S

IBM

Programming Systems Analysis Guide

7070 Input/Output Control System

IBM

Programming Systems Analysis Guide
7070 Input/Output Control System

©1960, 1961 by International Business Machines Corporation

Preface

This manual was prepared by Applied Programming to provide detailed information on the internal logic of the IBM 7070 Input/Output Control System (IOCS). 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. It is assumed that the reader has a basic knowledge of the 7070 Autocoder language. Such background can be obtained from the *7070/7074 Four-Tape Autocoder Reference Manual,* Form C28-6102. It is also assumed that the reader has a general knowledge of the objectives of the IOCS system as described in the *IBM 7070 Input/Output Control System Bulletin,* Form J28-6033-1.

Labels from the program listing are used in the text and figures to provide reference to the IOCS routines in the listing. Labels are shown in capital letters to simulate their appearance in the listing. Labels generated by Autocoder 76 contain a period in place of the "s" in IOCS and may be slightly different from those generated by Autocoder 74.

A listing may be obtained by assembling a program using IOCS macros with either Autocoder 74 or Autocoder 76 and printing the listing tape.

Contents

IBM 7070 Input/Output Control System
Correlation between IOCS Routines . . . 7
File and Channel Schedulers . . . 8

Operational Description of Routines . . . 10
Major Macros GET, PUT, and PUTX . . . 10
Channel and File Schedulers . . . 10
RLSE Macro (Release) . . . 13
Condition Code Routine . . . 14
Error Routine . . . 15
End of Reel Routine . . . 17
Open Routine . . . 19
End and Close Routine . . . 21
Checkpoint and Restart Routines . . . 22
BSP, RWD, WSM, and WTM Macros . . . 24
RDSF and RDSB Macros . . . 25
RDLIN Macro . . . 25
DEOR Macro . . . 26
FEORN Macro . . . 26
FEOR Macro . . . 27
Unit Record Card Files . . . 27

Flow Charts . . . 29
Figure 1. Correlation between IOCS Routines . . . 29
Figure 2. General Operation of Schedulers . . . 30
Figure 3. GET, PUT, and PUTX Macros . . . 31
Figure 4. Channel and File Schedulers . . . 32
Figure 5. RLSE Macro . . . 33
Figure 6. Condition Code Routine . . . 34
Figure 7. Error Routine . . . 36
Figure 8. End of Reel Routine . . . 38
Figure 9. Open Routine . . . 41
Figure 10. End and Close Routine . . . 43
Figure 11. Checkpoint and Restart Routines . . . 44
Figure 12. BSP, RWD, WSM, and WTM Macros . . . 45
Figure 13. RDSF and RDSB Macros . . . 46
Figure 14. RDLIN Macro . . . 46
Figure 15. DEOR Macro . . . 47
Figure 16. FEORN Macro . . . 47
Figure 17. FEOR Macro . . . 47

Memory Locations . . . 48
Storage Map . . . 48
DTF Contents . . . 49

Program Condition Analysis Aids . . . 51
Special Operational Situations . . . 51
Index Words . . . 51
DTF Fields . . . 51
File Scheduler . . . 52
Channel Scheduler . . . 52
Holding Loops . . . 52
Temporary Storage Words . . . 52

Appendix . . . 53
Glossary . . . 53
Abbreviations . . . 54

The Input/Output Control System for the IBM 7070 Data Processing System is comprised of program routines written by the IBM Applied Programming Department to provide users with efficient, pretested routines for reading and writing card and tape records.

Programming of input and output routines that handle records efficiently is difficult. The routines used in IOCS have been found through experience to be efficient. By using this system in all programs, standard input and output routines are provided. Such routines simplify and standardize console operations.

IOCS provides the following features while satisfying the requirements for reduced programming, efficient routines, standardization, and elimination of input-output programming errors:

1. Reading and writing of tape records simultaneously with processing.
2. Macro-instructions that handle records sequentially, even though they are in blocked form on an input tape, or are to be written in blocked form on an output tape.
3. Checks for proper mounting of input tapes and aids in the checking of each tape used. By the use of label records, each reel of tape may be identified and checked before being used in the program.
4. Provision for interrupting the program and continuing it later by the use of checkpoint and restart routines.
5. Routines for processing unit records. Unit records may be read, punched, or printed on-line using macros.
6. Error routines for tape records, that correct errors whenever possible.
7. SPOOL programs that can be run concurrently with programs that use the Input/Output Control System.

The functions provided by IOCS are incorporated into the user's program during assembly by IBM 7070 Autocoder. Certain precautions are necessary when assembly is by IBM 7070 Four-Tape Autocoder. See IOCS Bulletin, Part III.

Whenever programming or machine errors occur during an input or output operation, they are often difficult to diagnose because of the complexity of these operations. If IOCS is being used as a standard input-output routine, a thorough understanding of how it operates becomes desirable in order to be able to diagnose quickly any operation difficulties that might occur in this area.

The manual describes general as well as detailed flow charts to aid in understanding IOCS operation.

Correlation between IOCS Routines

The over-all flow of processing of the IOCS program is shown in Figure 1. This flow chart shows the interrelationship between the following subroutines:

1. File Scheduler
2. Channel Scheduler
3. Condition Code
4. Error
5. End of Reel
6. OPEN
7. END or CLOSE

The following description of the IOCS program explains each block in Figure 1.

Block 001, File Scheduler: This logic block represents the most important logical function in IOCS. One file scheduler is produced for each file specified during autocoder assembly and controls the reading or writing for that file. It is entered from the IOCS macros (GET, PUT, PUTX, or RLSE) and from priority branch control on a priority interrupt operation. It is also entered from block 002, the channel scheduler. Its exits are to test for error if the last I-O operation was not a correct length record, to the condition code routine if condition codes 3 through 7 occur, to the channel scheduler, and to return to the main program.

Block 002, Channel Scheduler: One channel scheduler is produced for each tape channel used, during Autocoder assembly. The channel scheduler can be entered only from the file scheduler except for re-entry from SPOOL. These blocks are very closely associated and control passes back and forth between them several times depending upon the existing conditions. Each channel scheduler will have associated with it, the file schedulers associated with the same channel. The three exits from the channel scheduler are priority release, SPOOL program, and a return to the file scheduler.

A clear understanding of the operation of the file and channel schedulers is required to understand thoroughly the IOCS program. They are described in general and in detail in this manual to assist in understanding their operation.

Block 003, Error: If the condition code of the final status word for the last input-output operation was not 2 (CLR) a test is made at IOCSIRTAIN to determine if it is 0 or 1 (error). If not, the condition code must be 3-7, and after modifying the availability switch for output, or the file scheduler exit for input files, control returns to the file scheduler and then to the channel scheduler.

Block 004, Error Routine: If an error was indicated by block 003, this block attempts to correct the error, and types out messages and halts if unable to do so. The operator may then manually correct the errors.

Block 005, Odd Length Record: If the previous record was not CLR and not an error (0 or 1), a test for condition code 3 through 7 is made. The condition code is compared to 4. If the low indicator is turned on, it is a short length record; if equal, a long length record. The user has the option of supplying special routines of his own for these conditions, ignoring the condition code 3 or 4, or having a message typed with a halt.

Block 006, End of Segment: If the condition code was neither of the above (LLR or SLR), it is now compared with the digit 6. If it is found equal, it is an end-of-segment condition. The user has the same options for this condition as those listed in block 005.

Block 007, End of Reel: Condition code 5 signifies an end-of-reel condition, and is sensed after the comparison to the digit 6, by a branch if low. If it was found high, it is condition code 7 (short character length record). Options to branch to user's routines are available in all cases. See the flow charts (Figures 6a and 6b).

Block 008, Process Trailer Record: Trailer label records are the last records on a tape reel and indicate whether a reel is the last reel of a file or whether it is to be followed by other reels. For greater detail see EOR flow chart (Figure 8a).

Block 009, End of File: If end of file is recognized in the trailer label, this block provides linkage to the user's EOF routine. The user must provide such a routine. He may return by branching to 0+IOCSIXG. If not EOF, control passes to block 010 to process the header of the next reel (if used).

Block 010, Process Header: If this is an input file, this block processes the header label (first record) to determine if this is the correct reel to process, and types an appropriate message. If it is an output file, it determines if the tape may be written on. If the tape should not be used, the operator is informed by a typed message and a halt.

Block 011, Write Last Record: When a CLOSE macro is encountered in the main program, the file specified is closed and removed from use by the main program. If it was an output file, any records remaining in the output areas are written on the output tape, and a tape mark is written following the last output record. Block 008 is then entered to process the trailer record.

Block 012, Program Switch: If Block 008 was entered during a CLOSE or END operation, this switch would be set to allow a return branch to the CLOSE routine.

Block 013, Make File Inactive: After processing the trailer label record in a CLOSE operation, the file is made inactive by inserting a zero in the ACTIVITY field of the DTF and the tape rewound as specified by the CLOSEPROC entry. Control then returns to the main program. The CLOSE flow chart, Figure 10, gives details.

Block 014: The ACTIVITY field in the DTF's of the files being opened are set to 3. This allows the test of the OPEN routine to initialize the files listed in the macro-instruction.

