbol table. All records are written on the output tape. When an entry is found with the operation code not blank, the number of RDW's (*N*) is multiplied by the number of words per area (*W*) and added to the assignment counter (LOCTR). The DA label, *N*, *W*, and the DA address are placed in the symbol table. A test is made of the area location. If the area falls in the index word range, those index words are reserved on the index-word availability table.

DC: The DC routine is entered on recognition of a DC operation code. This routine writes the header line and places the label and its assigned location in the symbol table. If the entry after the DC header contains a numeric constant, a test is made for the presence of field definition. If field definition is detected, an attempt is made to pack the entries into the least number of words possible. Additional words are assigned only when the sign of the entries changes or if successive field definition would overlap when assigned the same word. DC processing is concluded for all types of entries by recognition of other than a blank operation code. Alphameric constants are recognized by the preceding @ character, and are placed in groups of five double-digit characters, each group being assigned a new word. When an @ character is recognized during the scan, the next character is tested for another @ character. Succeeding @ characters are placed in the assigned words since this character is legal if it immediately precedes the final @ character. If the character after the @ is an R, a record mark is placed in the low-order position of the last assigned word, provided the positions remain unused. If they are used, a new word is assigned for the record mark. After the last @ is recognized in the operand, a new record is read for processing.

If the constant has an alphabetic character after the sign (ADCON), it is considered a four-digit numeric constant and the symbol is placed into words 20 and 21 of the expanded record. It is then packed, if field definition (first character of the field is left parenthesis) follows the symbol. If field definition does not follow numeric constants or ADCON's, each constant is assigned to the low order of a separate word.

DRDW: The DRDW routine processes two general types of DRDW's. One type contains two addresses (symbolic or actual) in the operand defining a given area of storage. The other has a single address that must be a label of a DA or DC appearing earlier in the sequence of instructions. For the DRDW, this routine generates *N* number of RDW's. (*N* is the number of RDW's given as the first parameter of the DA operand.) For a DC operation or a DA having no given first parameter, *N* is equal to 1. *N* is determined by a look-up on the symbol-request table for the symbol in the operand. Positions 0-2 of word 3, for that symbol in the table, contain the factor *N*.

EQU: The equate routine tests the operand of the EQU for an actual or symbolic value. If it is actual, the character after HCHAR (character which stopped the scan) is tested to determine if it is S or X. If it is an S or X, indicating an actual electronic switch or index word, the component is reserved on the electronic-switch or index-word availability table. If the operand is symbolic, the symbol is looked up on the symbol request table, and the assigned location with the operand label symbol is placed in the symbol table. If the symbol of the operand was not found on the symbol request table, communication word 96 is tested to determine if the symbol request table was filled in Phase 1 (signed minus). When it is full, a look-up is made of the symbol tape via the symbol table. After the search is completed, the symbol tape is repositioned properly and processing is continued.

Block 209: The operation code is obtained from the sign and positions 0 and 1 of the word following the mnemonic in the operation code table (OPTBL). This code is placed into the assembled instruction field (INST) in word 19 of the expanded record. Positions 4-9 are also stored and become the modifiers for augmented instructions.

Block 210: A test is made of D1 (placed into index-word DIGIM). If this digit is other than 0 or 3, a component is necessary to complete the instruction, and control passes to a scan starting at block 211.

Block 211, SCANR: The operand is scanned by a scan subroutine via a BLX. The linkage return from this routine is via SCANRX. If the first character is numeric, it is considered actual and control passes to block 214 by a branch to 1+SCANRX to test the range of the component. If the first character is alphabetic, control passes to block 213 on a branch to 2+SCANRX. If the character is not numeric or alphabetic, control passes to block 212 by a branch to 0+SCANRX (illegal component). All scanning is accomplished by compare-digit operations on the double-digit characters in the operand. The scan is continued until the end of the field is reached (change in character type; e.g., numeric to alpha or alpha to special character).

Block 212: Since the operand of the component is illegal, the message "OPERAND ERROR" is written.

Block 213: The symbolic component is stored into words 25 and 26 of the 26-word record. Also, D1 which has been stored in DIGIM, position 5, is placed into switch 1, word 17 of the 26-word record.

Block 214, INRNG: The range of the actual component is tested in one of two routines. Electronic-switch and index-word ranges are checked to determine if they are over 30 or 99, respectively. All other components are tested against a constant (maximum value). This constant is determined by the value in DIGIM, this digit varying with the type of component.

Block 215, RNGERR: If the actual value of the component was not within the limits prescribed by the range test (block 214), the out-of-range message is written.

Block 216: If the component is an index word or electronic switch, it is reserved on the index-word table (IWTABL) or electronic-switch table (ESTABL). These tables are created in storage in Phase 2 and consist of 13 words of +1111111111. Reservation is made by changing the 1, representing the specific index word or switch, to a 0 (zero).

Block 217: The component is inserted into the assembled instruction (word 19 of the 26-word record). Control is then passed to block 218, where the operand is further processed.

Block 218, DIG2: The proper controls for processing each type of operand are set by using a branch to REGN+DIGIT. The indexing portion of DIGIT had been previously set with the character D2 (position 3, word 2 of the operation in the operation code table). Electronic switches 14-20 are the major controls for processing operands. They are set to their proper status by a ZA and STD operation that makes word 102, positions 3 through 9, equal to a mask. This mask varies for the type of operand being processed.

Block 219: A BLX is made to SCANR on SCANRX. The SCANR (scanner) routine scans the next operand entry. If the operand is numeric, a return is made via 1+SCANRX to block 225, bypassing operations on literal or symbolic operands. If the operand is a special character, a branch is made using 0+SCANRX to block 221 to test further for a literal. If the operand was alphabetic, a branch is made using 2+SCANRX to process the symbol at block 220.

Block 220: The operand entry has been recognized by the SCANR routine as a symbol. This symbol is placed into the expanded record in words 20 and 21 (SYMAD and SYMAD+1). Switches 2 and 4 (word 17) are set to 1.

Block 221: The operand entry was recognized as a special character. This character is compared to an asterisk. If they are equal, a BE is made to asterisk processing at block 222. If they are not equal, control passes to block 223 for further testing.

Block 222: The operand entry is an asterisk. The value in the location assignment counter (LOCTR) is placed into accumulator 3 and replaces the asterisk in further operations on this statement.

Block 223: A test is made to determine if a literal is allowed (LITSW on). If a literal is not allowed, a branch is made to SRO to type operand error. If LITSW is on, control passes to block 224.

Block 224: The literal is processed by comparing the first character to an @ symbol. If the comparison is equal, indicating an alphameric literal, the literal is scanned and counted. The literal is placed in SYMAD (word 20 of the expanded record) and its length is placed into SYMAD+1. If the comparison to the @ symbol was unequal, the first character is tested for a sign (plus or minus). If a sign is recognized, the second character is scanned. If it is numeric, the remainder of the field is scanned, and the numeric literal and its length are placed into SYMAD and SYMAD+1 (expanded record). If the scan of the second character indicated alphameric, the literal is an ADCON. The literal is placed into SYMAD and SYMAD+1. Switch 4 (position 3, word 17 in the expanded record) is set to 3. Control then passes to block 225 via a BLX on FDX.

Block 225, FD: A test is made of the character that stopped the last scan. If it is a left parenthesis, the field definition is examined by stepping and testing. The stepping is done by the SCANR and AC (advance character) routines. The field is tested for single and double-position definition. If single-position definition, (n) format, is detected, the digit contained within the parentheses is placed into field definer positions 1 and 2. If double-position definition, (n, x) format, is recognized, each digit is placed into its corresponding definer positions. Control then passes to ENDX to determine if the operand has been completely processed.

Block 226: If the statement being processed can contain address arithmetic, recognition of the sign forces a test of the next character. If this character is numeric, indicating address arithmetic, control passes to block 227. If the test indicates alphabetic, control passes to block 228 to process indexing. Recognition of a special character forces control to SRO to initiate an error message.

Block 227: The characters after the sign were recognized as numeric (address arithmetic). The address arithmetic is stored into the expanded record at positions 6-9 of word 22. Four is added to switch two and control passes to ENDX (block 231).

Block 228, TXNN: The first character after the sign is compared to x. If they are equal, further tests are made to determine if the field is an actual address no longer than two characters. If the field meets the conditions, control passes to X1B (block 229). If any character being examined does not follow the x*nn* format, control passes to block 230 (NOTXNN).

Block 229, X1B: The actual index word is stored, and the index word indicated is reserved in the index-word availability table. Control then passes to ENDX.

Block 230, NOTXNN: The symbolic index word is stored into words 23 and 24 of the expanded record. Switch 3 is set to 1 and control passes to ENDX (block 231).

Block 231, ENDX: The character that stopped the scan is tested for a blank. If it is a blank, it indicates the end of the operand. The information that was processed is placed into the assembled-instruction field (word 19) in the expanded record. Control then passes to the test for label processing at block 232. If the next character is not blank, a return is made to 0+FDX to continue processing.

Block 232, PRLAB: The label is scanned to determine if it is blank (no label). If it is blank, the location assignment counter is stepped and control passes to PUTGET to get the next record.

Block 233: If the label is actual, it is compared to the object-storage size (communication word 95) and, if smaller, it is placed into the expanded record, positions 6-9 of word 17 (assembled instruction). If the label is alphameric and follows the rules for symbolic labels, the label is looked up in the symbolic request table. If found, the symbol (two words) and its equivalent (actual location) are placed into the symbol table. The symbol-table counter is then stepped three to the next symbol position. When the symbol table becomes full it is written on the symbol tape (SYMTAP).

Block 234, STEPP: The location assignment counter is stepped 1 in preparation for the next entry. Control then passes to block 235.

Block 235, PUTGET: The expanded record is written on the output tape, and control passes to block 202 to get the next record.

Block 236, EOF: The input and symbol tape files are closed. The output file is released in preparation for possible error messages during the symbol randomization in Phase 3, pass 1. The "End Phase 2" message is typed. If alteration switch 1 is on, or after a start, control is passed to system control to bring in the next phase.

Block 237, SRO: An operand error is recognized. A message indicating this event is written, and control passes to ENDX to determine if the end of the operand has been reached.

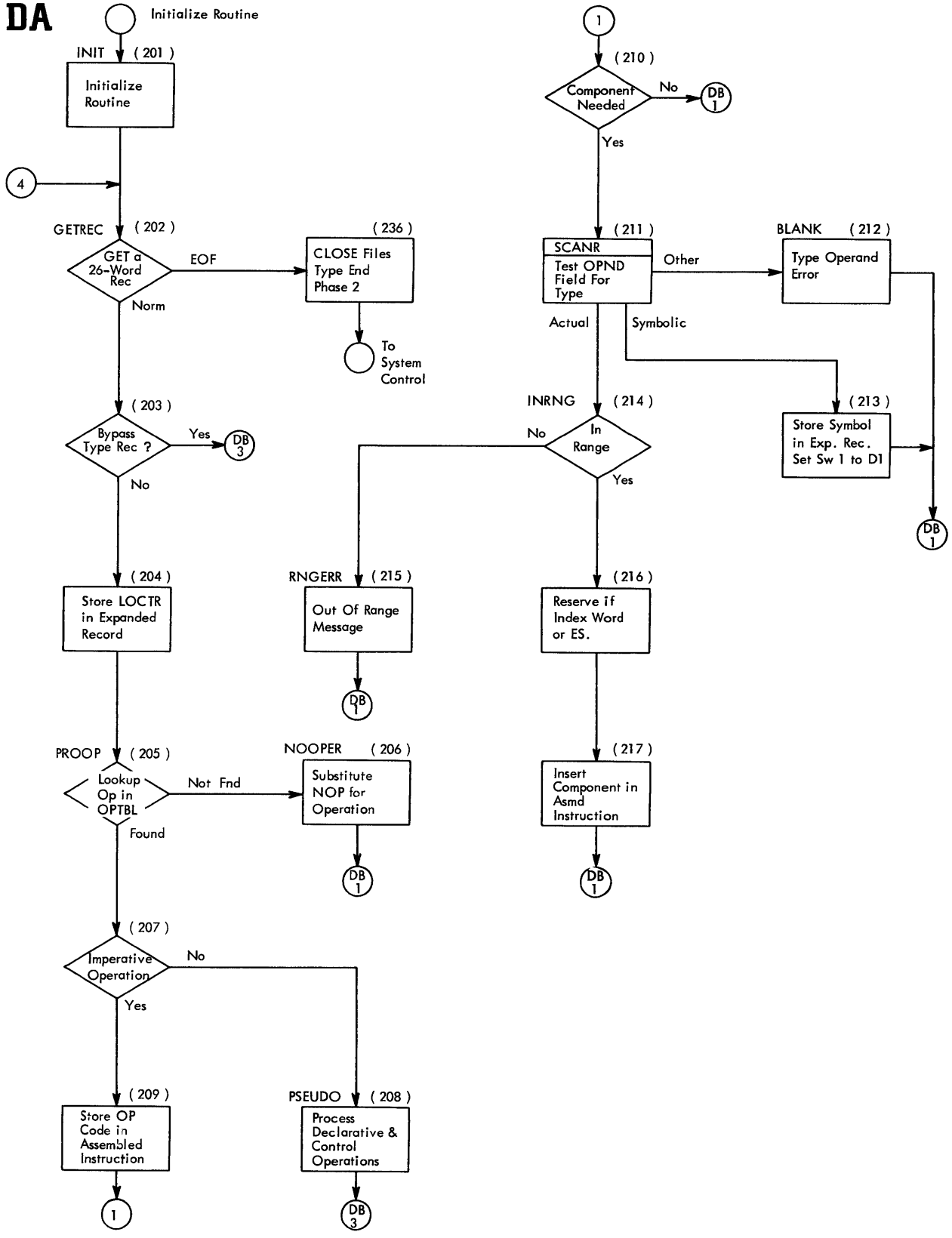


Chart DA. Phase 2

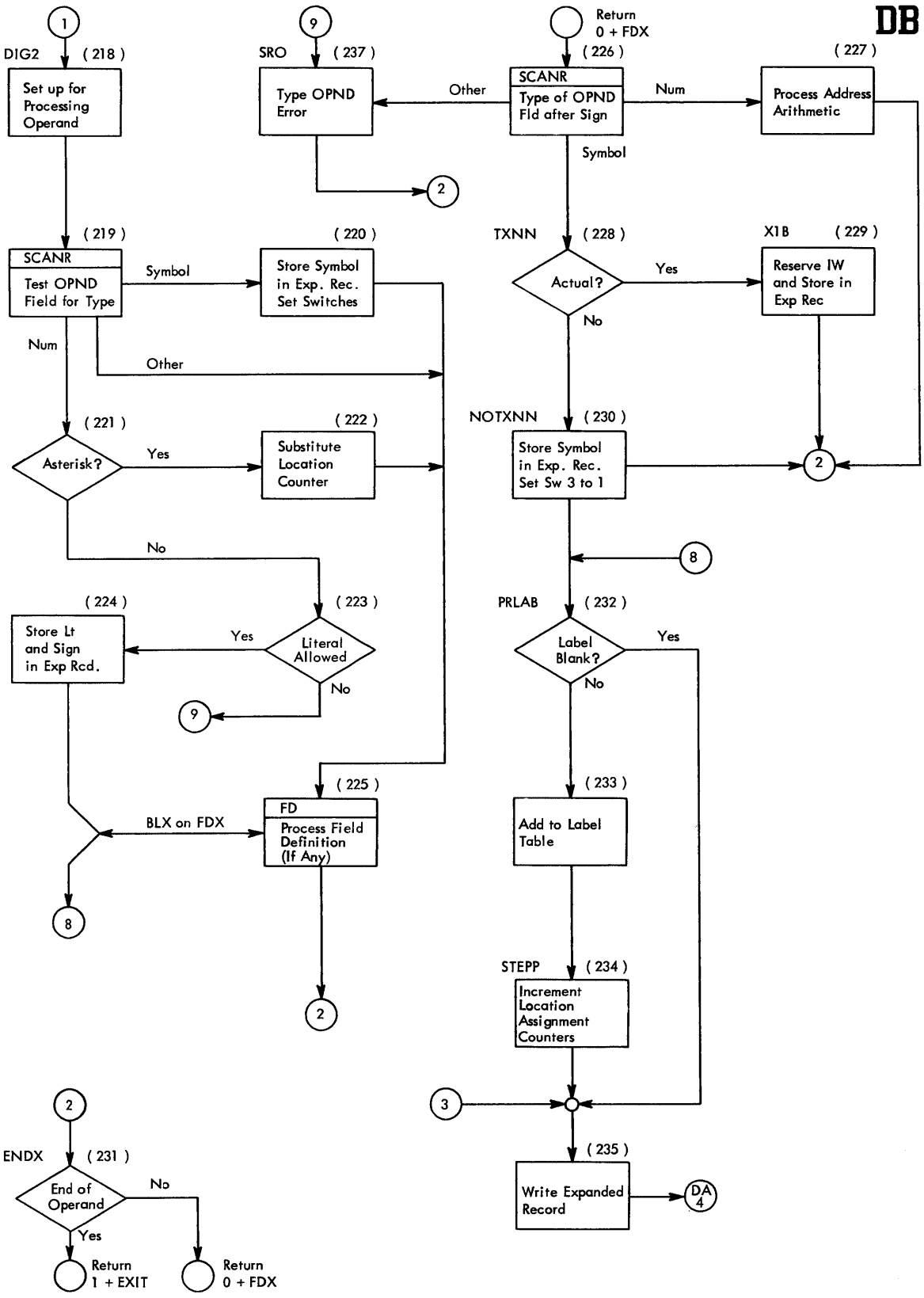


Chart DB. Phase 2, Continued

Phase 3

Phase 3 completes the assembly of each instruction except those with operands containing symbolic index words and electronic switches. The symbolic label table created in Phase 2 is used for Phase 3 processing. In general, Phase 3 is processed as follows:

1. Labels from the symbol tape are placed into the storage label table in random locations. Since storage is limited, extra labels are placed on another work tape for the next pass of Phase 3.
2. The symbolic operands from each of the source program statements are randomized to locate a given symbol in the label table.
3. The location assigned to the label and any field definition are then placed into the assembled instruction.