Block 015, Program Switch: If block 010 was entered to process the header record on an OPEN operation, this switch is set to cause a branch back to the OPEN routine.

Block 016, Initialize File Schedulers: This block causes a 1 to be inserted into the ACTIVITY field of the DTF and prepares the file scheduler for operation using information from the DTF. This logic block is entered from the switch (block 015) during the processing of an OPEN macro. See OPEN flow chart (Figure 9a).

Blocks 017, 018, 019, 020, User's Routines: Exits to these routines are optional except for the EOF routine. They are supplied by the user so that he may process unusual records. These routines must be terminated with the proper return branches as stated by the file specifications in the IOCS bulletin. The SLR, LLR, and SCLR routines return to the first instruction following the macro in the main program, but the other routines return to the proper locations in the file scheduler.

File and Channel Schedulers

The file schedulers and channel schedulers have complete control over tape input and output operations and the scheduling of tape files using the same tape channel. Scheduling of the tape units using the same channel is done by one channel scheduler. At the present time the IOCS program uses a maximum of two channel schedulers, one for channel 1 and one for channel 2. There is one file scheduler for each tape file. The file scheduler controls the areas (1, 2 or 3) used in the processing of records.

This section describes the interrelationship between the main program containing GET and PUT macros and the channel and file schedulers, as shown in the gen-

eral flow chart, Figure 2. Each block in the flow chart is discussed with reference to its operation for a GET and a PUT macro.

Block 026, GET: The user's program requests another data record for processing.

Block 026, PUT: The user's program requests that the data record just processed be added to an output area.

Block 027, GET: A check is made to determine if another data record is available for processing from the input area now being processed.

Block 027, PUT: A check is made to determine if a data record location is available in the output area now being filled with processed data records.

Block 028, GET: A data record is made available to the main program.

Block 028, PUT: The data record is added to the output area.

Block 029, GET: No data records are available for processing in the input area being processed. Before checking for another input area containing data records, the IOCS program will attempt a tape read operation for the tape specified by the GET macro. Block 029 tests the channel associated with this input tape to determine if it is busy.

Block 029, PUT: No locations are available for processed data records in the output area now being filled. Before checking for another output area, the IOCS program will attempt to write the completed area on the tape unit specified in the PUT macro. Block 029 tests the channel associated with this output tape to determine if it is busy.

Block 030, GET: The channel is found not busy. Therefore, a tape read operation is initiated for the tape specified by the GET macro.

Block 030, PUT: The channel is found not busy. Therefore, a tape write operation is initiated for the tape specified by the PUT macro.

Block 031, GET: When block 029 finds that the channel is busy, block 031 performs two functions. First, it turns on the pending switch, if it is not on, in the file scheduler for the input tape specified in the GET macro, and adds 1 to the pending counter. During an interrupt the pending switches will be checked to determine which input file has input areas ready to receive records. Secondly (and this occurs when coming from either block 029 or block 030), a test is made to determine if another input area has records available for processing. This is done by checking the availability switch, a switch and counter combination, that is turned on and incremented each time a file has been checked for unusual conditions and is ready or "available" for processing.

Block 031, PUT: When block 029 finds that the channel is busy, block 031 performs two functions. First, it turns on the pending switch, if it is not on, in the file scheduler for the output tape specified in the PUT macro, and adds one to the pending counter. During an interrupt, the pending switches will be checked to determine which output file has completely filled output areas. Secondly, and this occurs when coming from either block 029 or block 030, it checks to determine if there is another output area that is available for inclusion of output data records. It does this by checking the availability switch in the file scheduler for the output tape specified in the PUT macro.

Block 032, GET: No block of data records is available for processing, thereby causing the user's program to wait until a block of records is made available. The force switch is set to force this tape file to be read next.

Block 032, PUT: No output area is available to receive processed data records. The user's program must wait for an output area to be made available (i.e., able to receive processed data records). The force switch is set so that this tape file will be written on next.

Block 033, GET: This is a loop in the channel scheduler that continually checks the availability switch of the file scheduler specified in the operand of the GET macro-instruction. After a block of data records has been read into an input area and the tape operation checked for unusual conditions, the availability switch is turned on and the user's program will process the first data record in the new data block.

Block 033, PUT: This is a loop in the channel scheduler that continually checks the availability switch in the file scheduler. After a block of records has been written and the tape operation checked for unusual conditions, the availability switch is turned on and the user's program will place the next data record processed into the output area made available.

Block 034, GET: The next unprocessed area is made available to the GET macro by placing its locating addresses in the index word used by the macro for deblocking.

Block 034, PUT: The next unused area is made available to the PUT macro by placing its locating addresses in the index word used by the macro for blocking.

Major Macros GET, PUT, and PUTX

Three input and output instructions are available through IOCS: GET, PUT, and PUTX. When one of these macro-instructions is encountered in the user's source program during assembly, a series of generated instructions is inserted into the main program. The PUTX macro-instruction differs completely from the others in that it involves an exchange of RDW's.

The GET and PUT macro-instructions can be divided into two logical groups: those that make a data record ready for use and those that move the data record as well as making it ready for use. In the first group are the GET and PUT macro-instructions that name one file in the operand portion of the macro-instruction. The second group of GET and PUT macro-instructions contains two files or a work area and a file in the operand.

Example:

```
GET     INPUT FILE A
PUT     OUTPUT FILE B
GET     INPUT FILE A TO WORK AREA
PUT     WORK AREA IN OUTPUT FILE B
```

All three macro-instructions follow similar logic in determining if a block has been completely used, and in yielding control to the schedulers if it has. However, the instructions generated to perform this logic may differ depending on the macro-instruction used and also the form of the records. The logic of each macro is different if entry into the schedulers is not required. These similarities and differences are illustrated in Figure 3.

Block 051: A GET, PUT, or PUTX is encountered in the user's main program.

Block 052: A test is made to determine if the area has been completely used. Each macro may use different coding to perform this test, depending upon the type of macro-instruction (i.e., GET, PUT or PUTX) and upon the form of the records.

Block 053: All data records (or places for a data record) have been used. Another area must be made available, so control branches to the file scheduler, after storing the return address in index word IOCSIXG.

Block 054: The file and channel schedulers will determine if another area is available for processing. If none is available, a tape operation will be started to make the required area available, and the program will wait in the schedulers until it becomes so. Control will then return to the macro.

Block 055: GET or PUT a file. When at least one data record in an area remains unprocessed or after another area has been scheduled (by the schedulers) an index word is loaded with the address of the RDW for the area, for purposes of relative addressing, and control continues in the user's main program.

Block 056: GET a file to an area or PUT a file (or area) in a file. When at least one data record has not been processed in an area, or after another area has been scheduled, a data record is moved to the area designated in the operand of the macro-instruction.

Block 057: An index word is loaded with the address of the current RDW and control passes to the user's main program.

Block 058: PUTX. When at least one data record has not been processed in an area, or after another area has been scheduled, an index word is loaded with the address of the current RDW.

Block 059: The addresses of the current RDW's of both files referred to in the operand of the PUTX macro-instruction are located in index words. The contents of these index words are exchanged and control passes back to the user's main program.

Channel and File Schedulers

The channel scheduler is entered from one of the file schedulers except for the return from SPOOL. A file scheduler can be entered in either the priority or normal mode of operations. Figure 4 is a flow chart showing the channel and file scheduler operation. The completion of a tape operation signals for priority and enters at block 075. In the normal mode, when an IOCS macro (i.e., PUT or GET) refers to a block that has just been completely used, the file scheduler is entered at block 100 to obtain a new block (area) and initiate a tape operation to read into or write the area just completed.

A file scheduler is generated for each DTF by the Autocoder program during assembly. The format of these schedulers can vary, depending upon multiple area, I-O, and the form of the records being processed. The smallest file scheduler, for instance, would be for form 1 or 2 records with one area. The channel schedulers, however, are always the same size, since the only variable is the branch to the "high priority" pend-

ing switch. An address is inserted by the OPEN routine into the force-switch off exit.

In operation, each file scheduler has two switches and two counters. The switches are program switches (either a NOP or a B) with the counters in digit position 5 of the switches. These counters and switches are:

Availability Counter: This counter indicates the number of areas containing data records that are ready for use. For input files, it contains the number of areas that have been filled with data records; for output files, it contains the number of areas that have been written and are now able to receive processed records.

Availability Switch: This switch is turned on whenever at least one area is available. For an input area to be available, it must have been filled by a tape read operation and the condition code checked for a value of 2 through 7. An output area is available following a tape write operation and a check for error-free conditions after the write operation. It is turned off when the availability counter becomes zero, i.e., no area is available.

Pending Counter: This counter is incremented by 1, each time an area becomes ready for a tape operation. On input files, it is the number of areas that have been processed and are waiting to be read into; for output files, it is the number of areas that have been filled with processed records and have yet to be written on tape. This counter is decremented by 1, each time a tape operation for that particular file is started.

Pending Switch: This switch is turned on whenever at least one area for a file requires a tape operation, and is turned off when the pending counter becomes zero, i.e., when no area requires tape input or output operation.