The randomizing method for locating symbols in a label table is used in contrast to a sort or table look-up to save processing time. The formula for determining the random location in the label table for a given symbol is:

$$3\left[\text{remainder of}\ \frac{\text{word 1} + \text{word 2}}{\frac{\text{stop} - \text{start} + 1}{3}}\right] + \text{start}$$

Word 1 and word 2 refer to the symbol. Stop and start refer to the addresses defining the symbol table (SYMTAB) being filled for a given pass in Phase 3.

Symbols that define index words or electronic switches are placed in an index word and electronic switch table for processing in Phase 4. Index words cannot be assigned until all the labels are processed and the index-word availability table is complete. Also, electronic switches cannot be assigned until the electronic-switch availability table is completed after Phase 3.

The number of passes to completely process Phase 3 is determined by the number of symbols. Each pass constitutes the processing of all expanded records against the label table in storage.

The detailed description of Phase 3 operation refers to Charts EA and EB.

Block 300, BEGIN3: In the initialization of Phase 3, the channel number is picked up from the load program (location 310, digit position 1) and placed in the object output (OBJOUTPUT) DTF and symbol-output (SYMOUTPUT) DTF. Also, all the electronic switches are cleared, the range table is set to object size (index word 95, positions 6-9), and the DTF's are modified for the proper tape density option.

Block 301, INITLPASS: The symbol table, located in storage from the end of the Phase 3 program (approximate location, 3535) to the end of storage defined by EQUIPSIZ, is cleared to plus zeros in the initialization of each pass. The symbols in the symbol table are placed in storage at a calculated random address.

Block 302: A test is made to determine if sufficient storage space is available for all the symbols contained on the symbol tape.

Block 303: If there is sufficient storage space for all remaining symbols on the symbol tape, the last pass switch is turned on, indicating that the current pass is the last.

Block 304: The randomizing factor (RAMFACTOR) is calculated. This factor is that part of the randomizing formula which is common for all symbols. After calculation, it is stored in index word 93.

Block 305, GETSYMREC: A symbol block, consisting of a maximum of 43 symbols, is read into a 130-word input area. An end-of-file condition occurs when all of the symbols contained on the symbol tape have been processed, causing a branch to block 320. Note that an end-of-file condition is the sole exit from the processing of symbols to the processing of the 26-word records. When this exit is used, the object-output file in the previous pass is closed.

Block 306, GETNEXTSYM: The next symbol is placed into the accumulators by indexing with IXWA02 (symbol file RDW). This index word is incremented by three (number of words assigned to each symbol) before a return is made to the beginning of this processing loop.

Block 307, RANDOMIZE: The symbol is randomized using the randomizing factor (calculated in block 304).

Block 308, TESTFULL: The calculated address is tested to determine if it has been filled previously. This test is a CSP which would result in an equal comparison if the address were empty (the sign would be alpha, if filled). If the result is equal, a BE passes control to EMPTY (block 309).

Block 309, EMPTY: The symbol and its assigned location (three words in all) are stored at the calculated address of the symbol table. A branch is then made to determine if all records in the input block have been processed (block 318).

Block 310: Since the calculated random location is already filled, a test is made to determine if the symbol occupying the location is equal to the one just randomized.

Block 311, DUPSYMBOL: Since the symbols are identical, the addresses of the symbols are compared. If they are equal, control passes to block 318.

Block 312: An unequal comparison of the equivalents indicates an error. Therefore, a DUPLICATE SYMBOL message is PUT to the object-output file of the previous pass. If the options were given, the message would also be typed or printed.

Block 313: The calculated position is filled, and the symbol is different from the one already stored. A decision is made to determine if the symbol should be forced into the table. It is desirable to fill the symbol table as much as possible to attempt to reduce the number of passes in Phase 3. The STORANYWAY switch performs this function. This switch is turned on at the beginning of each pass. On all passes except the last one, it is turned off when the table is approximately 80 per cent full for 10K or 90 per cent for 5K. During the last pass, the switch is left on to force the remaining symbols into the table.

Block 314, BUMPADDR: The address of the symbol is incremented by three (number of words per symbol), and a return is made to TESTFULL (block 308).

Block 315: The symbol and its equivalent are stored in the output area, and the counter is stepped by three, making the next area in the output field available.

Block 316: The counter for the output field (SYMOUTCTR) is tested with a BCX to determine if the output area is full. If the area is full, control passes to block 317.

Block 317, PUTSYMBOL: Since the output area is full, a PUT is made to that file to write the record.

Block 318, BUMPGET: After each symbol has been processed, a test is made to determine if all symbols (43 at most) of the input block have been used. If they have not, a branch to block 306 occurs to pick up the next symbol. Otherwise, a branch to block 305 is made to GET the next block.

Block 320, GETOBJINPT: An input record is gotten for processing. End of file on the symbol tape (block 305) starts the second part of each pass. The second part does the processing of the 26-word records, using the symbol table.

Block 321: The first check for each 26-word record is a test of the four indicator switches. If they are all off (zero), the record is completely processed and is written on tape (block 329).

Block 322: If all the switches are not off, they will be tested individually. Switch 1 is tested. If it is off, the processor immediately tests switch 2. If switch 1 is on, its value must be determined.

Block 323: The value of switch 1 is determined. If it is equal to three, the compare-digit switch is turned on prior to the testing of switch 2. A switch-1 value of 8 or 9 causes the symbolic DRDW start address, contained in words 23 and 24 of the expanded record, to be looked up on the symbol table. If the symbol is not found, the processor tests switch 2, leaving switch 1 as it was.

Any value of switch 1 other than 0, 3, 8, or 9 signifies a symbolic component. The symbolic component is picked up from words 25 and 26 of the expanded record, randomized, and looked up on the symbol table. If the symbol is found, its assigned address is placed in the correct digits of the assembled instruction, and switch 1 is turned off.

If the symbol is not found, and it is not the last pass, the processor goes directly to test switch 2. If, however, it is the last pass, and the symbol is not found, a symbolic index word or switch is looked up on the index-word and switch-name table. If found, it is picked up and the values are stored in the correct digits of the assembled instruction. Switch 1 is then turned off. If the symbolic index word or switch is not found, it is assigned an index-word location or electronic-switch number and entered on the index-word and switch-name table.

Block 324, TESTSW2: Switch 2 is tested. If it is off, the processor immediately tests switch 3. If it is on (other than zero), control passes to block 325.

Block 325: Since switch 2 is other than zero, switch 4 is tested. If switch 4 has a value of either 0 or 2, signifying a literal, the program branches immediately to test switch 3 (block 327). If switch 4 is neither 0 nor 2, processing continues at block 326.

Block 326: The two other possible values of switch 4 (1 or 3) cause the symbolic address to be picked up from words 20 and 21 of the expanded record. The symbol's random address is calculated, and the symbol is looked up on the symbolic table. If it is not found, the processor goes to check switch 3 (block 327). If it is found, the first check made is to determine if the symbol is a DA or a DC label. It is given a field definition of 0,9, if no field definition is carried in the equivalent for the symbol. The processor then checks for address arithmetic.

If the symbolic address is not defined by a DA or DC label, it is tested to determine if the symbolic address represents a switch. If so, the switch number to which it has been assigned must be converted to an actual address. As an example, assume the instruction is STD1 MRSWITCH, MRSWITCH having been previously defined and assigned as electronic switch 2; i.e., position 1 of word 0101. The symbolic address MRSWITCH must be converted to the actual address, 0101 with a field definition of 1.

After the special processing of DA and DC labels and symbolic switches, a check is made for the presence of

address arithmetic (switch 2 greater than 4). If there is address arithmetic, it is added to the symbol's equivalent address.

Following the processing of address arithmetic, if needed, a range test is made to ascertain if the actual address is within the limits of the object machine. If not, a message is typed, "ADDRESS OUT OF RANGE."

Following the range test, if switch 4 is equal to 3, indicating an ADCON, the ADCON is picked up and stored in SYMAD as a literal. Switch 4 is set to 2, the literal length is set to 4, and the processor then goes to test switch 3. Any value of switch 4 other than 3 causes the actual address to be stored in digits 6-9 of the assembled instruction.

Field definition is processed next. If not allowed (switch 2 equals 2), the switch is turned off and the processor goes to test switch 3. If allowed, the relative field definition must be calculated. The message "FIELD DEFINITION ERROR" occurs if either the first or second field definer exceeds 9.

If the compare-digit switch is on, only digit 5 of the assembled instruction is modified by placing in it the actual digit to be used for comparison. Switches 1 and 2 are both turned off in this instance. If the compare-digit switch is not on, both digits 4 and 5 of the assembled instruction are affected. They receive the calculated relative field definition. Switch 2 is then turned off.

Block 327, TESTSW3: Switch 3 is the final switch to be tested. If it has a value of either 0 or 2, the processor goes directly to write the record (block 329).

Block 328: A value of 1 in switch 3 indicates the presence of a symbolic index word in words 23 and 24 of the expanded record. This index word is picked up, randomized, and looked up on the symbol table. If the symbol is found, the assigned index word address is inserted into digits 2 and 3 of the assembled instruction, and switch 3 is turned off.

If the symbol is not found, and it is not the last pass, the record is merely written (block 329). If, however, it is the last pass, the symbol is looked up on the index-word and switch name table. If the symbol is found, the actual index word assigned is placed in digits 2 and 3 of the assembled instruction, and switch 3 is turned off. If the symbol is not found, an index word must be assigned. The assigned index word is placed in the assembled instruction, and switch 3 is turned off.

If switch 3 is equal to 3, the packed DC ADCON must be looked up on the symbol table. If the symbol is not found, the record is written (block 329) regardless of which Phase 3 pass is being processed. If the symbol is found, it is stored as a numeric constant, switch 3 is set to 2, and the record is then written on tape (block 329).

A value in switch 3 greater than 3 signifies a DA or DC label or a DRDW address. Once again, the symbol must be looked up on the symbol table. If the symbol is not found, the processor writes the record on tape (block 329). If it is found, the needed RDW's are generated, switch 3 is turned off, and the record is written on the object output tape (block 329).

Block 329, PUTOBJOUT: The expanded record, processed as much as possible by Phase 3, is given to the object-output file by a PUTX operation. Control then passes to block 320, where a new record is obtained.

Block 330, INPUTEOF: Entrance to this block is made on recognition of EOF on the object-input file by IOCS. The object-input file is closed, and the object-output file is released in preparation for possible error messages.

Block 331: A BES is made on LASTPASSW. If the switch is on, control passes to ENDROUTINE (block 333).

Block 332: If LASTPASSW is off, another pass of Phase 3 is necessary. Therefore, the files (except for object-output) are initialized in preparation for the next pass. The initialization consists of changing positions 0-3 of the first word of the DTF's channel and unit number.

Block 333, ENDROUTINE: Phase 3 is concluded by setting up communication word 2 (96) with the information necessary for Phase 4, and typing the end-of-phase message. Control then passes to system control to bring in Phase 4.

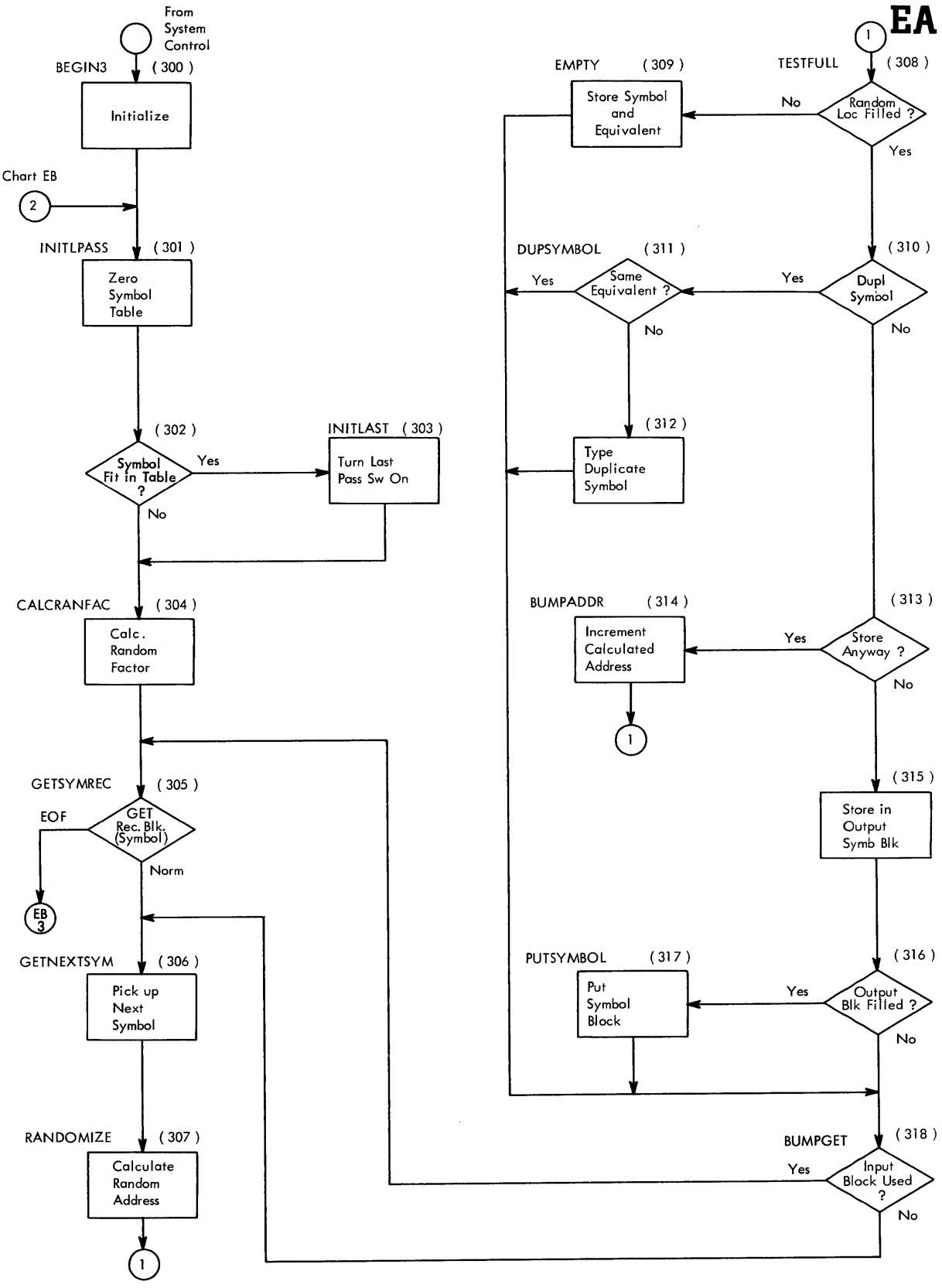


Chart EA. Phase 3

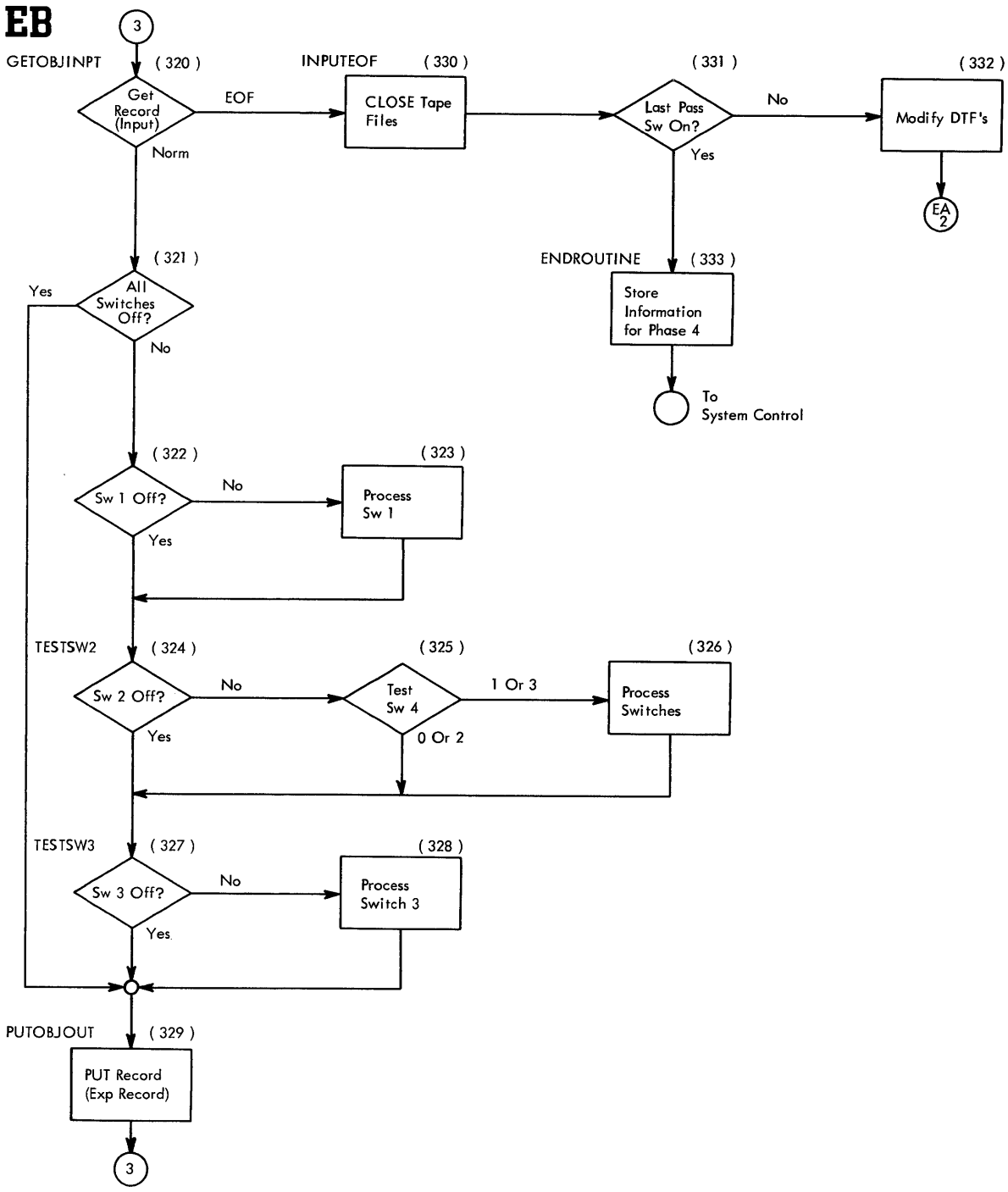


Chart EB. Phase 3, Continued

Phase 4 completes the assembly by:

1. Processing symbolic index words and electronic switches, using the index-word and switch table compiled in Phase 3.
2. Looking up, in the index-word and switch table, the symbolic operand of any incompletely processed ADCON, DRDW, or symbol defined in Phase 3.
3. Packing and assigning literals.
4. Creating the object-program listing tape, and, if the option is made, printing this listing on-line.
5. Preparing the condensed object-program tape complete with load and execute cards, and, if the option is made, punching these cards on-line.

The following description of the operation of Phase 4 refers to Charts FA, FB, and FC.

Block 400, INIT4: This block represents initialization of the DTF's (tape density, etc.), index words, switches, and counters. They are set to the condition necessary for proper operation at the beginning of Phase 4. The block also writes the load program on the condensed load tape, and, if the option is indicated, punches it on-line.

Block 401, BEGIN4: A 26-word input record is placed in the work area by a GET (tape record to area) operation.

Block 402: The label is tested for an asterisk. If it contains an asterisk and blanks (comments card), a branch is made to block 425, bypassing the switch-checking and processing section.,

Block 403: The operation code is compared to CNTRL. If they are equal, a branch is made to block 404.

Block 404: Since a control operation was recognized, any constants pending (incompletely processed) are processed.

Block 405: A branch (BZ1) is made to block 425 if all indicator switches located in word 17 (0-3) of the expanded record are off (zero). They are off if the entire instruction was processed.

Block 406: A compare test of switch 1 is made and, if it is off (zero), a branch is made to block 409 to test switch 2. If switch 1 is set to 3, a branch is made to block 407. If it is greater than 7, a branch is made to block 408.

Block 407: The CDSWITCH (electronic switch 1) is turned on, and the program branches to block 409 to test switch 2.

Block 408: The first address of the unprocessed DRDW is looked up in the index-word and switch table. If it is found, positions 2-5 of the assembled instruction are modified to their proper value.

Block 409: Switch 2 is tested. If it is off, a branch is made to block 414. If it is on, control is passed to block 410.

Block 410: Switch 4 is tested, and if it is set to 0 or 2 (literal), a branch is made to PRLIT (block 416) to process the literal. If it is 1 or 3, control is passed to block 411 to process the ADCON or other symbol.

Block 411: When switch 4 is set to 3, it signifies an ADCON and the ADCONSW (electronic switch 3) is turned on. If switch 4 is either 1 or 3, the symbol is looked up in the index-word and switch table and processed by modifying positions 6-9 of the assembled instruction.

Block 412: The ADCONSW (electronic switch 3) is tested and, if it is on, a branch is made to block 416.

Block 413: If required, a calculation of the relative field definition is made. If the instruction is compare digit, position 5 of the assembled instruction is modified. For other instructions, positions 4 and 5 are modified.

Block 414: Switch 3 is tested and, if it is off, the program branches to block 425; otherwise, control passes to block 415.

Block 415: If switch 3 is equal to 3 (DC ADCON), the symbol is looked up in the index-word and switch table. This equivalent or any other DC entry needing packing must be held if the word being packed is incomplete. Switches are set to control operation for examination of the next record.

Block 416: A look-up is made in the literal table to determine if the literal has already been used. If so, it will not be necessary to repeat it, and control passes to block 424.

Block 417: Switch 2 is compared to a 2 to determine if field definition is allowed. If it is equal to 2, the program branches to block 422, bypassing the packing operations.

Block 418: Literals are packed by means of pack-tables. These tables contain field definition of unfilled positions of a given assigned word and the address of that word. The pack-table is searched for an exact fit for the literal being processed. If one is not available, a test determines if a fit for this literal can be made. If no word indicated by the pack-table will hold the literal, control passes to block 420.

Block 419: The literal location and field definition are calculated and placed into the 26-word record. The

field definition of the pack-table word is then reduced accordingly. If an exact fit is available, a new record is added to the pack-table.

Block 420: Since the literal will not fit into any of the words indicated by the pack-table, the most complete word in the pack-table is assigned a new literal location.

Block 421: The literal under test is assigned to this new word in the pack-table. The remaining unused digits in this word determine the new field definition.

Block 422: The two counters (one for locating the next position in the literal table, and the second for designating the location assignment of the literals) are incremented by one.

Block 423: The literal and its equivalent are placed in the literal table.

Block 424: Digit positions 4-9 of the assembled instruction are modified to the calculated figures giving the literal's address to the instruction concerned.

Block 425, EDITPROC: This block represents the processing of a pending constant (incomplete DC word). If a bypass card (type 6) is recognized, and the pending constant has not filled a word, the bypass entry is written, leaving the constant still pending. This is necessary for DTF's where generated constant entries are alternated with the source entry (bypass card). Packing is necessary to complete the nine words. If the word has been filled, it is necessary to force output of this word for the condensed card output.

Block 426, EDIT: The edit area is set to alpha blanks. The line count is computed for the listing and, if new headers are necessary, they are prepared and written. All information (label, operation, etc.) is moved to the edit area.

Block 427: A test is made for TYPECODE6 (comments card, EQU, etc.). For TYPECODE6 records, a branch is made to block 429, bypassing the object-output operations.

Block 428: The object-output preparations are performed. The output is edited into card images in load card format. Each load card is PUT to the object-output tape. If the punch on-line option is used, the load cards are punched.

Block 429, WRITELIST: The record is PUT to the listing tape. If the printer on-line option is used, the record is printed.

Block 430: A comparison is made with the label (word 2 of the 26-word record). If it is END, a branch equal is made to block 435. If it is LITOR (litorigin card), a branch equal is made to block 431. Control passes to block 405 for any other label.

Block 431: The litorigin record is edited and PUT to the listing tape.

Block 432: The literals are packed using the literal-table index word (LITBLIW). The field definition and address of the literal (located in the literal table) are used to place its assigned literal in the table in its proper place. The RDW (2-5 portion) is incremented by 2. The literal is in the first word, right adjusted; the field definition and address are in the second word, positions 4-9. The process is repeated until a BCX instruction recognizes that the last literal has been placed.

Block 433: The literals are edited and listed on the listing tape, and they are also condensed and written on the object program tape. If the options are used, these records may be printed and punched.

Block 434: The literal routines are initialized for the next litorigin entry.

Block 435: The execute record for the load program is sent to the hold area.

Block 436: The end record source code is compared to 1 (generated instruction in Phase 1). If they are equal, indicating a generated instruction, a branch is made to block 438.

Block 437: The end record is listed because it was an original entry.

Block 438: The literals are packed using the literal-table index word (LITBLIW). The field definition and address of the literal, located in the literal table, are used to place the assigned literal in the table in its proper position. The 2-5 positions of the RDW LITBLIW are incremented by 2. The literal is in the first word, right adjusted; the field definition and address are in the second word, digits 4-9. The process is repeated until a BCX instruction on LITBLIW recognizes that the last literal has been processed.

Block 439: The literals are edited and listed, and they are also condensed and written on the object program tape. If the options are used, these records are also printed and punched.

Block 440: The final execute card is written on the object program condensed tape and punched, if that option is indicated.

Block 441: This block writes electronic-switch and index-word availability tables, followed by error messages, on the listing tape with appropriate labels.

Block 442: The object input, listing tape, and condensed-load tape files are closed.

Block 443: Messages indicating tape outputs and program completion are typed and a halt 4444 is made.