Normal Mode

When an IOCS macro refers to an area that has already been processed (i.e., the last record in a block of data records has been "used" by a macro), the file scheduler is entered in the normal mode. The following description refers to the channel and file scheduler flow chart, Figure 4, for the normal mode.

Block 100: The sign of the last RDW is changed to minus for the tape operation and the area is put on pending by turning on the pending switch and incrementing the pending counter for this file. Control is then given to the channel scheduler (ISTART in comments), block 101.

Block 101: The cross switch (block 104) is turned on. This switch allows one check to be made of the availability switch (block 093) in the file scheduler before forcing an area to be made available by starting a tape operation for that file.

Block 102: If the free switch is on, indicating that the channel is free, a tape operation is started for the block of data records that was just completed and caused entry into the schedulers. If the switch is off (channel busy), control passes to the cross switch.

Block 103: Priority is masked to prohibit an interrupt and the bypass switch is turned on. Control is passed to block 080. Blocks 080 through 085, except block 084, can be executed in either the normal or priority modes of operation. The bypass switch is turned on and off only in the normal mode, thereby differentiating between the modes when necessary. See block 080 under "Priority Mode."

Block 104: The cross switch will always be on the first time it is tested and off the second time it is tested. In this manner, a check is made to determine if another area is available for processing following the first test of the cross switch. If an area is not immediately available the cross switch is tested for the second time, and found off. This passes control to the force routine starting at block 105.

Block 105: If the cross switch (block 104) is tested the second time, control passes to this block. The address of the pending switch for the file containing unavailable areas is set in the force switch (block 078).

Block 106: The force switch is turned on, making this file the next to be scheduled for a tape operation on this channel after an interrupt.

Block 092: The program will leave this waiting loop only after a priority interrupt has made an area available for the file desired. When the availability switch is turned on, and a priority release is given, control passes to the availability switch in the file scheduler (block 093). The availability switch can be either on or modified for an output file.

Block 091: The cross switch is turned off before the first check of the availability switch so that following the initial check of this switch, if it is off, a force routine will be set up starting at block 105.

Block 093: The availability switch can be on, off, or modified. If it is on, indicating an area is available, control is passed to block 094. If it is off, indicating that no area is available, control will return to the free switch (block 102). If the file is an output file, it could have been modified between blocks 075 and 076 in the condition code routine, in which case it will be a branch to the output condition code routine at IOCSIPSLO.

Block 094: An index word is set with the control RDW for the new area. A file scheduler has an availability subswitch for each area used by that file. These subswitches are set to refer to the next area that is (or will be) available. An area was found available and will be used immediately in processing, so the availability counter is decremented.

Block 095: The sign of the last RDW is changed to plus and the availability switch is turned off if the availability counter is zero.

Block 096: If the file is an input file and an unusual condition occurred on the last tape read for this file, control passes to the input condition code routine. Under normal circumstances control returns to the macro that caused entry into the schedulers.

Priority Mode

When the completion of a tape operation signals a priority interrupt, a branch to the file scheduler (block 075) for that file occurs. Zero priority branch to block 075 is discussed under "OPEN Routine."

Block 075: If the tape operation just completed was a correct length record as indicated by the condition code in the final status word, control passes to the channel scheduler. If the condition code was anything other than a CLR, (condition code 2), a branch to the condition code routine occurs. The condition code routine (IOCSIRTAIN) determines if an error occurred during the tape operation. If an error occurred (condition code 0-1), control passes to the error routine. If no error occurred (condition code 3-7) it is determined if the file is used for input or output. If it is an output file, the availability switch for this file scheduler is modified to branch to the output condition code routine (at IOCSIPSLO). If it is an input file, block 096 is modified to pass control to the input condition code routine (at IOCSICHECK).

Block 076: Upon entering the channel scheduler, the contents of accumulator 1 are temporarily stored so that the scheduler may use it.

Block 077: The SPOOL switch is an electronic switch (ES29 for channel 2 and ES30 for channel 1) that is set on or off by the SPOOL program. If it is on, control is given to the SPOOL program and returned to one instruction beyond the SPOOL switch.

Block 078: The force switch is on if it has been set by the channel scheduler in the normal mode. If it is on, processing is suspended by a waiting loop at the test availability switch (block 092). The delay allows a data area to be made available to process a particular file. This switch then branches to the pending switch of the file that caused the main program to wait (an area is "forced" to be made available for processing). If this switch is off, the pending switches are tested, highest priority first. ("Priority" refers to the priority designated in the DTF's for the tape files, not to the priority interrupt system of the IBM 7070.)

Block 079: If the force switch is on, it is turned off and control passes to the "forced" pending switch.

Block 080: If a pending switch is on, a tape operation is waiting to be started. A pending switch can be entered in three ways. If the force switch was on, this pending switch is on and will be entered from the force switch. If the force switch was found off, control would go to the highest priority pending switch. The pending switches will be tested and the file for the first one found on is processed. If the file scheduler was entered in the normal mode and the free switch in the channel scheduler was found to be on (free), control is transferred (via block 103) to this pending switch and it will be on. If the pending switch is off, a branch to the next lower pending switch (on this channel) occurs or if this is the lowest priority pending switch, control passes to the channel scheduler (at block 084).

Block 081: The tape operation for the area on pending is started. Each area used by a multiple-area file has a pending subswitch. These subswitches cause the flip-flopping through the different areas. They are always in phase with the availability subswitches.

Block 082: An input or output operation has been started for this file, so the pending counter is decremented and the pending switch for this file is turned off if the pending counter is zero.

Block 083: The free switch is set to busy (off) because an I-O operation has been started and control passes to block 085.

Block 084: If all pending switches are found off, the free switch is set to free (on) and control passes to block 085.

Block 085: The bypass switch can be turned on only in the normal mode. Both the normal and priority modes have been using the same logic (except for block 084) from block 080 to block 085. This switch is necessary to differentiate between the modes, since the logic is different for each mode from this point. In the normal mode, the bypass switch would be turned on by block 103. If the switch is off, the program is in the priority mode and control branches to the file scheduler (block 086).

Block 086: The entry into the schedulers in the priority mode of operation immediately checked the condition code of the last tape operation for this file. After checking for unusual conditions, that area is made available for this file by incrementing the availability counter and turning the availability switch on. If the program in the normal mode is in the waiting loop around block 092 (the availability switch test), only the action of the file scheduler in the priority mode (at block 086) will ever allow it to leave this loop. Blocks 086 and 087 represent the same file as block 075, since block 075 set index word IOCSIXF and this index word provides linkage between blocks 085 and 086. Blocks 080, 081, and 082, however, could represent any file on the channel.

Block 087: The BLOCKCNT field for the trailer label, stored in the DTF, is incremented by one.

Block 088: This block represents a general routine that all the channel schedulers use to restore accumulator 1 and release priority.

Block 089: The bypass switch was on, indicating operation in the normal mode. It is turned off and priority is masked to allow priority interrupt.

Block 090: The cross switch is tested. If it is on, the free switch was on upon first entering the channel scheduler and as a result the availability switch was not tested in the file scheduler (block 093). A check is made of the availability switch to decide whether to set up the force routine (blocks 105 and 106) and if so, then wait at block 092 for a data area to be made available by a priority interrupt. If the cross switch is off, the availability switch in the file scheduler was tested once and now a data area is forced to be available (starting at block 105). The program will wait at block 092 while an area is forced.

Block 091: The cross switch is turned off so that an area will be forced if it is not available immediately (i.e., as determined at block 093).

RLSE Macro (Release)

The RLSE macro causes a new area to be used by the main program. By using this macro, input data records may be deleted and the remaining data records in a block will not be processed following execution of the RLSE macro. On output files the RLSE causes a short length block to be written if a block is only partially filled with data records at the time the RLSE is executed.

The generated RLSE macro is slightly longer for output files than for input files. It utilizes the file and channel schedulers by first modifying the necessary linkages and exits of these schedulers. These exits and linkages are restored when the RLSE leaves the schedulers.

Basically, the function of a RLSE is to put an area, either input or output, on pending, and if the channel is not busy, to write out or read a new record into the "released" area. The logic flow within the schedulers is not changed and the blocks function identically in Figures 4 and 5. However, only one decision block will operate in the schedulers when used by RLSE. On performing the RLSE macro, the decision blocks in the schedulers are set in the direction necessary for correct operation of the RLSE when those blocks are reached.

This section describes a file and channel scheduler as they would operate in conjunction with a RLSE macro, as shown in Figure 5.

Block 097: The RLSE macro is entered in the user's program.

Block 098: The control RDW work address is made equal to the stop address +1 for an input file. For an output file the control RDW stop address is made equal to the working address +1. This accounts for the primary difference in the relative lengths of the RLSE macros generated for input and output files.

Block 099: Index word IOCSIXG is usually the linkage between the file and channel schedulers in the normal mode. The RLSE requires IOCSIXG, however, as linkage between the RLSE and the channel scheduler. Consequently, the linkage from file scheduler to channel scheduler (block 100 to block 101) is modified so that IOCSIXG is not used. Control then passes to the file scheduler at block 100.