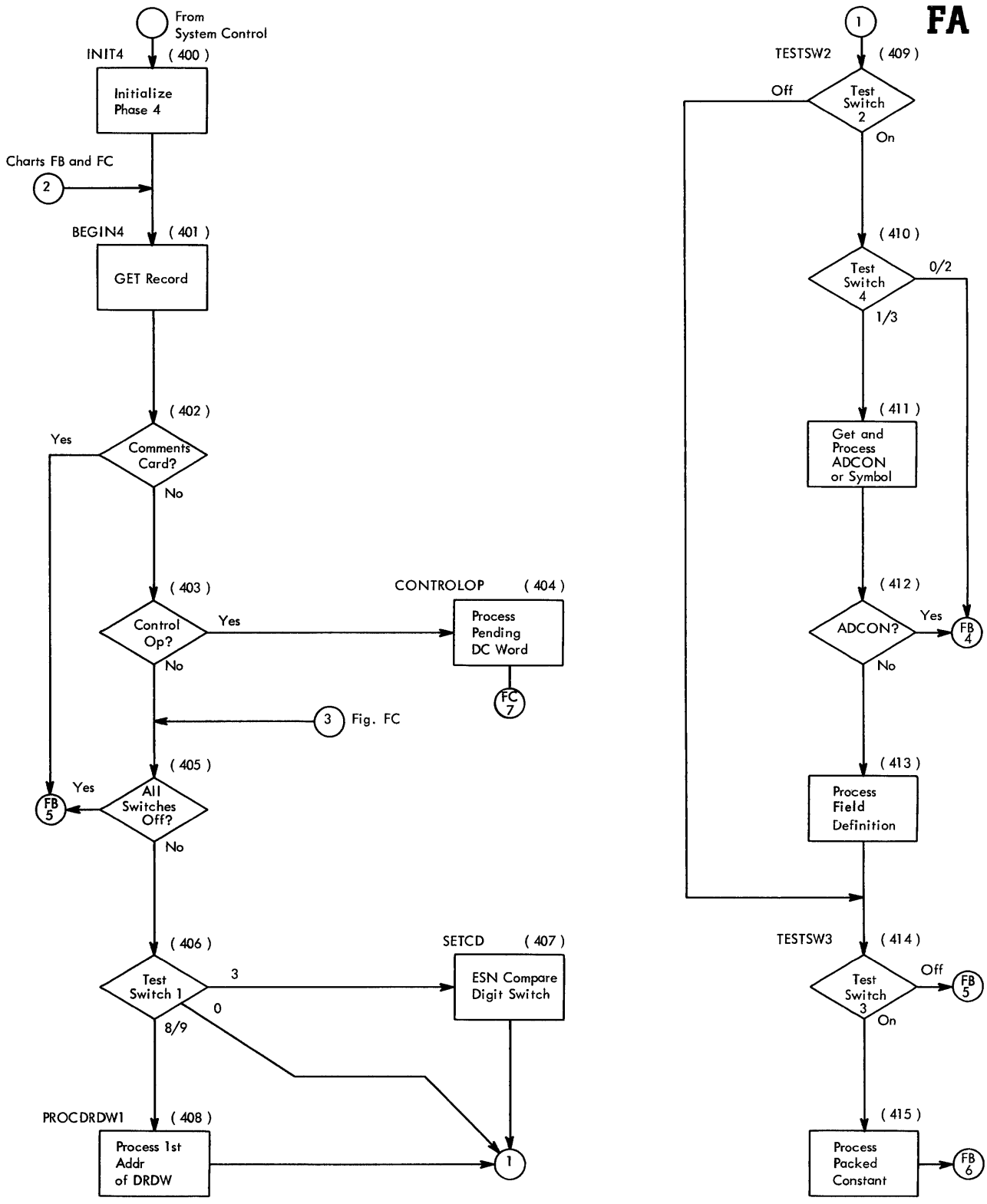


Chart FA. Phase 4

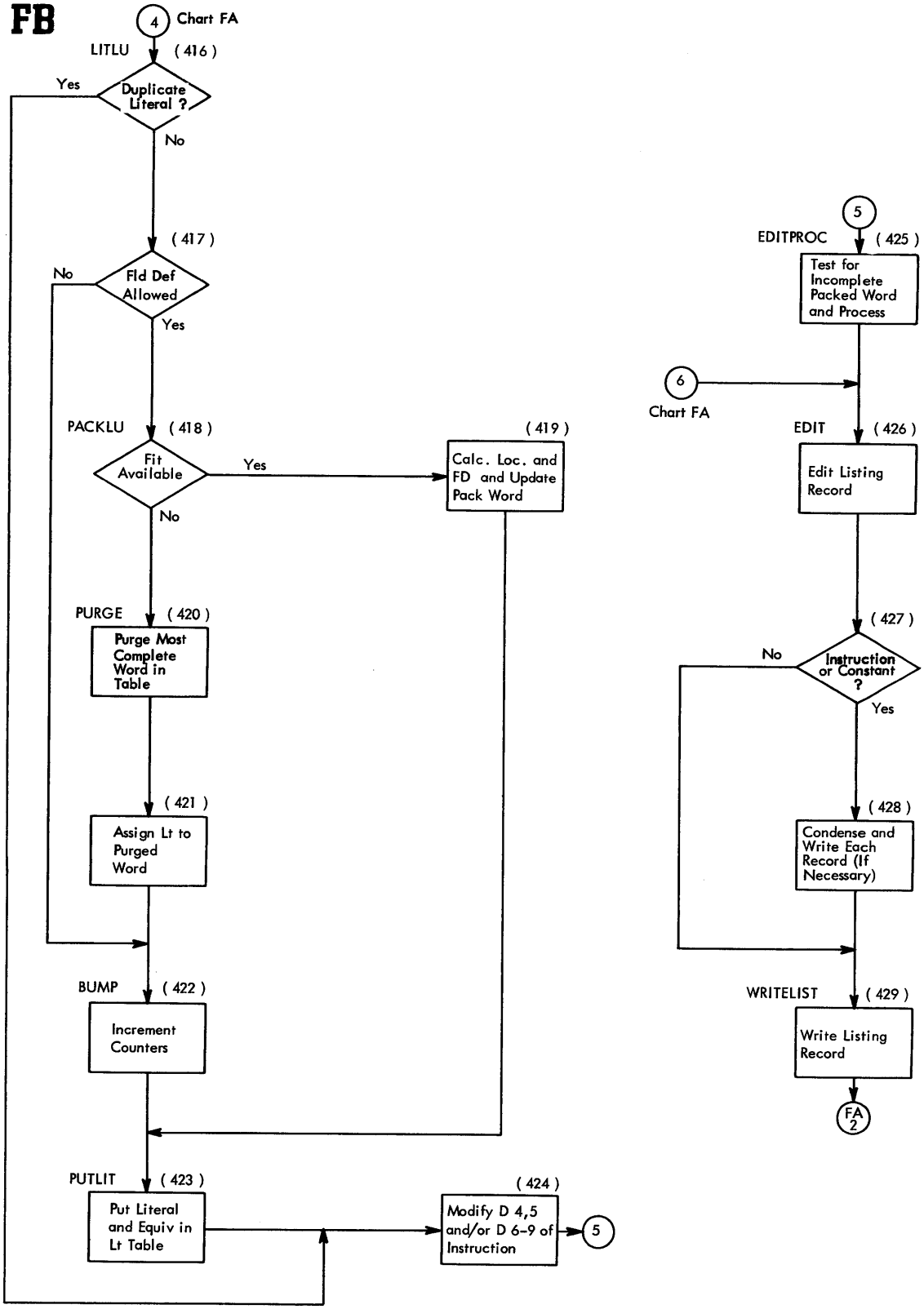


Chart FB. Phase 4, Continued

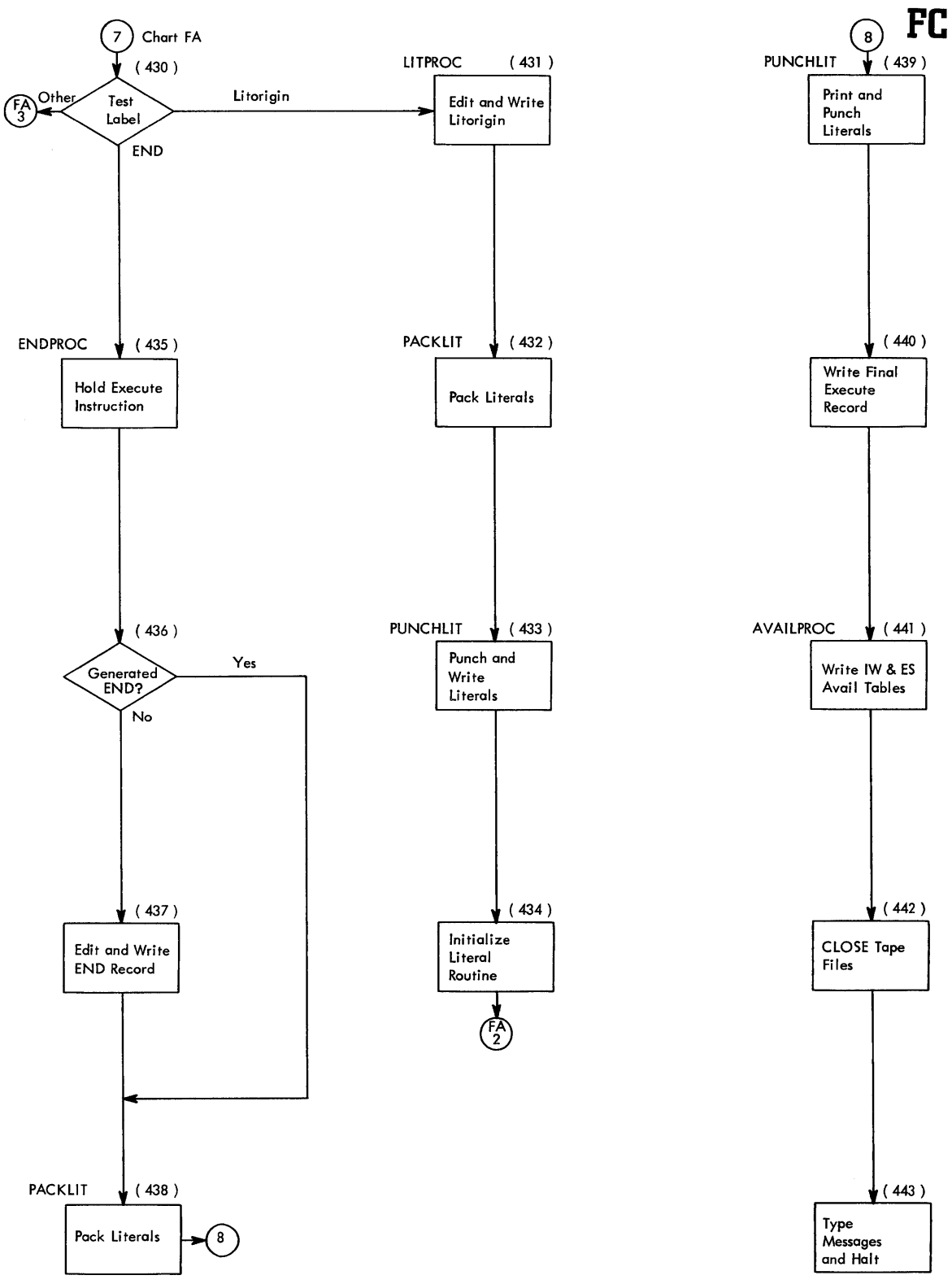


Chart FC. Phase 4, Continued

Expanded Record

Each part of every Autocoder statement (label, operation, and operand, including field definition, address adjustment, and indexing) must be processed. Phase 2 does as much of the processing as it can and passes to the succeeding phases the information needed to complete the job. The 26-word record developed by Phase 2 for each entry it processes (specifically, the settings of the four indicator switches) controls the entire operation.

The source-record entry, or the macro or subroutine skeleton in alphameric form, forms the first 16 words of the 26-word record. The 17th word contains control information for both processing and printing. The remaining nine words provide space for the assembled instruction, its actual location, and for those portions of the Autocoder statement other than the label which Phase 2 cannot fully process.

Refer to Figure 5 in reading the following more specific description of the 26-word record. (In actual use, output fields are not "blanked," so information from previous records will appear in unused fields.)

Words 1-16 contain either the original source record in alphameric form or, in the case of macros and subroutines, one of the sequence of instructions inserted by Phase 1. The contents of the first 16 words are unchanged throughout the entire processing.

Word 17 is a control word used for both processing and printing. It is appended by Phase 1 to each 16-word record, fully developed by Phase 2, and used by Phases 2, 3 and 4. Digit positions 0 through 3 contain the four indicator switches; 4 through 7, print control; digit position 8 contains the source code; 9 contains the type code.

Word 18 contains, in digit position 3, the relocation indicator and, in positions 6 through 9, the assigned machine location of the instruction or data. The other digit positions are not used.

Word 19 contains the assembled instruction or data.

Words 20 and 21 contain the symbolic name used in the source program to refer to the address portion of an instruction (digit positions 6 through 9 of the assembled instruction, including a DRDW).

Word 22 contains address arithmetic in digit positions 6 through 9. The remainder of the word is not used.

Words 23 and 24 contain the symbolic name used in the source program to refer to an index word used to modify an address. The assigned index word will appear in digit positions 2 and 3 of the assembled instruction. Words 23 and 24 may also contain, for a DRDW, the symbolic address which refers to digit positions 2 through 5 of the assembled instruction.

Words 25 and 26 may contain the symbolic name for any component, such as index words, electronic switches and tape units, used in the operand. These index words will be assigned to digit positions 4 and 5 of the assembled instruction. For some statements digit positions 8 and 9 of word 25 will contain the field definition, which is always relative to how the symbol is defined. The remainder of words 25 and 26 will not be used.

For other statements, digit positions 6 through 9 of word 25 will contain address arithmetic for the first

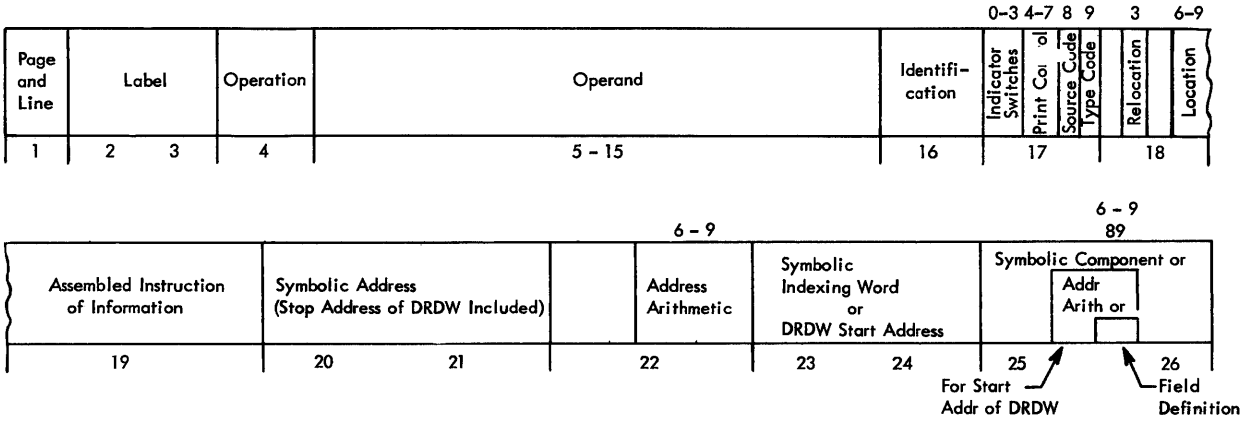


Figure 5. Composition of Expanded (26-Word) Record

address of a DRDW. The remaining digit positions of word 25 and word 26 will not be used.

In the composition of the 26-word record, note that:

1. Space is not allotted for the label, since it is written on the symbol tape.
2. The operation code is not stored, since all operation codes are processed by Phase 2.
3. Instructions which contain a component may not have field definition, so they have been assigned the same words.
4. The fields that are not used are not zeroed out and will, therefore, contain previous information. The four switches in positions 0-3 of word 17 determine the fields used by this statement for processing.

Indicator Switches

Phase 2 processes all operation codes and all actual operand elements. If an Autocoder statement has all its parts (other than the operation code) written in actual, its machine language representation will be completely assembled by Phase 2. No processing will be required by the subsequent phases. More common, however, will be the case in which Phase 2 cannot fully process an Autocoder statement because it contains symbols or literals.

Phases 3 and 4 must be able to determine what remains to be assembled so that they will know what further processing, if any, is required of them. All the necessary information is contained in four indicator switches, set by Phase 2 and tested and updated by Phases 3 and 4.

Figure 6 shows the functions of each of the four indicator switches. It defines the switch code numbers for each of the switches, the meaning of the code numbers, the portion of the element record associated with the code number, and the digit positions of the assembled machine instruction affected.

Digit positions 0, 1, 2 and 3 of word 17 of each expanded record produced by Phase 2 are used as indicator switches 1, 2, 3 and 4, respectively. The numbers placed in these positions are code numbers which convey to Phases 3 and 4 information about the remaining unprocessed words (18-26) of the expanded record.

If, for any record, Phase 3 or 4 encounters a zero in switch 1, 2, or 3, then those expanded record words associated with the switches containing a zero are either not used or completely processed for that record. All the switches are initially set to zero by Phase 1. Any one of the first three switches will remain at zero if the elements associated with that switch are fully

Switch	Sw Code No.	Meaning of Code Number	Expanded Record Word(s) Associated with Code Number	Digit Positions of Assembled Instruction (Word 19) to be Affected
1	0	No Symbolic Component	None	None
	1	Electronic Switch	25 and 26	1 and 5
	2	Index Word	25 and 26	4 and 5
	3	Compare Digit	None	5
	4	Misc: Alteration Switch, Channel, Synchronizer	25 and 26	4
	5	Tape	25 and 26	1 and 4
	6	RAMAC Channel	25 and 26	1
	7	Latch (Except Unit Record)	25 and 26	4 and 5
	8	DRDW Start Address	23 and 24	2 to 5
	9	DRDW Start Address with Address Arithmetic	23 and 24; 25(6-9)	2 to 5
2	0	No Symbolic Address	None	None
	1	Address Only (Field Definition Allowed)	20 and 21; 25(8,9)	4 to 9
	2	Address Only (Field Definition not Allowed)	20 and 21	6 to 9
	3	Address and Field Defintion	20 and 21; 25(8,9)	4 to 9
	5	Address with Address Arithmetic (Field Definition Allowed)	20 and 21;22(6-9);25(8,9)	4 to 9
	6	Address with Address Arithmetic (Field Definition not Allowed)	20 and 21;22(6-9)	6 to 9
	7	Address with Address Arithmetic with Field Definition	20 and 21;22(6-9);25(8,9)	4 to 9
3	0	No Indexing	None	None
	1	Indexing	23 and 24	2 and 3
	2	Packed DC Number	20	4 and 5
	3	Packed DC ADCON	20 and 21	4 and 5
	4	DRDW with a Single Address (Symbolic)	20 and 21	2 to 5
	5	DA or DC Entry	20 and 21	2 to 9
4	Used Only if Switch 2 is Not Equal to Zero			
	0	Literal (not Relocatable)	20 and 21	6 to 9
	1	Symbol	20 and 21	6 to 9
	2	Relocatable "Literal"	20 and 21	6 to 9
	3	ADCON	20 and 21	6 to 9

Figure 6. Indicator Switches of Expanded Record

processed by Phase 2 or if these elements do not appear in the original Autocoder statement. (See Figure 7, examples 1 and 2.)

Phase 2 sets the switches to indicate the type of processing that will be necessary in Phases 3 and 4. As the required processing is completed, the switches are reset to zero (off). The ultimate setting of the switches is zero, indicating a completely processed statement in word 19 and in word 18 its corresponding actual location, where applicable.