Block 100: IOCSSENTXX. The sign of the last RDW is changed to minus. For an output file this probably will not be the last RDW of the block, since the stop address was made equal to the working address +1, but for an input file, it will be the last RDW of the block. The area is put on pending by incrementing the pending counter and setting the pending switch on. Control is then passed to the channel scheduler without setting IOCSIXG with a return address.

Block 101: The cross switch is turned on, but under no conditions will it ever determine whether to branch to set up the force switch. It will only be tested once by the channel scheduler modified for a RLSE and it will be on because of this block.

Block 102: The free switch provides the only decision in the modified schedulers. It determines if the channel is free or busy.

Block 104: The channel is busy, so the tape operation for the area released cannot be started at this time. The cross switch is tested and is found on.

Block 091: The cross switch is turned off and control is returned to the RLSE macro (block 107).

Block 103: The channel is not busy, so the tape operation for the released area can be started. Priority is masked to prohibit and the bypass switch turned on. Control passes to the pending switch for the file referred to in the operand of the RLSE macro-instruction.

Block 080: The pending switch is on, having been turned on in block 100.

Block 081: The tape operation for the area released is started and a pending subswitch is set to refer to another area for the next tape operation.

Block 082: The pending counter is decremented and the pending switch is turned off if the counter is zero.

Block 083: Since a tape operation has just been initiated, the free switch in the channel scheduler is set to busy (off).

Block 085: The bypass switch is on, indicating the schedulers are being used in the normal mode. (See block 089.)

Block 089: The bypass switch is turned off and priority masked to allow priority interrupt.

Block 090: The cross switch is tested and is found on. It was turned on in block 101 and never tested after that. Since it is only turned off after having been tested, it must be on.

Block 091: The cross switch is turned off and control is passed back to the generated RLSE macro (block 107).

Block 107: Those linkages modified by the RLSE macro before entering the schedulers are restored.

Block 108: The file scheduler is further modified before leaving the RLSE macro. The next GET or PUT following the execution of this RLSE macro has no data record ready for processing because the control RDW working and stop addresses were adjusted to simulate end of block (block 098). When the next GET or PUT enters the file scheduler, the RLSE macro will have already put the area on pending (and perhaps started the tape operation for the area). Therefore, block 100 must be modified so that the released area does not increment the pending counter twice. The RLSE macro makes this modification in block 108, by bypassing those instructions which increment the pending counter and turn on the pending switch. For output files only, the RLSE macro also changes the sign of the last RDW used to plus (since this RDW would not be the last RDW of the area if a short-length block was written out). These one (or two) modifications set up their own restoration of the file scheduler.

Condition Code Routine

A final status word is automatically created at the conclusion of each tape read or write operation. Position 1 of the final status word will contain a digit indicating the condition of the last tape operation at the time of the priority signal (interrupt). The condition code routine is used whenever the file scheduler involved senses that some condition other than correct length record (condition code 2) occurred on the preceding tape operation. Provision is made so that the user can add routines to handle the special conditions that may occur.

A detailed description of the processing in the condition code flow charts, Figures 6a and 6b, follows.

Block 125, IOCSIRTAIN: The contents of accumulator 1 are saved so that accumulator 1 can be used by the condition code routine.

Block 126: If the condition code is less than 2, a branch is made to the error routine, connector CA.

Block 127: The digit in position 1 of the initial status word is tested to determine if the operation was input or output.

Block 128: The pending switch for this file is turned off by making its sign plus.

Block 129: The file scheduler is modified by setting a BLX to IOCSICHECK into location IOCSEXITXX. IOCSICHECK is the entry point for condition codes 3 through 7 when the unusual condition occurred, for an input operation.

Block 130: Index word 99 and accumulator 1 are restored and a return branch is made to the file scheduler.

Block 131: If block 127 found that the unusual condition occurred on an output operation, the availability switch in the scheduler is modified to contain the address of IOCSIPSLO. This will allow the scheduler to later branch to this entry point to check condition codes 3 through 7.

Block 132, IOCSIPSLO: Is the pending switch on? If it is on, some output areas are filled and unwritten, so the file scheduler is re-entered to cause this file to be forced. The program will continue to return to the file scheduler as long as any areas remain pending.

Block 133: The free switch is tested to determine if the channel is busy. If it is busy, a holding loop is entered until the channel becomes free.

Block 134, 134-1: The availability switch is repaired (branch to itself +5). If a CLOSE operation, the CLOSE routine is re-entered.

Block 135: Prohibit any interrupt. This is done in case there should be a SPOOL program or a tape operation that might demand priority while the special condition is being processed by some other routine (user's routine) outside of IOCS.

Block 136: The routine is modified for this particular file. IOCSIXF is set with the DTF address. Tape channel and unit are set into the condition code message, and a return to the file scheduler is set in (6, 9) of IOCSIXG.

Blocks 137, 138, 139, 140, 141: The condition code (position 1 of the final status word) is compared to the digit 4 in block 137 and 6 in block 139. Branches to the proper routines are made depending on whether the comparison was low or equal. If no branch occurs, it is a SCLR condition.

Block 142, IOCSICHECK: Entry is made at this block for condition codes 3 through 7 for an input file. Is the availability counter zero? If not, it signifies that other areas have input records to be processed before processing the block that caused the unusual condition code. Return is made to the file scheduler to process all records up to the input area having the 3-7 condition code. Example: When EOF (condition code 5) occurred, all data records must be processed before entering the EOR routine.

Block 143: Restore the file scheduler exit (Figure 4, block 096) and proceed to block 135 to prohibit priority as when entry is made from block 134.

Block 145, IOCSLLR: The DTF entry LLRPROCD is brought to accumulator 1.

Block 146: This block, on recognition of the checkpoint record from a shared output-checkpoint tape, branches to block 158 via connector DG.

Block 147: A BLX to IOCSPROC is executed. IOCSPROC is the start of a routine that is used by all conditions except EOF. (See blocks 148 through 151 of condition code flow chart, Figure 6b.)

Block 148: The address of the user's routine, if one is supplied, is placed in the exit instruction address. If the user does not supply a special routine, this address will be either 9999 or 0000.

Block 149: Accumulator 1 is compared to 9999 and, if they are equal, the program skips any further processing of the condition code and returns to the file scheduler.

Block 150: If an unequal compare occurred in block 149, accumulator 1 is tested for zero.

Block 151: If accumulator 1 was zero in block 150, a message stating the condition code found is typed, and a return branch is made to the file scheduler. For LLR, SCLR, and SLR, IOCSIXG is modified so that a return is made to the program location of the GET macro at the BLX plus 1. For EOS, IOCSIXG is modified so that a return is made to the GET macro at the BLX.

Block 152: If accumulator 1 was not zero in block 150, a BLX is executed on IOCSIXG to the address of the user's routine. The user must provide a branch to 0+IOCSIXG in order to return to this routine and then back to the file scheduler.

Block 153, IOCSSCLR: The DTF entry (SCLRPROCD) is placed into accumulator 1. This will be the beginning address of the user's routine, if one is supplied. Otherwise, this entry must be either 9999 or 0000.

Block 154: A BLX on IOCSIXG to IOCSPROC is executed. (Refer to blocks 148 through 151.)

Block 155, IOCSSLR: Modify the routine with addresses of IXA and IXB (index words) for this file. Modify the file scheduler at location IOCSSENTXX by setting it to "BLX on IOCSIXG to IOCSSLR+20." This will cause a return to the SLR routine to restore it where needed after the condition sensed has been taken care of, either by the user's routine, by ignoring it, or typing the SLR message and ignoring it.

Block 156: IXA is loaded with the first RDW of the area containing the short length record, and the non-indexing portion of IXB is set to the address of the last RDW used when this condition occurred. The program then executes a BLX to IOCSPROC. (See blocks 148 through 151. IOCSIXG has been modified to allow a return to the next instruction in the object program.)

Block 157: This block starts at IOCSSLR+20 and is entered from the file scheduler by a BLX on IOCSIXG at location IOCSSENTXX. The index word holding the RDW list addresses is repaired by restoring the non-indexing portion to normal. A branch is then made back to the file scheduler at IOCSSENTXX.

Block 158, IOCSEOS: The block count is reduced by one and the IOCSEOSSW (end of segment switch) is set on. The routine is modified to use IXB for this file and IXB is counted up to make it appear as though a normal block had been read in from tape.

Block 159: The non-indexing portion of IOCSIXG is modified so that the exit from IOCSPROC returns to the EOS routine (block 160), instead of to the file scheduler.

Block 160: IOCSIXG is loaded with the return address to the file scheduler. The file scheduler entry is restored and IXB is counted up to make it appear as if a normal block had been read from tape. Priority interrupt is allowed, and a return branch to the file scheduler (IOCSSENTXX) is executed.

Error Routine

IOCS contains an error routine which is used to detect and correct tape errors. Different error correction routines are used for input and output files. When an error occurs in a tape read or write operation, position 1 of the final status word is set with a condition code indicating the condition found at the end of the tape operation. When an error occurs on reading an input file, an error correction routine repeats the reading operation up to nine times or until the record has been read correctly. If the condition code still indicates an error occurred, the record is handled according to TPERROPT, in the DTF. TPERROPT may be 60, 50, 10 through 49, or 00. If it is 60, the error routine keeps trying to reread the tape until it is read correctly. If the error option is 50, the error block is typed and the block of data records is searched. The invalid words and their locations are then typed and the machine stops to allow the operator to correct these errors manually. If the error option is a number 10 through 49, the block of data records is searched, any invalid words are replaced with asterisks, and the block is written (dumped) onto the tape specified by the option code. If the option is 00, the machine types invalid words and halts. This allows the operator to correct any errors and resume the operation.

If an error occurs in writing an output file, backspacing and error-free rewriting is attempted twice. If an error is still indicated, a tape skip is included in the rewrite routine to try to skip over bad spots on tape. If an error still occurs after 25 attempts to write,

an error message is typed and the data record block is scanned for validity. If an invalid word is found, the machine will stop, since the disable switch is not on. The operator can then correct the word in error manually, or take whatever action is deemed necessary.

The error routine operation is shown in Figures 7a and 7b.

Block 175, IOCSERROR: Accumulator 2, index words 97, 98, and IOCSIXG are saved and the error routine is modified for this particular file. The last word of the last RDW for the error record is located.

Block 176, IOCSERRRT: The error is counted and a test is made for input or output operation.

Block 177, IOCSEOUTPT: The tape skip field of the DTF is used as an output error counter (tape write errors). The program tests this field to determine if thirty write errors were made on this file.

Block 178: If it is determined that thirty write errors have occurred, a statistics message is typed and the error counters for this file are reset to zero.

Block 179: The tape for this file is backspaced.

Block 180: IOCSRETRY+1 will be "branch to itself +3" if input, and NOP after the first retry of an output file. This allows the first retry to be written on the same portion of tape as the original record.

Block 181: The tape is skipped and the skip counter is incremented by 1.

Block 182, IOCSETPOP: The tape operation (read for input or write for output) is retried.

Block 183: This block is a switch that prevents error checking on the first two read operations of the tape cleaner routine (explanation at block 198). It is made a branch by block 198 and a NOP by block 200.

Block 184: The condition code is tested in the tape final status word. If it is a correct length record, a branch is made to IOCSERENTR to restore the registers and return to the tape priority branch address. If not CLR, retries can be made with the BIX loop at block 187 for output or block 195 for input.

Block 185: This block represents the exit from the retry routine. It consists of a branch to 0+IOCSIXF and allows a return to the proper routine. This is necessary as the retry routine is used for input retry, output retry and the input cleaner routine. If noise was encountered, a return is made via connector CD to block 176.

Block 186: This block is entered after the first output retry and sets block 180 to a NOP to allow skipping tape before retrying the write operations.

Block 187: This block represents a BIX that allows 24 additional retries before testing the write record in storage for invalid characters. IOCSIXF is left as it was set by BLX1, so a return from the retry routine is to connector CH.

Block 188: After leaving the BIX loop, the limits of the area to be searched are set into IOCSIXF.

Block 189: A search loop is entered to zero and add each word of the block in core storage. If a word containing invalid digits is brought to the accumulator, the validity check light (IB) will turn on and the machine will halt at this location. (IOCSOUTRCT+4.)

Block 190: A halt and branch occurs before retrying the write operation up to 25 more times. The operator can hit START and continue trying to write, or take appropriate action for the condition.

Block 191: It was input operation. Do we now have 50 errors?

Block 192: Fifty errors have occurred. Type the error statistics message and reset the error counters for this file.

Block 193: Was it a noise record? Check if three words were read. If less than three words were read into memory, the record is considered noise.

Block 194: Count the noise record and BLX on IOCSIXF to IOCSETPOP to read the next record. After completion of the read operation in the retry routine, if an error occurred return is made, via IOCSIXF, by a branch to IOCSERRRT (connector CD).

Block 195: This block represents a BIX that allows eight additional retries before taking more elaborate steps to read a good record. IOCSIXF is left unchanged from BLX2 to allow return to this block for the eight retries.

Block 196: If any noise was recognized, blocks 197 through 200 are bypassed by a branch to block 201.

Block 197: The block count is tested. If less than 3, the program branches to block 201, since the tape is too close to load point to use the cleaner routine.

Block 198, IOCSCLEAN: This block represents the start of the cleaner routine, which performs three backspace operations and then reads forward three times. Error checking occurs only on the last read operation. This series of operations passes the tape over the tape cleaner. If the error was caused by a particle of oxide on the tape, the cleaner may remove it and allow correct reading. This block (198) represents a backspace loop that reads backward over three records. The bypass around the error test is formed by making block 183 a B. A BLX is then made to IOCSETPOP to read the first record. The return after the read operation will be to block 199.

Block 199: A BLX is made to IOCSETPOP to read the second record without error checking it. After completion of that operation return is made via IOCSIXF to block 200.

Block 200: Block 183 is reset to a NOP to allow error checking, and a BLX is made to IOCSETPOP to read and error-test the third record. If an error still occurs, a return will be made, via IOCSIXF, to block 201.

Block 201: Type the read failure message (TP WD ERR) and then check for option 60.

Block 202: TPERROPT of the DTF determines the error correction procedure. If 60, the program will branch back to IOCSIRETRY and continue to branch there until either the error is eliminated or the operator intervenes. If the cleaner routine corrects the error, the control will pass to the tape priority branch address.

Blocks 205, 206: If the procedure is not option 60, the machine types a disable switch message and stops for the operator to turn the error disable switch on and press START. This is changed to a HB after the errors have been corrected and then the program branches from block 215 to block 205.

Block 207: If the TPERROPT of the DTF for this file contains 50 or 00, the machine will type the location of all the error words in this block and stop so that they can be corrected manually. If TPERROPT contains a number 10 through 49, the program will scan the record area and locate any words containing invalid characters. It will set the signs of these words to alpha, and the contents of each invalid word are changed to five asterisks. After all of the invalid words have been changed to asterisks, the block of records will be written out on a "dump" tape as designated by TPERROPT in the DTF for the file. This option is tested here.

Block 208: If the TPERROPT of the DTF is a number 10 through 49, the dump routine is initialized with the dump tape channel and unit.

Block 209, IOCSESTSR: The index words for the area to be searched are set into the invalid word search routine.

Block 210: A ZA1 is made of the first word of the record from the search area and the error latch (IB) is tested using operation code +09 with field definition of 32. If the latch is on, indicating that the word is invalid, a branch is made to IOCSINVALD. If the latch is off, a loop is made back to zero-and-add the next word of the record area in core storage.

Block 211, IOCSINVALD: This block puts the invalid word of accumulator 1 into the message area and resets the error latch. IOCSDUMPSW will be on to branch to IOCSEDUMP for a dump option (10-49), where the sign of the error word is changed to alpha and the contents set to five asterisks.

Block 212: The error location message is typed and the invalid search continues. IOCSERSW is set to NOP (yes).

Block 213: The sign of the invalid word is set to alpha and the contents are set to five asterisks.

Block 214, IOCSECONT: A BIX loop tests to determine if the invalid search routine is finished.

Block 215: If errors were found, IOCSERSW is yes (NOP), and a halt and branch occurs (block 216) to search the area again. If no errors were found, IOCSDISABL+2 is made plus (HB) and a branch to block 205 occurs.

Block 216: A halt and branch is performed to allow the operator to correct the errors manually and, after START, a return is made to IOCSESTSR to retest the record for proper correction.

Block 217: For a dump option (10-49), IOCSDUMPSW sign is plus or yes.

Block 218: Write the block of records on the specified dump tape, increment the block count, and go to IOCSETPOP to get the next record.

Blocks 219, 220: If it was not an error dump, calculate the condition code, store it in position 1 of the final status word, restore those registers saved by block 175, and branch to the tape priority (interrupt) branch address.

End of Reel Routine

The function of the end-of-reel routine is to process all header and trailer labels and to write end-of-file records (tape marks). It will read and check all header and trailer labels written on input tapes (if the user specifies), read and check header labels written on output tapes (if specified), and write new header and trailer labels on output tapes (when specified). Label processing is important. For example, it can prevent the destruction of valuable information stored on tapes. Also, information in the labels can be checked to make certain that all tape records have been read. A description of the EOR routine follows, with reference to the EOR flow charts (Figures 8a, 8b, and 8c).

Block 225: The DTF address of the file being processed is set into the indexing position of index word 97. The non-indexing position of word 97 is set to 0000 for an input file or 9999 for an output file. This information will be used to obtain specifications from the DTF's and act as a switch to differentiate between input or output files in the EOR procedure. This block also sets up the user's exits for special EOR procedures by determining whether these exits (NOP) will be set to branches. This is done by checking the exit mask in the label DC entry. The user, on completion of his routine, after using any exit may return to the EOR routine with a branch to 0+IOCSIXF.

Block 226, IOCSOPNSW1: A branch around trailer operations and between-reel functions will be made to block 244 if the EOR routine was entered from the OPEN routine to process a header label.

Block 227: This block starts the processing of a trailer. It turns off the EOF indicator and determines if the file is an input or output file. If it is output, a branch is made to block 233.