Switch 1 usually refers to the contents of words 25 and 26 of the expanded record. With the exception of code numbers 3, 8 and 9, a value other than zero in switch 1 indicates that words 25 and 26 contain alphameric information which is the symbolic name of a component (Figure 7, examples 2 and 3). Setting of switch 1 at 8 or 9 indicates the presence of alphameric information in words 23 and 24 of the 26-word record. In this instance, the information is the symbolic name of a core location and is processed as the starting address of a DRDW. A setting of 9 also refers to digit positions 6 through 9 of word 25, where an address arithmetic constant has been stored by Phase 2 (Figure 7, example 4). Switch 1 is set at 3 for a compare digit instruction (Figure 7, example 5), which requires special processing in its assembly; none of the words 20-26 is associated with this setting of switch 1.

Switch 2, when on (other than zero), always refers to the contents of words 20 and 21. In addition, it usually specifies other words to be used in completing the assembly of an instruction. Where field definition is allowed, digit positions 8 and 9 of word 25 are used; for address arithmetic, digit positions 6 through 9 of word 22 are used.

Switch 2, together with switch 4, characterizes for Phases 3 and 4 the address of the original Autocoder statement. In the case of the compare-digit instruction (Figure 7, example 5), switch 2 is set to 3, indicating that the address in words 20 and 21 is to be field defined by the numbers in positions 8 and 9 of word 25. Switch 4 indicates whether the information in words 20 and 21 is numeric or alphameric, and what kind of processing is required. Consider Figure 7, example 6; switch 2 is set to 1, since field definition is allowed by the processor and there is no address arithmetic. Switch 4 is set at 1 because the address is symbolic.

Switch 3, when it is other than 0 or 1, also refers to the contents of words 20 and 21 (Figure 7, example 7). A code number of 1 designates that words 23 and 24 will contain the symbolic name of an index word used to modify the address of an instruction (example 8).

The remaining examples, 9-12 in Figure 7, further illustrate the use of the indicator switches in conjunction with words 20-26 of the expanded record.

Words 1-16			Word 17 (0,3) Switches 1 2 3 4	Expanded Record Words 20	21	22	23	24	25	26
1	SI	9991	0 0 0 0							
2	ESN	SWA	1 0 0 0						SWA	
3	XU	ZEROS,3650	2 0 0 0						ZEROS	
4	DRDW	-A+2, B+6	9 6 0 1	B		0006+	A		0002+	
5	CD	DAYOFWEEK(3),5	3 3 0 1	DAYOFWEEK					33+	
6	ZAI	PAYMENT	0 1 0 1	PAYMENT					09	
7	DRDW	-INPUT	0 0 4 0	INPUT						
8	B	SUM+IWEXIT	0 1 1 1	SUM			IWEXI	T		
9	ZAI	+1000	0 1 0 0	1000+	40010+	Length of literal				
10	ZAI	+ADCONSYMB	0 1 0 3	ADCON	SYMB					
11	XL	C,A-10+IWB	2 6 1 1	A		0010	IWB		C	
12	ZAI	PAYMENTS(5,9)+3	0 7 0 1	PAYMENT		0003			59	

Any field not used is not zeroed out. Therefore, it will contain information left from the last autocoder statement which made use of the field. The information is no longer pertinent and consequently classified as not used.

Figure 7. Examples of Autocoder Statements

Print Control Mask

The print control mask occupies digits 4-7 of word 17 in the control word and refers to the assembled instruction. Each position may contain a 1 or a 0. A 1 indicates the suppression of printing:

DIGIT POSITION	
4	Digit positions 2 and 3 of instruction
5	Digit positions 0 and 1 and sign of instruction
6	Entire instruction including sign
7	Location of instruction

Source Code

The source code, digit 8 of word 17 in the control word, is used by Phase 4 for controlling the printing of an entire line in the listing. It may have any of three values: 0, 1 or 2. A 0 indicates that the entire line is to be printed. Both 1 and 2 refer to Autocoder-generated lines (information produced by Autocoder other than a one-for-one interpretation of the source program).

A value of 1 indicates Phase 1 generation; a value of 2, Phase 2 generation. In either case the line of listing will contain an X in the page and line column. A source code of 2 further suppresses the printing so that the line of listing contains besides the X only the assembled instruction and its location.

SOURCE CODE	MEANING
0	Print entire line
1	Blank page and line, insert X
2	Blank everything to left of location, insert X

Type Code

The digit in position 9 of word 17, the control word, designates the type of card (or record) and its form (numeric or alphameric). The information is used by Phase 4 in packing the condensed cards.

TYPE CODE	MEANING
0	Execute card containing an unconditional branch to a location specified in a branch or end-control entry.
1	Card contains a ten-digit numeric word with sign which is included in the object program deck.
6	Card contains a source program entry which does not produce instructions in the object program deck. (Examples are: comments cards, EQU card and a DA subsequent entry.)
7	Card contains the double-digit representation of a five-character alphameric word which is included in the object program deck.
8	Actual IOCS equate
9	Symbolic IOCS equate

System Control and Librarian

Storage Map

AREA	SYMBOLIC ADDRESS	ACTUAL ADDRESS
System Control	PHASELOAD	0130-0149
System Control	POSITION	0180-0287
Librarian	TEST	0325-0423
Macro and Subroutine Position Tables	OPL	0493-1292
Permanent Options	MORPERMOPS	1293-1295
General Scheduler	IOCSMASK	1296-1301
Channel 1 Scheduler	IOCSC1S	1302-1336
Channel 2 Scheduler	IOCSC2S	1337-1371
EOR Routine	IOCSEOR	1372-1772
OPEN2 Routine	IOCSIOPEN	1773-2015
Condition Code Routine	IOCSIRTAIN	2016-2164
Error Routine	IOCSERROR	2165-2412
END and CLOSE Routines	IOCSIEND	2413-2517
IOCS Index		
DTF's	TAPEFILE20	2518-2545
File Schedulers	IOCSSCED01	2546-2666
Areas and RDW's	INFILE	2667-3100
Initialization Program for Librarian	STARTSF	3101-3902
IOCS Subroutine LMSR1	IOCSLMSR	3903-3985
Literals		3986-4030

NOTE: Some actual locations may change with reassembly

Communication Words, System Control

INDEX WORD 88

This word is unchanged for all phases; it is set by system control. It contains the version and level number of the system tape.

INDEX WORD 89

S	0 - 1	2	3	4 - 9
+	Not Used	Tape Density	No Load Program	Not Used

Position 2: Zero indicates the source program tape. The listing tape and the program tape are to be in low density (200 characters per inch). Two indicates that all three tapes are to be in high density (556 characters per inch).

Position 3: Non-zero indicates that the load program is to be deleted on the object-program output.

INDEX WORD 95

Contents of this word are as specified by the permanent options, or by the temporary option change cards. The word remains unchanged throughout the four phases with the exception of the sign which is made plus in Phase 2, and is unchanged thereafter.

S	0	1	2 - 5	6 - 9
+ -	Obj Deck	Report List	Equip Size	Object Size

The sign, if minus, indicates that header labels are not to be expected by system control and Phase 1 on the work tapes. The sign, if plus, indicates header labels are to be expected on the work tapes.

Position 0-1: These positions are set by system control with the synchronizer number. They are used by Phase 4.

Positions 2-5: These positions are set by system control and used by Phases 1, 2, and 3.

Positions 6-9: These positions are set by system control and used by Phases 2, 3, and 4.

INDEX WORD 96

The contents of this index word vary for the different phases, and are used for between-phase communication.

S	0	1	2 - 5	6	7	8 - 9
+	Source Deck	Option Unit	Size of Macro Table	Labels Out	Labels In	Messages

Positions 0-1 contain the input units for the source deck and options. The positions contain digits 1 through 3, indicating the reader synchronizer number, or zero indicating input on tape 12.

Positions 6-7 contain digit zero, indicating "no labels," or 1, indicating that labels are expected. This information is transferred to the sign of index-word 95.

Positions 8-9 contains digits to specify the output unit for the messages produced by Phases 1-4. Zero specifies the messages produced on listing tape only; 04, the messages typed in addition to being printed on the listing; S3, the messages printed on the printer designated by S, and in addition written on the listing.

PHASE CONTROL WORD

This word, location 893, is used only by system control during all runs.

S	0	1	2 - 5	6 - 9
+	Phase No	Not Used	Loc of Last Instruction of Phase	Loc of Start Instruction of Phase

Position 0 is used to check if the proper phase is loaded.

Positions 2-5 are used only in a system run. They specify the size of the block to be written on tape.

Positions 6-9 specify the branch location to start the phase that is being loaded.

Index Words Table

Figure 8 shows all index words for the system control and librarian.

NO.	LABEL	LABEL NAME	FUNCTION
3	SORTX	Count for Sorting Macro Table	
4	WRITERDW	RDW for Writing Phases on System Run	
5	PHASE	Position of Phases on System Tape	
6	X310R	Save 310 of Load Program	
7	X311R	Save 311 of Load Program	
8	IOCSIXF	IOCS Index Word	
9	IOCSIXG	IOCS Index Word	
10	INDXWRDA20	System Tape File, Index Word A	Also SYSTEMSXA, IOCSFIXA01
11	INDXWRDB20	System Tape File, Index Word B	Also SYSTEMSXB, IOCSFIXB01
12	INDXWRDA12	Input Tape File, Index Word A	Also INFILEXA, IOCSFIXA02
13	INDXWRDB12	Input Tape File, Index Word B	Also INFILEXB, IOCSFIXB02
14	INDXWRDA13	Output Tape File, Index Word A	Also OUTFILEXA, IOCSFIXA03
15	INDXWRDB13	Output Tape File, Index Word B	Also OUTFILEXB, IOCSFIXB03
16	MOVE	RDW for RS to Move Macro Table	
17	INSERCOUNT	Count of Macros in Library	
18	CONVEX	Convert Exit	Linkage to "Alpha to Numeric Conversion" Routine
19	SKELCOUNT	Skeleton Record Counter	Count of Detail Library Change Cards
90	RPTCOUNT	Repeat Count Constant	
91	SKIPCOUNT	Skip Count Constant	
92	EXIT2	Return from BLX	Linkage index word for BLX
93	EXIT	Return from BLX	Linkage index word for BLX
95	TEMPOPT	Communication Word 1	Holds temporary options
96	TEMPOPT2	Communication Word 2	Holds temporary options

Figure 8. Index Words, System Control and Librarian

Phase 1

Storage Map

AREA	SYMBOLIC ADDRESS	ACTUAL ADDRESS
System Control	PHASELOAD	0130-0149
System Control	POSITION	0180-0324
Macro Name Table	MACTBLAREA	0325-0524
DTF Table	DTFTBLAREA	0855-1021
General Scheduler	IOCSMASK	1022-1027
Channel 1 Scheduler	IOCSC1S	1028-1062
Channel 2 Scheduler	IOCSC2S	1063-1097
EOR Routine	IOCSEOR	1098-1498
Condition Code Routine	IOCSIRTAIN	1499-1647
Error Routine	IOCSERROR	1648-1895
END and CLOSE Routines	IOCSIEND	1896-2000
IOCS Index		
DTF's	TAPEFILE01	2001-2037
File Schedulers	IOCSSCED01	2038-2225
Areas and RDW's	CARDIP	2226-2579
IOCS Name Table and RDW	IOTBL	2580-2618
Special Name Table and RDW	SPECTB	2619-2637
DTF Control Table and RDW	TFCTL	2638-2674
Constant Area	FLSEND	2675-2723
Phase 1 Programming	INITP1	2724-4569
OPEN Routine (overlaid)	IOCSIOPEN	4650-4892
Symbol Request Table and RDW	REQTBL	4580-Equip. Size

NOTE: Some actual locations may change with reassembly

Communication Words, Phase 1

INDEX WORD 89

S	0 - 1	2	3 - 9
+	Messages	Tape Density	Not Used

Positions 0-1 are the same as positions 8-9 of index word 96.

Position 2 is unchanged from system control.

INDEX WORD 95

Contents of this word are as specified by the permanent options as applicable, or by temporary option change cards. The word remains unchanged throughout the four phases.

S	0	1	2 - 5	6 - 9
+ -	Obj Deck	Report List	Equip Size	Object Size

The sign is a work-tape label indicator. Plus indicates that labels are not expected by Phase 1.

INDEX WORD 96

S	0	1	2 - 5	6 - 7	8 - 9
+	Source Deck	Run Type	Size of Macro Table	Not Used	Pos 4-5 of Write Msg Opt

Position 0 contains the source deck option unit. Digits 1-3 indicate the reader synchronizer number; digit 0 specifies tape 12.

Position 1 contains coding for the type of run. Digit 0 indicates an original run; digit 1 specifies a reassembly run with the listing on tape unit 13.

Positions 2-5 contain the size of the macro-name table.

Positions 8-9 contain positions 4 and 5 of the message-writing instruction. Zero indicates that messages are to be written on the listing tape only. If, in addition, they are to be typed or printed, positions 8-9 contain digits that indicate the synchronizer number and operation-code digit. This information is transferred to digit positions 1-2, index word 89.

Table Formats, Phase 1

MACRO NAME TABLE (MACTBLAREA)

The RDW for MACTBLAREA (words 0325-0524) is MACTBL. The table increment is one word for each macro name. This table contains the names of all macros and subroutines contained in the library.

0326	0372
999AB	ZSUMb

DTF TABLE (DTFTBLAREA)

This table (words 0855-1021) contains information necessary to locate correct file schedulers and other macros for each file operation. The RDW of DTFTBLAREA is DTFTBL.

1	2	3	4	5	6	7	8	9
0855	0856	0857	0858	0859	0860	0861	0862	0863

10	11	12	13	14
0864	0865	0866	0867	0868

Words 1 and 2 contain the file name (alpha).

Word 3 (signed +) contains character codes:

- 0 BASETAPE
- 1 ALT1TAPE
- 2 FILETYPE
- 3 FILEFORM
- 5 TIOAREAS
- 6-9 DTF number in double-digit form

Word 4, positions 0-3, contains the blocking factor for form 4 records.

Words 5 and 6 (alpha) contain the record length indicator (double-digit) for form 3 records.

Words 7 and 8 contain the file's index word, IOCSFIXA*nn*.

Words 9 and 10 contain the file's index word, IOCSFIXB*nn*.

Words 5 through 10 are used to form the four possible EQU's after the DTF. When these EQU's are completed, these words will be overlaid with information for the next DTF; e.g., DTF02 is located in words 5-14.

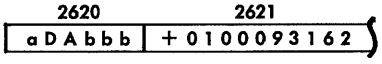
DUF TABLE (DUFTBL + 1)

This table (words 2526-2543) contains the DUF name and number, in double-digit form, of all DUF's used in the object program. The table increment is three words per DUF. The RDW of the table is DUFTBL.

Words 2526 and 2527 contain the name of the DUF in alphameric form. Word 2528 contains the DUF number in positions 0-1 (alphameric).

SPECIAL OPERATION CODE TABLE (SPECTB)

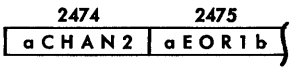
This table (words 2620-2637) contains the names of operation codes which may contain symbols that must be previously defined, and codes needing special processing. The table increment is two words per operation code.



Word 2620 is in double-digit form, signed plus. Word 2621 is a branch-pivot to allow entry to the special routine; e.g., the DA routine is located at 3162.

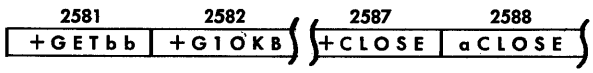
SUBROUTINE NAME TABLE (SUBNAM)

This table (words 2474-2523) holds the subroutine names from an INCL (include) operation or a DIOCS entry. The table increment is one word per name. Each word used in the table contains the name of a subroutine that is to be included in the object program.



IOCS MACRO-NAME TABLE (IOTBL)

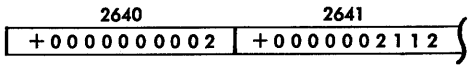
This table (words 2581-2618) contains the names of the IOCS macros and, if the sign of the second word is plus, the processor's basic name of the macro. These basic names are used to locate a specific macro, if the macro needed for a given operation is dependent on variables; e.g., FILEFORM, FILETYPE, etc.