Block 228: The block counter is reduced by 1 to prevent the tape mark from being counted as a block. The DTF LABELINF is then tested. If it is 0000, indicating no label, a BZ1 is made to block 232.

Block 229: This block reads the label, types label statistics, and compares the block count in the DTF to that in the trailer. If they are equal, a branch is made to block 231.

Block 230: A message indicating an incorrect block count is made followed by a halt.

Block 231, IOCSEX6: This block is an exit available to the user so that he may do additional checking of the trailer.

Block 232, IOCSEOFEX: The trailer record is tested for EOF. If EOF is indicated (label word 2 contains EOF), a branch is made to the user's EOF routine at the address he has specified in EOFPROCD of the DTF. If no labels are specified, this exit is used for all end-of-reel conditions for input files.

Block 233, IOCSLBTM: This block starts output trailer procedures by writing a tape mark, typing trailer statistics, and checking for a trailer.

Block 234, IOCSEX1: This block completes the trailer record and provides an exit for the user to add additional information to the trailer if he wishes.

Block 235: The trailer is written on the tape.

Block 236, IOCSEX2: This block provides an exit to allow the user to write additional trailers.

Block 237: A tape mark is written after the trailer or trailers.

Block 238, IOCSICEST: An exit to the CLOSE routine is made if the EOR routine was entered on a CLOSE operation.

Block 240: This block is entered only as a result of an end-of-reel condition. It will restore the DTF's BLOCKCNT field to zero and rewind the tape as directed by the DTF's RWDPROCD digit.

Block 241: The tape unit addresses are rotated in BASETAPE, ALT1TAPE, and ALT2TAPE, of the DTF for the file.

Block 242, IOCSEOREX: This block provides an exit for the user to alter the tape addresses in the DTF if he desires other than normal operation, such as using two files for three tape units. This exit is always made for EOR with unlabeled output files.

Block 243: The appropriate message concerning the tape address changes is typed and a halt is made if a manual tape change is necessary.

Block 244: The proper channel and tape addresses for this file are placed in the remaining EOR instructions.

Block 245, IOCSERSEL: The tape is tested for ready and a branch is made to block 246 if not ready.

Block 246: The not-ready message is typed and a return is made to block 245, where the machine will hang until the tape is readied.

Block 247, IOCSOPNSW3: This block is a switch that returns control to the OPEN routine (during an OPEN operation) to rewind the tape according to OPENPROC in the DTF. After this operation, a return is made to the EOR routine to block 249.

Block 248, IOCSRWNRL: The tape is rewound as specified by the normal EOR specification code located in RWDPROCD of the DTF.

Block 249: The header density is set as specified by the TDENSITY in the DTF for the file.

Block 255: A BCX determines if labels are to be processed and, if not, a branch is made to block 270.

Block 256: The file is tested for input or output, and, if output, a branch is made to block 262.

Block 257: The header label for the input file is read.

Block 258, IOCSCKLOOP: The label mask is tested and the header is checked against specifications.

Block 259: The label is tested for error and, if in error, control passes to block 260.

Block 260: This block types the error message and halts.

Block 261, IOCSEX7: A user's exit is provided so that additional input header information may be processed.

Block 262, IOCSLOPRO: The header of this output file is read.

Block 263: The retention cycle is tested to determine if this tape may be written on. If the current date stored in word 109 is within the retention cycle, a branch is made to block 264.

Block 264: The error message is typed to direct the operator to check the retention date, and a halt is made.

Block 265, IOCSEX3: This block provides an exit for the user to perform any other header checks he may want.

Block 266: The label mask is tested, the tape is backspaced, and a new label is set up if specified.

Block 267, IOCSEX4: This block provides an exit for the user to add label information if he wishes.

Block 268: The new label is written and the label statistics are typed.

Block 269: A user's exit is provided to write other labels if he desires.

Block 270, IOCSFLDN: The file density is set according to the control digit of the file's TDENSITY in the DTF.

Block 271: This block represents a NOP operation that is initialized by the initialization and assignment sec-

tion of the checkpoint routine. The instruction is altered into a branch to IOCSCPEOR where a decision to take a checkpoint is made.

Block 272, IOCSOPNSW2: A return to the OPEN routine is made if the EOR routine was entered during an OPEN operation.

Block 273: A test is made to determine if the file is input or output to enable a return to the proper place in the condition code routine.

Open Routine

An OPEN operation activates those files named in the operand of the OPEN macro-instruction, by supplying the schedulers with the addresses necessary for the running of the program. Following the BLX in the macro, one branch containing the address of the DTF is generated for each file. A NOP immediately follows the last branch instruction.

Another important function of the OPEN routine is to supply priority branch addresses for the file schedulers. These addresses are determined by the PRIORITY field in the DTF. All files with a PRIORITY of zero use the common priority branch address at word 150. This address will branch to IOCSPZCOM where a routine determines the specific file scheduler to be used. (For more detail see block 281.) The manner of initialization of files depends on FILEFORM and the number of areas for that file.

The OPEN flow charts are shown in Figures 9a and 9b.

Block 275: This block recognizes the NOP after the last DTF address listed in the macro, and branches to block 278 with a BXM. Blocks 275, 276, and 277 form a loop to process each DTF in sequence, placing a 3 in the ACTIVITY of the DTF for each file specified in the OPEN macro.

Block 276: A 3 from accumulator 1 is placed into the ACTIVITY field for all files to be opened.

Block 277: This block advances the operation to the location of the next DTF address in the OPEN macro. This is the address of the next branch instruction.

Block 278: The NOP instruction after the last word of the final DTF in storage is recognized by a BM1. This indicates that the scheduling operation for all files to be opened have been initialized except for alignment of the pending switch addresses (priority). The BM1 is made to block 290 (IOCSISORT) to sort the pending switches in the order of their priority for each channel.

Block 279: The ACTIVITY code of the first and succeeding DTF's in storage is compared to 3. If the comparison is low, the file is not to be opened, and a branch is made to block 280.

Block 280: The operating address of block 279 is stepped by 9 (number of words per DTF) to advance the operation to the next DTF in storage.

Block 281: This block sets the file scheduler address into the priority branch address, to prepare the file for interrupt operation. The PRIORITY in the file's DTF will determine the priority branch address to be used; i.e., a file with a PRIORITY of 1 will use word 151. If the PRIORITY of an opened file is zero, the address of IOCSPZCOM (located in the condition code routine) is placed into word 150. This routine (IOCSPZCOM) compares positions 2-5 of the initial status word (originating tape operation address +1) to that of the pending and availability switch. This is done because the tape operation is located, in the scheduler, between the pending and availability switches. When the initial status word address is higher than the pending switch address and lower than the availability switch address, a branch is executed to the DRDW address +1 for this file. This address will be the location for the condition code test for this file in its file scheduler.

Block 282: This operation is located in the EOR routine and sets index word 97 with the address of the file's DTF and, by checking the label masks, sets up the user's exits to enable a branch into "other than normal" label operations.

Block 283: A branch is made back to the OPEN routine to perform rewind procedures according to the DTF's OPENPROC rather than the RDWPROCD used for the EOR routine.

Block 284: This block shows a return to the EOR routine for header label processing (explained in EOR routine). After label procedures are completed a branch is made back to the OPEN routine.

Block 285, IOCSOPEOR: The error fields in the DTF are zeroed and all necessary channel scheduler addresses are placed into the file scheduler for this file. To accommodate all files on a channel, returns to the file schedulers from the channel schedulers will be made via index words loaded by a BLX. Between an OPEN and the next CLOSE for a given file, the branches from the file scheduler to the channel scheduler will be relatively unchanged; therefore, these addresses may be placed into the file scheduler. They are:

Exit to ITESTS (in comments) which makes a BLX to the first instruction of the channel scheduler.
Branches to the bypass switch.
IOCSSENTXX, restored by dumping the previous identical instruction on it.
Exit to ISTART (in comments) which makes a BLX to the channel scheduler to turn the cross switch on.
Exit to IFORCE (in comments) which makes a BLX to the channel scheduler force switch.

Pending switch on instruction IOCSAVSWXX-2, which is restored.

Block 286: The pending and availability subswitches in the file scheduler are initialized and the tape instructions are set with their proper operation code and address. This block will also prepare the RDW's, INDXWRDA, and INDXWRDB, for operation during the first PUT or GET and read or write operation. These operations are:

Set the DTF's SRBFORM4 to 1 on form 1, 2 and 3 records. Calculate the number of sections per block on form 3 records. Zero the availability counter, turn on the pending switch, and set the pending counter to the number of areas minus 1 for input files. Zero the pending counter and set the availability counter to the number of areas for output files.

Check for single area processing; if so, bypass instructions dealing with subswitches (i.e., directly to pending-switch off instruction). If multiple-area processing is used:

1. Turn first pending and availability subswitch off.
2. Turn second availability subswitch off.
3. Turn second pending subswitch off for all files except form 3, input.
4. Turn second pending subswitch off for form 3, input files.
5. For *all* files, turn the pending switch off and the availability switch on.
6. Turn the availability switch back off for input files.