Word 2581 is the macro name in double-digit form, signed plus. Word 2582 is the basic name of the macro in double-digit form, signed plus. Word 2587 is the macro name in double-digit form, signed plus. Word 2588 is the macro name in alphameric (no basic name needed).

DTF CONTROL TABLE (TFCTL)

This table (words 2639-2674) contains control information for processing the DTF entries. Each table entry consists of one word of control coding. The entries in the table are in the specified sequence for the DTF source entries. This table allows processing of the parameters and coding without field definition, distinctive labels, etc., in each source entry.



Word 2640 is the control word for FCHANNEL. Word 2641 is the control word for BASETAPE. The control code location and usage determine the processing of the DTF entries.

Position 0 is unused.

Position 1: 0 is normal, 1 is RLIFORM3, 2 is SCHEDINF.

Positions 2 and 3: Range if position 9 is 1 (e.g., FILEFORM is 04).

Positions 4 and 5: Field definition in the DTF table (DTFTBL) if position 6 is not equal to 0.

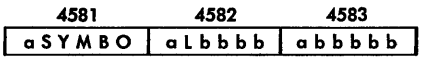
Position 6: 0 indicates the entry is not needed to store in DTFTBL; any digit other than 0 indicates one less than its word position in the DTFTBL (e.g., SRBFORM4 is 3, located in word 4).

Positions 7 and 8: Field definition of entry in the object program.

Position 9: Operand type; 0 is blank; 1 is actual; 2 is actual or blank; 3 is actual, symbolic, or blank; 5 is ALT1TAPE being processed; 6 is ALT2TAPE being processed.

SYMBOLIC REQUEST TABLE (REQTBL)

The symbol request table (word 4581 to equipment size) is built by Phase 1 and remains in storage for use by Phase 2. It contains all symbols which must be previously defined; i.e., they must have appeared as labels earlier in the program sequence. The symbols so classified are: those contained in the operand of any control operation and of the declarative operation EQU, and those appearing as the single address of a DRDW or the relative address of a DA. The table increment is three words.



Word 4581 is the symbolic address from the operand, first word. Word 4582 is the symbolic address from the operand, second word. Word 4583 contains alphameric blanks (equivalent in Phase 2).

Index Word and Electronic Switches

Figure 9 shows the index words for Phase 1, and Figure 10 summarizes the electronic switches for Phase 1.

NO.	LABEL	LABEL NAME	FUNCTION (CONTENTS)
3	IOCSIXF	DIOCS IXF	IOCS index word
4	IOCSIXG	DIOCS IXG	IOCS index word
5	IOCSFIXA01	DTF IXA	System tape index word A
6	IOCSFIXB01	DTF IXB	System tape index word B
7	IOCSFIXA02	DTF IXA	Output tape, 17-word record, index word A
8	IOCSFIXB02	DTF IXB	Output tape, 17-word record, index word B
9	IOCSFIXA03	DTF IXA (CARDX)	Input tape (card image), source prog, index word A
10	IOCSFIXB03	DTF IXB	Input tape (card image), source prog, index word B
11	IOCSFIXA04	DTF IXA (TAPEX)	Input reassembly, 23-word record, index word A
12	IOCSFIXB04	DTF IXB	Input reassembly, 23-word record, index word B
13	SEGCTR	Segment Character Control	Location of segment character for next litorigin
14	ZEROS	Zero Held	Contains zeros to zero words for initialization
15	DTFCTR	DTF Counter	No. of DTF's in source program
16	DUFCTR	DUF Counter	No. of DUF's in source program
17	BLANKX	Blank Hold	Blanks for initialization
18	COMMAX	Comma Hold	,bbbb for initialization
19	DUFX	DUF Table RDW	RDW for table of pertinent source program DUF information
20	DTFX1	DTF Table RDW	RDW for table of pertinent source program DTF information
21	CTLX1	Control Word 1 Hold	Miscellaneous control words - unloaded into word 17 of output record to control subsequent processing & printing
22	CTLX2	Control Word 2 Hold	
23	CTLX3	Control Word 3 Hold	
24	CTLX4	Control Word 4 Hold	
25	REQX	Request Table RDW	RDW of symbol request table
26	SNTX1	Subroutine Name Table IX 1	RDW of subroutine name table
27	SNTX2	Subroutine Name Table IX 2	No. of names in subroutine name table
28	EXIT	Exit	Exit address from some of the subroutines
29	MOVEX	Move Index 1	Location of numeric information in edit operations
30	SAVEACC1	Save Accumulator 1	Preserve contents of accumulators before entering error routine
31	SAVEACC2	Save Accumulator 1	
32	SAVEACC3	Save Accumulator 1	
33	ERROR	Error Exit	Exit address from error routine
34	ERRMSGRDW	Error Message RDW	RDW of error message work area
35	MSGGATHER	Message Gather	To address for moving messages
36	MOVE	Move Index 2	From or to address in move operations
37	TEMPERRX	Temporary Index Word	Temporary storage for out-of-sequence routine
38	REXIT	Read Exit	Exit address from some of the subroutines
39	TEMPX	Temporary Index Word	Miscellaneous
40	ENDX	End Card Index	Control word hold for end card
41	IW	Indexed Word	Word location (Used in Scan 1 (SCANR))
42	IC	Indexed Character	Character within word
43	SCANRX	SCANR Exit	Exit address from SCAN
44	ACX	Another Character Exit	Exit address from next character pickup
45	SNTX3	Subroutine Name Table IX 3	Location of name of next subroutine to be generated
46	WI	Word Indexed	Same as IW, ACX, & IC --Reverse of scan (work area build-up)
47	CAX	Character Another Exit	
48	CI	Character Indexed	
49	DTFX3	DTF Index 3	Miscellaneous
50	TFCTLX	Tape File Control Index	Location of control word for DTF processing
51	DTFX2	DTF Index 2	Location of next available word in DTF table
52	CEXIT	Characters Store Exit	Exit address from construction routines
53	TEMPNAME	Temporary Name Hold	Temporary hold for macro name
54	AAX	Address Arithmetic Exit	Exit address from macro AA routine
55	FDX	Field Definition Exit	Exit address from macro FD routine
56	IXX	Indexing Exit	Exit address from macro IX routine
57	OPDCTR	Operand Counter	Number (by type) of IOCS macro parameters
58	AM	Same as IC, IW, ACX and SCANRX, respectively, for Scan 2 (SCANM)	
59	IM		
60	MAX		
61	SCANMX		
62	SEXIT	Substitution Exit	Exit address from substitution routine
63	IXHOLD 1	Index Word Hold 1	Temporary hold for index word from macro or skeleton instr
64	IXHOLD 2	Index Word Hold 2	
65	MAHOLD	Macro Character Hold	Hold for next character from skeleton instr
88	LEVELX	System Level No.	Control information from Sys Ctrl to phase 4 (Level no. for listing)
89	MSGOPT	Message Option	Control information from phase 1 to phase 2 (Error msg option)
93		Systems Control Exit	Exit address from system control tape spacing rtn
95	COMUNC	Communications Word 1	Control information from phase 1 to phase 4
96		Communications Word 2	Control information from phase 1 to phase 2 (Request tbl RDW)

Figure 9. Index Words, Phase 1

NO.	LABEL	LABEL NAME	FUNCTION (WHEN ON)
1	FSTSUB	First Subroutine Switch	No names in subroutine name table
2	CONLY	Card Only Switch	Card image input only (Tape 12 or Card Rdr)
3	MACROS	Macro Switch	Read input from systems tape library
4	SKEDSW	Scheduler Switch	Generating file schedulers
5	INSSUB	Insert Subroutine Switch	Generating subroutines
6	DIOCSS	DIOCS Switch	Processing DIOCS card
7	MOVESW	Move Switch	Next input card is in temporary hold
8	EOJSW	End-of-Job Switch	When processing last card
9	TONLY	Tape Only Switch	Read previous listing only (tape 13)
10	DTQSKEDSW	DTQ Scheduler Switch	Generating DTQ file schedulers
11	ENDSW	End Card Switch	End card of EOF inputs have been encountered
12	REQFUL	Request Table Full Switch	Request table has filled available storage
13	IOCSS	IOCS Switch	Processing IOCS macro
14	LITSW	Litorigin Switch	Processing litorigin cntrl statement
15	DTQSW	DTQ Switch	Processing source program which uses DTQ & DIOQ
16	AA1SW	Address Arith 1 Switch	AA from macro in temp. hold
17	AA2SW	Address Arith 2 Switch	AA from skeleton in temp. hold
18	FD1SW	Field Def 1 Switch	FD from macro in temp. hold
19	FD2SW	Field Def 2 Switch	FD from skeleton in temp. hold
20	IXISW	Indexing Switch	Index word from macro or skeleton in temp. hold
21	ALFSW	Alphabetic Const. Switch	First @ character encountered in scan
22	ALF2SW	Alphabetic Const. 2 Switch	Ending @ character encountered in scan
23	LOZER1	Lozenge Zero 1 Switch	Previous skeleton instr. had □ 0 label and was deleted
24	ADDR	Address Switch	Address already placed in generated insts
25	MACMOD	Macro Name Modify Switch	IOCS macro name is to be modified according to type
26	LOZER2	Lozenge Zero 2 Switch	Skeleton instr. being processed has □0 label
27	MISSING	Missing Switch	Parameter was illegally missing
28	IGNORE	Ignore Switch	First skeleton condition was satisfied, ignore second
29	SIGN	Sign Switch	Minus sign was associated with addr. arithmetic

Figure 10. Electronic Switches, Phase 1

Phase 2

Storage Map

AREA	SYMBOLIC ADDRESS	ACTUAL ADDRESS
Index Word Availability Table	IWTABL	0325-0334
Electronic Switch Availability Table	ESTABL	0335-0337
Litorigin Table	LITORT	0338-0373
Symbol Table (written on symbol tape)	TABLEXZ	0374-0503
Areas and RDW's	INPUTZ	0504-0977
General Scheduler	IOCSMASK	0978-0983
Channel 1 Scheduler	IOCSC1S	0984-1018
Channel 2 Scheduler	IOCSC2S	1019-1053
EOR1 Routine	IOCSEOR	1054-1454
OPEN2 Routine	IOCSIOPEN	1455-1697
Condition Code Routine (IGEN4)	IOCSIRTAIN	1698-1846
Error Routine	IOCSERROR	1847-2094
END and CLOSE Routines	IOCSIEND	2095-2199
IOCS Index		
DTF's	TAPEFILE01	2200-2236
File Schedulers	IOCSSCED01	2237-2384
Areas and RDW's	WORKA	2385-2436
Phase 2 Programming	INIT	2437-4234
Operation Code Table		4275-4672

NOTE: Some actual locations may change with reassembly

Communication Words, Phase 2

INDEX WORD 89

Error-message option information is contained in this word.

S	0	1	2	3 - 9
+	Error Message Output		Tape Density	Not Used

Positions 0-1: Zeros indicate that messages are to be produced only on the end of the program listing. If, in addition, they are to be typed or printed, they specify positions 4 and 5 of the error-message output instruction and contain 04 if the typewriter is specified. If an on-line printer is specified, these positions contain S3, with S being the synchronizer used.

Position 2: Unchanged from system control.

INDEX WORD 95

Contents of this word are as specified by the permanent options applicable or by temporary option change

cards. The word remains unchanged throughout the four phases.

S	0	1	2 - 5	6 - 9
+ -	Obj Deck	Report List	Equip Size	Object Size

INDEX WORD 96

S	0 - 1	2 - 5	6 - 9
+ -	Not Used	Req Tbl Start	Req Tbl Stop

The sign, if plus, indicates the table was not filled; if minus, the table was filled, and all necessary symbols may not be on the table. If the word consists of all zeros, no symbols are on the table.

Table Formats, Phase 2

INDEX WORD AVAILABILITY TABLE (IW1)

This table is contained in words 0325-0334. The following example shows a reservation of index words 1, 2, and 5.

0325
+00110111111

ELECTRONIC SWITCH AVAILABILITY TABLE (ES)

This table is located in words 0335-0337. The example shows reservation of electronic switches 2 and 6.

0335
+10111101111

LITORIGIN TABLE (LITORT)

The RDW for LITORT (words 0338-0373) is LITORTBL. The increment of this table is one word. Positions 2-5 *(nnnn)* show the high counter (HICTR) at the litorigin or end control. Positions 6-9 *(xxxx)* show the object storage size.

0338
+00nnnnxxxx

SYMBOL TABLE (TABLEXZ)

The RDW of TABLEXZ (words 0374-0503) is TABLEX.

0374	0375	0376
aSYMBO	aLbbbb	+0000F1F2nnnn

Words 0374 and 0375 are the symbol in double-digit form. In word 0375, the sign becomes plus if the symbol is equated to actual. In word 0376, the sign is normally plus, but is minus for an electronic switch or a DA/DC with a minus-signed RDW.

Positions 0-2 = number of RDW's (NNN) for a DA, set to 1 for a DC
Positions 3-5 = number of words (WWW) for a DA or DC
Positions 4-5 = field definition normally (F_1, F_2)
Positions 6-9 = symbolic location (equivalent)
Positions 8-9 = number of electronic switch

OPERATION CODE TABLE (OPTBL)

The operation code table (words 4275-4672) is a system-supplied section of Phase 2. The table contains all valid 7070 Basic Autocoder mnemonic operation codes in alphameric sequence.

Each operation code requires two words. The first word is always plus, and contains the mnemonic operation in double-digit form. In the case of declarative and control operations, the second word contains a branch instruction to a specific routine for processing the entry (recognized by zeros in D1 and D2). In the case of imperative operations, the contents of the second word are as specified in the diagram.

S 0 1	2	3	4 5 6 7	8	9
Operation in Machine Language	D1	D2	Zeros Except for Augmented Ops	Reloc Indic	Zero

D1 and D2 define each operation code as to what must or may be present in the remainder of the instruction. D1 notifies the processor whether the operation code requires a component and, if so, what type. D2 specifies the allowable format of the operand. It refers to the address portion (digit positions 6 through 9) of most instructions, shift and compare digit instructions being the exceptions.

The relocation indicator in digit position 8 contains either a 1 or a 0. A 1 indicates that the instruction may have a relocatable address contained in digit positions 6 through 9. A 0 indicates that the instruction may not have a relocatable address; digit positions 6 through 9 contain part of the operation code rather than an address.

The remaining digit positions of word 2 (4-7 and 9) contain zeros unless the operation code is to be augmented by a digit in any one of these positions.

Index Words and Electronic Switches

The table of index words is shown in Figure 11, and the summary of electronic switches is given in Figure 12.

NO.	LABEL	LABEL NAME	FUNCTION
3	IOCSFIXA01	DTF IXA	17-word input, index word A
4	IOCSFIXB01	DTF IXB	17-word input, index word B
5	LOCTR	Loc. Asgn. Ctr.	Contains location assignment counter in 2-5, obj mem limit in 6-9
6	DICTR	DA & DC Counter	Counts digits in DA & DC routine
7	HICTR	Used in Litogigin	Keep track of highest location reach by LOCTR
8	CHCTR	DA & CD Counter	Keeps track of characters in DA & DC routine
9	IOCSFIXA02	DTF IXA	26-word output, index word A
10	IOCSFIXB02	DTF IXB	26-word output, index word B
11	IOCSFIXA03	DTF IXA	Symbol table output, index word A
12	IOCSFIXB03	DTF IXB	Symbol table output, index word B
13	IOCSFIXA04	DTF IXA	Symbol table from phase 1 input, index word A
14	IOCSFIXB04	DTF IXB	Symbol table from phase 1 input, index word B
15	ZEROS	Zeros	Contains zeros to zero electronic sw. and other words
16	SHIFTNO	Not Used	XSN 9
17	NOSHFT	Counter	Used to set IW availability table to 1's
18	SAVCTR	Not Used	XSN 2
19	AVCTR	Not Used	XSN 9
20	TABLE	Store Symbol Routine	Contains working RDW for storing symbols in table
21	LORCTR	Litorigin Routine	Contains working RDW for storing locations in litorigin table
22	REQTBL	Symbol Routine	Contains RDW defining symbol request table
23	EXIT	Error Routines & Others	Used as return linkage from subroutines
24	REXIT	PUT & GET	Used as return linkage from subroutines
25	IW	Operand Scan	Contains working RDW defining operand limits
26	IC	Character Scan	Contains word limit
27	IWEXIT	Reserve IW Routine	Subroutine linkage return
28	XRDWX	Process Operation	Contains working RDW for op code table lookup
29	DIGIM	Process Operation	Used for locating routine for op code types
30	DIGIT	Process Operation	2-5 contains type of operand called for by op code
31	SCANRX	Operand Scan	Subroutine linkage return
32	CEXIT	Range Check	Subroutine linkage return
33	RNGX	Range Check	Modifier to locate proper range word
34	RNGEX	IW & ES Range Check	Subroutine linkage return
35	IWEX	Reserve IW Routine	Subroutine linkage return
36	ERROR	Type Error Msg Routine	Subroutine linkage return
37	EXITC	Check Halt Char. Routine	Subroutine linkage return
38	PRONDIX	Process Operand Routine	Subroutine linkage return
39	STEPX	Bump LOCTR Routine	Subroutine linkage return
40	ACX	AC Routine	Subroutine linkage return
41	N	RDWSR Routine (DA, DC)	Holds in 2-5 count (N) of RDW's to be produced for DA or DC
42	NOLABELCTR	RDWSR Routine	Counter used to insure unique symbol when pg & line used for label
43	TSTX	Test Sign Routine	Subroutine linkage return
44	FOX	Process Field Def Routine	Subroutine linkage return
45	COUNT	Scan Routine	Holds number of characters scanned
46	SAVEACC1	Save Accumulator 1	
47	SAVEACC2	Save Accumulator 2	
48	SAVEACC3	Save Accumulator 3	
49	ERRMSGRDW	Error Message RDW	
50	MSGGATHER	Message Gather (to Address)	
51	EX	Process Symbol Routine	Subroutine linkage return
52	SPEX	Process Symbol Routine	Holds lookup RDW for symbol lkup also counts to check for full table
53	SYMCNT	Process Symbol Routine	Counter for total number of symbols
54	XPL	DA Routine	Temporary hold word for page & line
55	TEMPX	DA Routine	Temporary hold word for LOCTR
56	TEMPCTL	DC Routine	Temporary hold word for control word
57	TEMPOP	DC Routine	Temporary hold word for op code
58	TEMPC	DC Routine	Counter of number of characters on a line
59	ALFX	DC Routine	Subroutine linkage return
60	SAVEX	DC Routine	Temporary hold word for acc 1
61	TEMPERRSTX	Temporary Storage Index Word	Temporary hold word for out of range routine
62	SAVEBLKCNT	(patch on system uses IW62)	To save the blockcount on the symbol tape when the symbol request table is full
86	IOCSIXF	DIOCS XF	IOCS index word
87	IOCSIXG	DIOCS XG	IOCS index word
95	CTLWD1	Communication Word	S | 0 - 1 Printer-Punch Sync. No. | 2 - 5 Object Size | 6 - 9 Memory Size |
96	REQSIZ	Communication Word	+ Output tape has label - Output tape has no label sign made plus in phase 2 Into phase 2, contains RDW defining symbol request table. Sign minus when table exceeded max. limit Into phase 3: 0-5, Garbage; 6-9, Count of number of words on symbol table. Sign not used

Figure 11. Index Words, Phase 2

NO.	LABEL	LABEL NAME	FUNCTION
1	RELOCSW	Relocation	On indicates address of inst is relocatable
2	GETSW	GET PUT Routine	On--go to GET next record
3	COUPL	Process Op Routine	On--coupled shift inst
4	RNGSW	Range Check Routine	On--continue range check
5	SWSW	Range Check & Reserve IW & ES	On--for electronic switch instruction
6	SW2Z	Process Operand	On--zero SW2 in expanded record
7	RELSW	Rel-Equate Routine	On--instruction is an REL (Relocate)
8	EQSW	Rel-Equate Routine	On--instruction is an equate
9	SIGN	Sign Check Routine	On--sign was plus; off--sign was minus
10	SIGNL	RDW Routine	Holds sign of previous DRDW; settings on-off same as above
11	RDW2SW	Process Operand	On--6-9 addr. of RDW being processed
12	MACSW	Process Operand	On--instruction is result of a macro
13	CTLREL	Control Operation Routine	On--actual operand is result of an REL
14	SPLTSW	Process Operand Routine	On--splitshift instruction
15	SHFTSW	Process Operand Routine	On--shift instruction
16	RDW1SW	Process Operand Routine	On--2-5 addr of RDW being processed
17	LITSW	Process Operand Routine	On--literals allowed
18	BLSW	Process Operand Routine	On--blanks allowed
19	FLDSW	Process Operand Routine	On--field definition allowed
20	INDXSW	Process Operand Routine	On--indexing allowed
21	AASW	Process Operand Routine	On--address arith. allowed
22	SIGNSW	Operand Scanner (Scans)	On--scan returns with sign; off-sign always plus
23	NUMSW	Define Constant	On--indicates room available in word being packed
24	ALFSW	Define Constant	On--first word of a line being processed
25	ACSW	Define Constant	On--ADCON being processed
26	DCSIGN	Define Constant	On--previous packed

Figure 12. Electronic Switches, Phase 2

Phase 3

Storage Map

AREA	SYMBOLIC ADDRESS	ACTUAL ADDRESS
Index Word Availability Table	IWAT	0325-0334
Electronic Switch Availability Table	ESAT	0335-0337
Litorigin Table	LITORTABLE	0338-0373
Areas for Input and Output	INPUTAREA3	0374-0892
Index Word & Switch Symbol Table (to Phase 4, moved at end of Phase 3)		0374-Table Size
General Scheduler	IOCSMASK	0894-0899
Channel 1 Scheduler	IOCSC1S	0900-0934
Channel 2 Scheduler	IOCSC2S	0935-0969
EOR1 Routine	IOCSEOR	0970-1370
OPEN2 Routine	IOCSIOPEN	1371-1613
Condition Code Routine (IGEN4)	IOCSIRTAIN	1614-1762
Error Routine	IOCSERROR	1763-2010
END and CLOSE Routines	IOCSIEND	2011-2115
IOCS Index DTF's	TAPEFILE1	2116-2161
File Schedulers	IOCSSCED01	2162-2376
Areas and RDW's		2377-2572
Phase 3 Programming	BEGIN3	2573-3531
Symbol Table (randomized)	SMTBL	3536-Equip. Size
IW and ES Symbol Table (last pass)	XSTBL	3536-Table Size

NOTE: Some actual locations may change with reassembly

Communication Words, Phase 3

INDEX WORD 95

Contents of this word are as specified by the permanent options applicable or by temporary option change cards. The word remains unchanged throughout the four phases.

S	0	1	2 - 5	6 - 9
+	Obj Deck	Report List	Equip Size	Object Size

INDEX WORD 96

Positions 6-9 contain the number of symbols that were compiled in Phase 2.

S	0 - 5	6 - 9
+	000000	No of Symbols from Phase 2

Table Formats, Phase 3

INDEX WORD AVAILABILITY TABLE (IWAT)

This table is located in words 0325-0334. The diagram shows reservation of index words 1, 2, and 5.

0325
+0011011111

ELECTRONIC SWITCH AVAILABILITY TABLE (ESAT)

0335
+1011110111

This table is located in words 0335-0337. The diagram shows reservation of electronic switches 2 and 7.

LITORIGIN TABLE (LITORTABLE)

This table is located in words 0338-0373. Positions 2-5 *(nnnn)* show the high counter (HICTR) at litorigin or end control; positions 6-9 *(xxxx)* show the object storage size.

0338
+00nnnnxxxx

SYMBOL TABLE

The symbol table RDW is SMTBL. The table is located at word 3536 to equipment size. The symbols and their equivalents are placed into this table at their calculated random address. The table increment is three words.

3640	3641	3642
aSYMBO	aLbbbb	+0000F1F2nnnn

The format of the entries in this table is identical to that of the symbol table records written on the symbol tape.

INDEX WORD AND SWITCH TABLE (XSTBL)

This table (starting at word 3536) is moved to words 0374-0892 before entering Phase 4. It receives symbols for index words and electronic switches, and their equivalents, for use in Phase 4. The table increment is three words.

3536	3537	3538
aSYMBO	aLbbbb	+0000F1F200nn

The format of the entries in the table is identical to that of the symbol table records on the symbol tape.

Index Words and Electronic Switches

Figure 13 shows the index words for Phase 3, and Figure 14 is a summary of Phase 3 electronic switches.

NO	LABEL	LABEL NAME	FUNCTION
3	IOCSFIXA01	DTF IXA	Object prog.input, index word A
4	IOCSFIXB01	DTF IXB	Object prog.input, index word B
5	IOCSFIXA02	DTF IXA	Symbol input, index word A
6	IOCSFIXB02	DTF IXB	Symbol input, index word B
7	IOCSFIXA03	DTF IXA	Object progr.output, index word A
8	IOCSFIXB03	DTF IXB	Object prog.output, index word B
9	IOCSFIXA04	DTF IXA	Symbol output, index word A
10	IOCSFIXB04	DTF IXB	Symbol output, index word B
11	IOCSFIXA05	DTF IXA	Messages output, index word A
12	IOCSFIXB05	DTF IXB	Messages output, index word B
13	ERRMSGCTLX	Error Message Control Index Word	Constant for identification and print control of messages
14	LOXSTBL	Low Index Word and Switch Counter	Contains RDW of table of symbolic index words and switches
15	HIXSTBL	High Index Word and Switch Counter	Contains loc.to store next symb IW or ES (indexing portion)
16	OFLOWCTR	Overflow Counter	Count of number of symbols on symbol tape
17	STOREDCTR	Stored Counter	Contains capacity factor
18	EXIT	Exit	IW used to store exit location when exiting to various subroutines
19	CALCADDR	Calculated Address	Stores calculated random address of symbol being stored
20	WRAPADDR	Wrap Address	Contains the lower limit of the symbol table in storage
21	TLUEXIT	Table Look-Up Exit	IW used when BLXing in and out of table lookup subroutine
22	TLURDW	TLU RDW	RDW of table of symbolic index words and switches
23	SAVEACC1	Save Accumulator 1	Preserve contents of accumulator before entering error routine
24	SAVEACC2	Save Accumulator 2	
25	SAVEACC3	Save Accumulator 3	
26	ERROR	Error	Linkage to type error message subroutine
27	ERRMSGRDW	Error Message RDW	RDW of work area for messages
28	MSGGATHER	Message Gather	To-address for moving messages
29	SYMOUTCTR	Symbol Output Counter	Symb.output ctrl.- contains addr. in output area of ovflo symb.file where next ovfl symb.may be stored
30	DUPX		Location for editing
31	TEMPSAVE3	Temporary Save	Temporary hold for first word of symbol
32	ATCTR	Availability Table Counter	Contains count of word in electronic switch & index word avail. tables
33	SHCTR	Shift Counter	Contains count of digit within ATCTR word being tested for avail. of associated electronic switch or index word
34	RANGEX	Range Index Word	Contains parameters of allowable range of a symbolic component set up prior to entering range test subroutine
35	TEMPFD	Temporary Field Define	Temporary storage for field definers during range testing
36	RDWCTR	RDW Counter	Count of RDW's to be generated by DA or DC header line
86	IOCSIXF		
87	IOCSIXG		
93	RAMFACTOR	Random Address Factor	Precalculated factor used in calculating random address for storing symbols and their equivalents
95	CONTROL	Control	Store indicative information to be passed from phase to phase
96	SYMSIZE	Symbol Size	

Figure 13. Index Words, Phase 3

NO.	LABEL	LABEL NAME	FUNCTION
1	STORANYWAY	Store Anyway	Turned on at beginning of each pass; off when capacity factor exceeded (except last pass)
2	LASTPASSW	Last Pass Switch	Turned on to indicate last pass of phase 3
3	UNDEFSW3	Undefined Switch	Turned on to indicate processing of undefined operand message
4	WRAPSW	Wrap Switch	Turned on when the search has reached the upper limit of the table from CALCADDR and is continuing from the lower limit to CALCADDR
5	SW1SW	Switch 1 Switch	Turned on to indicate table lookup is for an index word or electronic switch
6	RANGESW	Range Switch	Turned on if permissible range is described by more than one word
7	ADDRSW	Address Switch	Turned on if range testing an address, off if range testing a symbolic component
8	ENDRSW	End Routine Switch	Turned on to indicate ENDROUTINE is in process
9	CDSWITCH	Compare Digit Switch	Indicates instruction being assembled is CD, field definition to be inserted in digit 5 only
10	INDEXSW	Index Switch	Indicates symbolic switch is to be searched for in table
11	FD2SWITCH	Field Definition - Digit 2 Switch	Controls exit from routine for calculating relative field definition so that same routine may be used for calculation of both digits 4 and 5. On indicates D5 is being calculated
12	FDERRSW	Field Definition Error Switch	Controls typing of field definition error messages so that only one message is typed even though multiple errors may be encountered

Figure 14. Electronic Switches, Phase 3

Phase 4

Storage Map

AREA	SYMBOLIC ADDRESS	ACTUAL ADDRESS
Index Word Availability Table	IWAT	0325-0334
Electronic Switch Availability Table	ESAT	0335-0337
Litorigin Table	LITORREC	0338-0373
Index Word and Switch Symbol Table	XSRECORD	0374-size
Communication Word (all phases)	893	0893
General Scheduler	IOCSMASK	0894-0899
Channel 1 Scheduler	IOCSC1S	0900-0934
Channel 2 Scheduler	IOCSC2S	0935-0969
EOR1 Routine	IOCSEOR	0970-1370
OPEN2 Routine	IOCSIOPEN	1371-1613
Condition Code Routine (IGEN4)	IOCSIRTAIN	1614-1762
Error Routine	IOCSERROR	1763-2010
END and CLOSE Routines	IOCSIEND	2011-2115
IOCS Index DTF's	TAPEFILE1	2116-2143
File Schedulers	IOCSSCED01	2144-2284
Areas, Constants, and RDW's	OBJINAREA	2285-2918
Phase 4 Programming	INITLIT	2919-4088
Literal Table	LITERALTBL	4089- beginning of F.D. error table
Field Definition Error Table		End of LITERALTBL- Equip. Size

NOTE: Some actual locations may change with reassembly

Communication Words, Phase 4

INDEX WORD 89

This index word contains error-message option information.

S	0	1	2	3 - 9
+	Error Message Output		Tape Density	Not Used

INDEX WORD 95

Contents of this word are as specified by the permanent options applicable, or by temporary option change cards. The word remains unchanged throughout the four phases.

S	0	1	2 - 5	6 - 9
+ -	Obj Deck	Report List	Equip Size	Object Size

INDEX WORD 96

Positions 0-1 are the channel and unit address of the listing tape. Positions 2-5 are the channel and unit address of the input tape for double-channel and single-channel assembly. Positions 6-9 are the number of words contained in the index word and switch symbol table.

S	0 - 1	2 - 5	6 - 9
+	C U List	Loc of Input	Size of IXSW Table

Table Formats, Phase 4

INDEX WORD AVAILABILITY TABLE (IWAT)

This table is located in words 0325-0334. The example shows reservation of index words 1, 2, and 5.

0325
+00110111111

ELECTRONIC SWITCH AVAILABILITY TABLE (ESAT)

This table is located in words 0335-0337. The example shows reservation of electronic switches 2 and 5.

0335
+10110111111

LITORIGIN TABLE (LITORREC)

This table (words 0338-0373) contains, in positions 2-5 *(nnnn)*, the high counter (HICTR) at litorigin or end control, and in positions 6-9 *(xxxx)*, the object storage size.

0338
+00nnnnxxxx

PACK TABLES

These tables are used to reserve field definition for packing literals. Since literals must have the same sign to be packed in any given word, three tables control the packing (one for each sign). The contents of the tables are shown as they would appear after assembly of a program using four literals. They are: +0, aABC, +1, −10.