Set tape operation code and unit into the file scheduler's tape read or write instructions in all file schedulers, also the tape instruction address (RDW address) for form 1, 2 and 4 files, and form 3 input files.

Restore the file scheduler exit address, since it may have been altered if the last operation had been a condition code 3-7 operation. Make all RDW signs plus on form 1, 2, and 4 files. Make the last RDW of each area minus for input files.

Set up the area control and make the RDW minus for form 3 input files and set up the area control and write control for form 3 output files.

Set up INDXWRDA and INDXWRDB for input files to prepare them for the first GET. Set up INDXWRDA and INDXWRDB for output files. Make the file active and zero the BLOCKCNT for the file by setting 10000 into positions 5-9 of word 1 in the DTF.

Blocks 290 through 304: IOCSISORT sets up the order in which the pending switches of the file schedulers are tested on an interrupt when no file is forced. Its operation is described for each of blocks 290-304.

Block 290: The address of the second word of the first DTF is set into IOCSIXF with the address of the NOP after the last DTF. This index word holds the address of the DTF being operated on. The address of the last channel is set to make the force switch address and the bypass switch entry address available. The highest channel number, incremented by 1, is placed into accumulator 2.

Block 291, IOCSAD: The channel number in accumulator 2 is decreased by 1. Also a BZ2 is made to the instruction after the OPEN macro in the main program when accumulator 2 becomes zero. This branch is performed after the last channel has been completed.

Block 292: This block will subtract 35 (number of words per channel scheduler) from the channel operating address (address of the bypass entry) in X98, and zero the priority test in accumulator 1. For the first operation for each channel, the test number will be incremented to 1 (highest priority).

Block 293: IOCSIXG is reset to the start and finish addresses for stepping the DTF's, and the priority test (accumulator 1) is incremented by 1.

Block 294: The operation advances to the next DTF in storage by increasing IOCSIXG by nine (number of of words per DTF).

Block 295: The working DTF's PRIORITY is compared to that in accumulator 1. If they are equal, a BE is made to block 296.

Block 296, IOCSAC: The DTF's ACTIVITY is tested and, if inactive, a BL is made to block 301.

Block 297: The FCHANNEL in the DTF is tested to determine if this file is on the working channel (accumulator 2). If so, a BE is made to block 298.

Blocks 298, 299 and 300: These blocks can be processed only when files meet the proper specifications. They must be in the proper channel (block 297), and must have been activated (block 296). The sorting is done by block 295, allowing operation on the files in the order of their priority.

Block 298: This operation consists of three instructions. They are:

```
XL      99,9998+IOCSIXG
ZA3     0(2, 5)+X99
STD3    0(6, 9)+X97
```

These instructions place the pending switch address of this file into the location specified by the indexing portion of index word 97. The first time blocks 298, 299, and 300 are used for a given channel, index word 97 contains the address of the channel force-switch off address (entry to the pending switches on interrupt with no force operation). During later passes through blocks 298, 299 and 300, word 97 contains the address of the pending switch of the previous pass (next higher priority for the channel).

Block 299: The instruction STD3 97(2, 5) alters the indexing portion of the index word of the third instruc-

tion in block 298. This will change the force switch off address to that of the highest priority file's pending switch. Index word 97, when it is used next, then holds the address of the pending switch for the last file operated on (next higher priority) for this channel.

Block 300: This operation consists of two instructions. They are:

```
ZA3     98(2,5)
STD3    0(6,9)+X97
```

These instructions place the address of the channel bypass-switch entry into the pending switch of the file being handled. If a file of lower priority on this channel is handled next (by blocks 298, 299, and 300), the channel bypass-switch entry address will be overlaid by the pending switch address of the file of lower priority. If no other files are handled, the address of the channel bypass switch entry will remain in the lowest priority pending switch (last file handled), providing the linkage from the file schedulers back to the channel scheduler if no operations are pending.

Block 301: The DTF is tested to determine if it is the highest one in storage and, if not, a BCX is made to block 294 so that all DTF's may be tested for this priority.

Block 302: A BZ1 is made to block 291 if the priority test number equals zero by exit FB.

Block 303: The priority test number is compared to 9. If equal, a BE is made to block 304.

Block 304: The priority test number is set to minus 1 so that when incremented by block 293 it will become zero. This is the priority lower than 9. A branch is then made to block 293 to check for the DTF's of zero priority.

End and Close Routine

The END and CLOSE macros use the same routine with the exception of a few instructions. CLOSE performs deactivation procedures on those files that are named in the operand of the CLOSE macro-instruction. A sequence of machine instructions is generated containing a BLX to IOCSICLOSE, followed by one or more branch instructions, one for each file to be closed. The address portion of each branch instruction contains the location of the DTF entry containing the activity field for that file. A NOP immediately follows the last branch instruction.

END closes all files that have been previously opened. END has the option of returning to a waiting loop located at IOCSEJLOOP after completion, so that SPOOL operations may continue after end-of-job occurs.

This section describes the END and CLOSE operation as shown in the flow chart in Figure 10.

Block 325, IOCSIEND: The BLX in the END macro enters the routine at this point. The ACTIVITY code for OPEN (1) is placed into accumulator 1. The digit in the accumulator becomes the compare-digit (block 331) to determine if a file is to be closed.

Block 326, IOCSICLOSE: The ACTIVITY code (4) for files to be closed, as a result of a CLOSE macro, is placed into accumulator 1.

Block 327: The NOP, located after the last DTF entry listed in the CLOSE macro, is recognized by a BXM.

Block 328: A STD1 operation is used to store the 4 into the ACTIVITY in the DTF for the file listed in the CLOSE macro. The address of the DTF location in the macro is incremented by 1. This steps the operation to the next DTF location, or the NOP if the last DTF has been processed by this loop (blocks 326, 327, and 328).

Block 329: The character from accumulator 1 (placed there in block 325 or 326) is stored into the compare instruction (decision) of block 331.

Block 330: The NOP instruction after the last word of the final DTF in storage is recognized by a BM1. This indicates that the CLOSE or END operation is completed, and control returns to the main program.

Block 331: For an END macro, a 1 is compared with the activity code of each DTF. A branch is made on an equal comparison to CLOSE procedures. For a CLOSE macro, a 4 is compared with the activity code of each DTF. A branch is made on an equal comparison to CLOSE procedures.

Block 332: The operating address is stepped by 9 (number of words per DTF) to advance to test the next DTF in storage.

Block 333: The file being closed is tested for type (input or output). If it is output, a branch is made to the condition code routine (block 334).

Block 334, IOCSIPSLO: A loop is entered in the condition code routine to delay operation until output operations are completed for this file. The exit from the loop will be made after the pending switch has been turned off.

Block 335, IOCSCEBACK: A test is made to determine if an incomplete block remains for this file. If there is none, a branch is made to block 337.

Block 336: The short block is written on the output file.

Block 337, IOCSEOR: A branch is made to the end-of-reel routine to perform trailer operations for this file.

Block 338, IOCSIECOV: The file is deactivated by placing a zero in the ACTIVITY field of the DTF for this file.

Block 339: The tape is rewound, if specified by the DTF, and a branch is made to block 332 to operate on the next file.

Checkpoint and Restart Routines

The checkpoint and restart routines are beneficial for time-consuming programs. Their function is to make periodic records from which all machine conditions can be restored to their present status at a future time. This is advantageous if it should be necessary to stop the program before its completion. It also allows completion of the program if an error should occur, such as might be indicated by a block count or hash total discrepancy. Under these circumstances, by using the last good checkpoint, it is unnecessary to rerun the portions of the program which had been correctly processed. Many minutes or even hours of system time could be saved in this manner.

The checkpoint and restart routine consists of the following components:

CHPT1

This routine is produced on assembly, as a result of the DIOCS entry CHPT. It is composed of three sections:

1. Assignment and initialization (IOCSRSAV): This section prepares the checkpoint routine for its specified operation, prepares the checkpoint tape, and writes the restart routine on the checkpoint tape. It is then overlaid in storage by other routines.
2. Checkpoint routine (IOCSCHPT): This routine writes the checkpoint records.
3. Restart routine: This routine is read from the checkpoint tape during a restart. It checks the header labels on tapes, positions the tapes according to the block count in the appropriate DTF's, and restores the machine conditions and storage to what they were at the time the checkpoint was taken.

DCHPT

This is a data entry of one word and contains information which CHPT1 uses in its operations.

CHPT

This is a macro-instruction which is used by the programmer to take checkpoints when he desires.

RSTRT

This is a macro that initiates the restart procedure. It is brought into storage before a restart or remains in storage during the running of the program. Linkage is made to this routine to begin a restart. The routine will read the restart program from the checkpoint tape so that the restart may be made.

The format of the checkpoint record is:

Word 1 = bCHPT
Word 2 = +CULD000nnn
C = Channel for checkpoint tape
U = Tape unit for checkpoint tape
L = Label code
D = Density code
nnn = Checkpoint number
Word 3 = Latch and mode settings
RDW's of storage
DTF's
Storage

Figure 11 is the flow chart for checkpoint and restart routines.

Block 351, IOCSRSAV: The checkpoint and restart routines are modified at object time according to the specifications in the DCHPT entry.