ALPHAMERIC PACK TABLE (ALFATABLE)

2782	2783
+0000033952	a99999

This table is located in words 2782-2786.

In word 2782, positions 4 and 5 contain the field definition of unused positions available in the word defined by positions 6-9. Positions 6-9 contain a location in storage assigned to this pack table for packing alphameric literals.

Word 2783 is shown as initialized. No packing has been assigned to this word. All words are initialized to alphameric 9's.

MINUS PACK TABLE (MINUSTABLE)

2787	2788
−0000083953	a99999

This table is located in words 2787-2796.

In word 2787, positions 4 and 5 contain the field definition of unused positions available in the word defined by position 6-9. Positions 6-9 contain a location assigned to this pack table word for packing minus literals.

Word 2788 is shown as initialized. No packing has been assigned to this pack-table word. It was initialized to alphameric 9's.

PLUS PACK TABLE (PLUSTABLE)

2797	2798
+0000083951	a99999

This table is located in words 2797-2816.

Word 2797, positions 4 and 5, contain the field definition of unused positions available in the word defined by positions 6-9. Positions 6-9 contain the location assigned to this pack-table word for packing plus literals.

Word 2798 is shown as initialized. No packing word has been assigned to this pack-table word. It was initialized to alphameric 9's.

LITERAL TABLE

This table is located from word 4089 to the beginning of the field definition error table. It is used to store literals, as they are found in the 26-word expanded records. A location and field definition are assigned to the literal, and also placed in the table. When end control is recognized, the literals are packed using the assigned field definition and location. The RDW for the literal table is LITERALTBL. The examples show the literal table before packing the same literals indicated in the pack tables.

4089	4090	4091	4092
+0000000000	+0000993951	abbABC	+0000493952

4093	4094	4095	4096
+0000000001	+0000883951	−0000000010	+0000893953

The next examples show the literal table after packing, using the same literals indicated in the pack tables and the before-packing storage map.

4089	4090	4091	4092
+0000000010	abbABC	−0000000010	+0000493952

4093	4094	4095	4096
+0000000001	+0000883951	−0000000010	+0000893953

NOTE: Words 4092 to 4096 are identical before and after packing because they were not overlaid by the packing procedure.

FIELD DEFINITION ERROR TABLE

This table is used to store the page and line (PGLIN) of statements which have Phase 4 field definition or relocation errors. The table is built from equipment storage size until it reaches the end of the literal table. The table is not zero'ed out in initialization.

The increment of the table is one word. The page and line are stored in double-digit form. The sign is alpha for field definition errors and minus for relocation errors. The following illustrates three errors; two field definition errors and one relocation error.

9989	
a9090919290	Field definition error, PGLIN 00120.
9988	
−9092919490	Relocation error, PGLIN 02140.
9987	
a9096949290	Field definition error, PGLIN 06420.
9986	
Immaterial	Remains of previous phase.

Index Words and Electronic Switches

Figures 15 and 16 summarize the index words and electronic switches assigned to Phase 4.

NO	LABEL	LABEL NAME	FUNCTION
3	IOCSIXF	DIOCS IXF	IOCS index word
4	IOCSIXG	DIOCS IXG	IOCS index word
5	IOCSFIXA01	DTF IXA	Object prg . input, index word A
6	IOCSFIXB01	DTF IXB	Object prg. input, index word B
7	IOCSFIXA02	DTF IXA	List tape output, index word A
8	IOCSFIXB02	DTF IXB	List tape output, index word B
9	IOCSFIXA03	DTF IXA	Condensed card output, index word A
10	IOCSFIXB03	DTF IXB	Consensed card output, index word B
11	LITBLIW	Literal Table Index Word	Contains adjusted RDW of literal table
12	LITBLX	Literal Table Index Word	Contains address of next available location in literal table
13	LTCTR	Literal Table Counter	Address in table of assigned literal
14	LLCTR	Literal Location Counter	Adjusted actual address of assigned literal
15	LITORCTR	Litorigin Counter	Contains count of no. of litorigins used in program being assembled
16	LITHLD	Literal Hold	Temporary storage for literal being processed
17	PACKX	Packing Index Word	Indicates packing table to be used (plus, minus, or alpha)
18	ALFA9	Alphabetic Nine	Constant of @9999999999 for clearing tables
19	EXIT	Exit	Linkage to various subroutines
20	ADDREX	ADDRESS Index Word	Controls setting up tape channel and units nos. in DTF's. Addresses change when more than one pass in phase 3 occurs.
21	FDERRCNTX	Field Definition Error Count Index Word	Count of field definition and relocation errors
22	LOADX	Load Index Word	No longer used
23	COUNT	Count	Counts load program cards in output
24	EDITX	Edit Index Word	Set up for editing (index word used in RG & RS instr.)
25	CDCTR	Card Counter	Card number counter for condensed card output
26	ZERO	Zero	Constant of plus zeros
27	BLANKXWD	Blank Index Word	Constant of alpha blanks
28	BLANKX	Blank and X	Constant of four alpha blanks and alpha X
29	LINECT	Line Count	Counter for no. of print lines on output listing page
30	ASTERBLANK	Asterisk Blank	Constant of * followed by four alpha blanks
31	WORKX	Word Index Word	General purpose index word
32	INSTCTR	Instruction Counter	Controls maximum no. of instructions per condensed card
33	OPERANDX	Operand Index Word	Loc. of first word in operand - used for editing
34	ERRORX	Error Index Word	Constant of alpha word ERROR
35	DCEXIT	Define Constant Exit	Linkage to subroutine for processing DC extra lines
36	LUXS	Look up Index Word & Switch	RDW of symbolic index word and switch table
37	ERROR	Error	Linkage to subroutine to type error messages
38	FDERRTBLX	Field Definition Error Table Index Word	Address of next location to be used in field definition error table
39	EDIXIT	Edit Exit	Linkage to subroutine for editing output listing line
40	CONDEXIT	Condense Exit	Linkage to subroutine for preparing condensed card output
41	PRINTCT	Print Count	Cycle count of on line printer operations - two print cycles per line required by 7400
42	SEQCTR	Sequence Counter	Count of no. of instructions on condensed card
43	CLODTAPEX	Condensed Load Tape Index Word	Start address of condensed load tape output area
44	PUNCHEXIT	Punch Exit	Linkage to literal output routine
45	RELOCHOLD	Relocation Indicator Hold	Temporary storage for relocation indicator
46	ATCTR	Availability Table Counter	Counter of words in electronic-switch and index-word availability tables
47	MOVX	Move Index Word	Contains constant of zero when listing available index words Contains constant of +100 for translating electronic switches
48	SHCTR	Shift Counter	Counts no. of shifts performed when developing switch and index-word availability listing
49	INITHDRX	Initialize Header Index Word	Used to initialize header line to start a new listing page
50	ERRMSGRDW	Error Message RDW	RDW of message work and output area
51	MSGHDRX	Message Header Index Word	To address for moving the message header line
52	MSGGATHER	Message Gather	To address for moving messages
53	SAVEFDCNT	Save Field Definition Error Count	Temporary hold for count of error messages
88	LEVEL	Level Index Word	Contains version and level number of system (alpha)
95	CONTROL	Control	Contains control information from phase 3
96	SIZE	Size	Contains control information from phase 3

Figure 15. Index Words, Phase 4

NO.	LABEL	LABEL NAME	FUNCTION
1	DONESW	Done Switch	
2	NOTYPMSGSW	No Typed Message Switch	Turned on when messages are to be written on the listing only
3	CDSWITCH	Compare Digit Switch	Indicates processing of compare digit instruction
4	PENDSW	Pending Switch	Indicates DC extra line pending
5	LITBLFULSW	Literal Table Full Switch	Turned on when no more space is available for literal table
6	ADCONSW	Address Constant Switch	Indicates processing of address constant type literal
7	FD2SWITCH	Field Definition 2 Switch	Indicates processing of second digit of field definition
8	FDERRSW	Field Definition Error Switch	Indicates field definition error has been encountered
9	FRSTADCON	First ADCON	Indicates processing of first address constant in a packed word
10	FORCESW	Force Switch	Bypasses packing of DC ADCONS if next card is type 6
11	XTRASW	Extra Switch	Hold location for relocation indicator
12	LITRANGCK	Literal Range Check	Turned on when literal addresses exceed object size
13	PRINT1SW	Printer 1 Switch	Indicates first cycle of two-cycle 7400 printer operation
14	PRINTERRSW	Printer Error Switch	Indicates 7400 printer error
15	FULLSW	Full Switch	Indicates a full condensed card
16	SPACESW	Space Switch	Indicates extra space on output listing
17	AVAILSW	Available Switch	Indicates electronic switches or index words available
18	NOMSGSW	No Message Switch	Turned on to indicate there were no messages in phase 4
19	ENDFDMSGSW	End Field Definition Message Switch	On to indicate that all the field definition error messages have been processed
21	PUNCHSW	Punch Switch	Indicates on-line punching desired
22	PRINTSW	Print Switch	Indicates on-line printing desired
23			Test 30 BRANCH on BES to BLANKD23, BLANKOP, BLANKINST, BLANKLOC per print control
24			
25			
26			

Figure 16. Electronic Switches, Phase 4

The following is a list of the unique IOCS macro names found in the macro table (MACTBL) in Phase 1. The variations of IOCS specifications make it necessary to accommodate variations in FILEFORM, FILETYPE, and TIOAREAS options obtainable from the DTF's. Also, the specific DUF macro required is determined by the type of unit record equipment and whether SPOOL is used.

As the DTF's are read in Phase 1, the symbolic file name, FILEFORM, FILETYPE and TIOAREAS are placed into the DTF table (DTFTBL). A count of the number of files is also kept. When the last DTF record has been processed, the information in the DTFTBL is used to generate the proper macro name from the basic name (999AA). Addition is used on the double-digit, alphameric, basic name to derive the proper macro name. The factors to be added are determined by a comparison of the variables with a compare-digit operation.

A scheduler macro-name is developed for each file stored on the DTFTBL. The macro is then found on the library, and the skeleton instructions in the macro are processed.

For the IOCS macros GET, PUT, PUTX and RLSE, the title name from the operand of the macro is used to locate the DTF information needed for the statistics to construct the proper macro name.

File Schedulers, Basic Name (999AA)

999AB	Forms 1 and 2, 1 Area
999AC	Forms 1 and 2, 2 Areas
999AD	Forms 1 and 2, 3 Areas
999BB	Form 3, 1 Area – Input
999BC	Form 3, 2 Areas – Input
999BD	Form 3, 3 Areas – Input
999CB	Form 3, 1 Area – Output
999CC	Form 3, 2 Areas – Output
999CD	Form 3, 3 Areas – Output
999DB	Form 4, 1 Area
999DC	Form 4, 2 Areas
999DD	Form 4, 3 Areas

GET, Basic Name (G10KB)

G11LB	Forms 1, 2
G11MB	Form 3
G11NB	Form 4
G21LB	GET File to Work Area, Forms 1, 2
G21MB	GET File to Work Area, Form 3
G21NB	GET File to Work Area, Form 4
GS0KB	GET Card File
GT0KB	GET Card File to Work Area

PUT, Basic Name (P10SB)

P11TB	Forms 1, 2
P11UB	Form 3
P11VB	Form 4
P12TB	File in File, Forms 1, 2
P12UB	File in File, Form 3
P12VB	File in File, Form 4
P21TB	PUT Work Area in File, Forms 1, 2
P21UB	PUT Work Area in File, Form 3
P21VB	PUT Work Area in File, Form 4
PL0SB	Card File in Punch File
PS0SB	Punch File
PT0SB	Work Area in Punch File
PS1TB	Tape File in Punch File, Forms 1, 2
PS1UB	Tape File in Punch File, Form 3
PS1VB	Tape File in Punch File, Form 4

PUTX, Basic Name (X10SB)

X12TB	File to File, Forms 1, 2
X12VB	File to File, Form 4

RLSE, Basic Name (R10BK)

R11BK	Form 3
R11BL	Forms 1, 2 and 4 Input
R11BM	Forms 1 and 2 Output
R11BN	Form 4 Output

DUF, Basic Name (DUFbb)

DUF1	GET Card File or Card File to Work Area, No SPOOL
DUF2	PUT or PUT A in B, No SPOOL
DUF3	GET Card File or Card File to Work Area, SPOOL
DUF4	PUT or PUT A in B, SPOOL

Appendix

Glossary

ASSEMBLY RUN: Execution of the Autocoder program using source entries as input, resulting in the production of a condensed card-record program tape and a listing tape containing all necessary information from the source entries and the assembly.

AVAILABILITY TABLE: A table that retains allocation of specific components during assembly. Two tables are formed, one for index words, and the other for alteration switches.

COMMUNICATION WORD: A word of data that retains information from one phase to another.

DECLARATIVE ENTRY: A source entry to an assembly program that provides the processor with necessary information to produce the assembled instructions. The declarative entry may define an area, a constant, or a machine component, but will not produce, by itself, an executable instruction.

DETAIL CARD: A change record preceded by a header card used during a system run, containing an alteration to be made in a record on the system tape.

EQUIVALENT: The actual machine location assigned to a symbol.

EXPANDED RECORD: The record created by Phase 2 and used by the Autocoder program through Phase 4. It is 26 words in length, contains the source entry, switches controlling further operations on the statement, and a breakdown of the source statement for use during the assembly.

HEADER CARD: A record that initializes for changes made to any record on the system tape during a system run.

IMPERATIVE ENTRY: A statement that produces, on assembly, one word of machine operational digits. This word, in operation, causes the instruction to be performed.

LIBRARIAN: That portion of the Four-Tape Autocoder program which alters and updates the system tape during a system run.

LIBRARY: A series of records on the Autocoder program tape that contain the macros and subroutines in skeleton-instruction form.

LOCATION ASSIGNMENT COUNTER: A counter used in Phase 2 to assign actual machine locations to source entries that require allocation in storage.

MACRO: An open-ended sequence of machine instructions produced by a processor on recognition of a source-language statement. These instructions can perform a function defined by the parameters given in the source statement. They may consist, in part, of a linkage to a closed subroutine.

MACRO-INSTRUCTION: A source-language statement to a processor resulting in the production of a variable number of instructions in machine language. These instructions can perform a given function in the object program.

PACK: The process of placing a number of separate constants or literals into a word of storage, to utilize efficiently the storage necessary for a given object program.

PARAMETER: A field in the operand of a macro-instruction that directs the construction of a macro for performance of a unique function.

RANDOMIZE: To calculate a unique address from a symbolic name. If information dealing with the symbol is placed at the calculated location, it may later be found by using the same calculation, thereby allowing information retrieval.

RANDOMIZING FACTOR: The common computations necessary in randomization to place the results in a limited portion of storage.

REASSEMBLY RUN: Execution of the Autocoder program using a program listing as input and correction cards to modify the input records. The run results in the production of an updated listing tape and a new condensed card-record program tape.

SUBROUTINE: A routine that is common to the program being entered and left by means of linkage in the program.

SYSTEM RUN: Execution of the Four-Tape Autocoder program resulting in the production of two updated (identical, if no change cards are used) Four-Tape Autocoder program tapes.

Abbreviations

AA	address arithmetic
Act	actual
Addr	address
Alpha	alphameric
Asmd	assembled
Avail	availability
C	channel
Calc	calculate
Chars	characters
Ctrl	control
D	digit
Def	definition
Dupl	duplicate
EOJ	end of job
Equiv	equivalent
ES	electronic switch
Exp	expanded
FD	field definition
Fld	field
Fnd	found
Incr	increment
Info	information
Inst	instruction
IW	index word
Lib	Library
Lkup	look-up
Loc	location
Lt	literal
Msg	message
No	number
OP	operation
OPND	operand
Opt	option
Ovflo	overflow
Pos	position
Prg	program
Rdr	reader
Rec	record
Req	request
Sw	switch
Symb	symbol, symbolic
Sync	synchronizer
Sys	system
Tbl	table
Tp	tape
Temp	temporary
U	unit
Wd	word

IBM 7070 Series Four-Tape Printed in U.S.A. C28-6139

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