Block 352: Label procedures are performed for the checkpoint. If the checkpoint tape is also an output tape (shared), the normal EOR routine is used for the label procedures. The OPEN exits in the EOR routine are modified so that returns can be made to the restart routine. The DTF of the shared file is used in the EOR routine in performing the label checking (retention cycle) and the new label is written. The DTF's OPENPROC and LABELINF are zeroed to prevent moving this tape if its file were to be opened. Control then passes to write the restart record (block 353). If the checkpoint tape has no output records on it, the assignment routine rewinds the tape and, if labels are specified, reads the label. A branch is made to the EOR routine to test the retention cycle. After the retention cycle test, a return is then made to the restart routine via IOCSEX3. (See Figure 8c.) The tape serial, file serial, and creation date of word 109 are set into the new checkpoint label, and the label is written on the checkpoint tape. Control then passes to block 353 to write out the restart program.

Block 353: The restart program is written on the checkpoint tape and a return is made to continue with the main program.

Block 355: This is the first of two machine instructions generated by the CHPT macro-instruction. It provides an entry to the checkpoint routine and prepares for the return by loading IOCSIXG with the instruction counter +1.

Block 356, IOCSCPEOR: This operation will be initialized by IOCSRSAV (block 351) according to the FREQ in the DCHPT entry. This entry (FREQ) determines if a checkpoint is to be taken before processing of input, output, or both input and output reels or whether it is taken only by a CHPT macro. These instructions also

prevent taking a checkpoint while processing labels during an OPEN.

Block 357, IOCSCHPT: The checkpoint operation is delayed by testing the free switches on both channels. Looping and testing occurs until both free switches are on insuring that all pending tape file operations have been completed.

Block 358: Accumulators 2 and 3 are saved by storing them in the block-count error message. They will be restored before returning to the main program.

Block 359, IOCSCPINT: An interrupt is forced by allowing priority, turning on the tape 10 latch, and waiting for it to cause an interrupt. This allows the latch and mode settings (word 100 after the interrupt) to be stored into a holding area (IOCSCPCNT+1).

Blocks 360, 361, IOCSCPTO: The checkpoint record is written on the checkpoint tape and checked for error during writing. If an error is detected, the tape is backspaced and skipped, and the record is rewritten. Five attempts can be made before an error halt occurs. If EOF occurs on the checkpoint tape, a halt is made to allow changing the reel.

Block 362: Accumulators 2 and 3 are replaced with the proper information from IOCSBCERMS+4 and IOCSBCERMS+2. Exit from this block will be a branch to 0+IOCSIXG. This returns control to the CHPT macro or the end-of-reel routine (routine from which checkpoint had been taken).

Block 363: The priority mask is set to allow priority interrupt and control is passed to the next instruction in the main program.

Block 365, RESTART: The RSTART macro is initialized by placing the channel and unit from word 50 into the tape operations. Word 50 is manually stored from the keyboard at the beginning of the restart operation. This information is typed when the checkpoint is taken.

Blocks 366, 367, READRESTRT: The label (if any) is processed and the restart program is read into storage. It is tested for error and if an error occurred, the tape is backspaced, reread, and tested again. Nine retries are attempted before a HALT occurs. The same routine is used for a label error. Control is then passed to the restart routine.

Block 368, IOCSRSTRT: The first 204 words of the checkpoint record are read into storage and checked for a CLR. If the first word of the record is not bCHPT, the next record is read into storage. When a record whose first word is bCHPT is detected, positions 5 to 9 of the second word in the record are compared to the checkpoint number. This number had been keyed into word 50 at restart and stored in accumulator 3 by the RSTRT macro. If positions 5 to 9 of the stored checkpoint indicator are higher than those of the record, a BH is made to read again. If it is equal, a BE is made to proceed with the restart operation at block 369. If neither of the previous conditions is met, the tape position is past the proper checkpoint record. When this occurs, a halt is made. If the start key is depressed, a branch is made to the load program at 0308 so that the restart may be attempted again.

Block 369, IOCSRRWLP: The checkpoint tape is backspaced so that the next read operation can restore storage. All other active files have their associated tapes rewound. These operations are accomplished by examining the DTF's for activity and, if active, using the DTF channel and unit to alter the rewind and associated instructions for the indicated tape unit. After the last DTF has been tested, a branch is made to the label routine at IOCSRLBCK.

Block 370, IOCSRLBCK: Using information from the DTF's, labels are processed on all active tapes except for the checkpoint tape. All DTF's are examined. After the last DTF is tested, a branch is made to the tape positioning routine at IOCSRPOS.

Block 371, IOCSRPOS: The tapes are positioned in two sequences of operation. The first sequence positions channel 2 tapes, and the second positions those on channel 1. The DTF's are examined for activity and channel. If they are active and on the proper channel, the tape for the given file is positioned according to the block count in the DTF. All DTF's are tested and their corresponding tapes are moved to the same position they had when the checkpoint was taken. The checkpoint tape is not moved, as it will be in its proper position after storage has been restored by reading this tape.

Block 372: The sign change and mode settings are restored in word 100 during a forced interrupt, setting the conditions into the machine. If SPOOL is specified, the SPOOL switches are saved before storage is restored. A read is initiated on the original checkpoint RDW's to restore storage. Control then passes to block 362, where accumulators 2 and 3 are restored. A branch is made to 0+IOCSIXG, returning control to the PC instruction in the CHPT macro (or the EOR routine) and then to the remainder of the main program. At this point, storage and all tape units are in the same condition as if this checkpoint had just been made.

BSP, RWD, WSM, and WTM Macros

During normal processing, it is occasionally necessary for the main program to request the following input-output operations on tape units under the control of IOCS.

MACRO NAME	OPERATION
BSP	Backspace
RWD	Rewind
WSM	Write segment mark
WTM	Write tape mark

MACRO FORMAT

Operation	*Operand*	
BLX	IOCSIXG,IOCSLMSR	
DC		
	+operation code	(0, 1)
	+number of operations	(2, 5)
	+DTF address for file	(6, 9)
INCL	LMSR1	

Since the conditions of these tape units are unknown to the main program, a special IOCS subroutine is necessary to perform the operations requested by the macros. The subroutine positions tapes according to the parameters of the macros and performs the requested number of operations.

Initial positioning of input tapes is required to backspace the tape over unprocessed records before processing the macros listed above.

The following description of the flow chart, Figure 12, explains the operations to process the BSP, RWD, WSM and WTM macros.

Block 375, IOCSLMSR: This block represents the beginning of the subroutine to process BSP, RWD, WSM, and WTM operations. It is entered by a BLX from one of the above macros, or from IOCSFEORI during a FEORN (force end of reel, input) operation. The free switches are tested until both channels are free, making certain that all pending tape operations have been completed. The block also initializes the subroutine, using information (channel number, tape unit number, etc.) from the DC in the macro.

Block 376: A test is made to determine if the tape unit is input or output. If it is output, a branch is made to block 382 to bypass input operations.

Block 377: For input files, the pending counter is increased and the block count decreased, by the number of areas indicated by the availability counter, to correct the counters for tape positioning that may be necessary.

Block 378: A test is made of the availability switch. If it is off, a branch is made to block 380, indicating that the tape has been positioned correctly.

Block 379, IOCSLMRLB: A backspace operation is performed. The scheduler for this file is then entered to make the next area available and decrement the availability counter. If the counter becomes zero, the availability switch is turned off. A branch is then made back to block 378 to retest the availability switch.

Block 380: The number of operations to be done (macro-instruction parameter) is placed into accumulator 1 to keep the operation count.

Block 381: A five, indicating the number of retries on an error, is placed into index word 99. This allows a BIX on 99 to perform five retries before halting.

Block 382, IOCSLMTPO: The tape operation is performed. This instruction was initialized with channel number, tape unit number and operation code by block 375.

Block 383: The operation type is tested. A rewind operation branches to block 384. A backspace operation branches to block 385. A write segment mark or write tape mark operation continues processing in block 386.

Block 384, IOCSLMRWD: The block count is zeroed to make it agree with tape position at load point, after the rewind is completed. A return is then made to the instruction after the RWD macro in the main program.

Block 385, IOCSLMBSP: The block count is reduced by one to make it agree with the new tape position. A branch is then made to block 387 to test for completion.

Block 386: A ready loop is entered to allow completion of the tape operation. When the channel becomes free, the condition code is tested for error. If an error has occurred, control passes to block 388 to attempt correction. If no error, processing continues to block 387.

Block 387: Accumulator 1 is decremented by one and a BZ1 is performed. If the branch is made, the required number of operations are completed and a return is made to the instruction after the macro, in the main program.

Block 388, IOCSLMER: The tape is backspaced and a skip operation is performed to skip over a possible bad area of tape.

Block 389: A BIX is made on index word 99. The non-indexing portion was set to 5 by block 381. A branch on the BIX returns control to block 382 to retry the operation.

Block 390: After five retries an error condition still exists. This block represents a HP to notify the operator of this situation. A start at this point resets the retry count and repeats the operation.

RDSF and RDSB Macros

The RDSF (read segment mark forward) and RDSB (read segment mark backward) macros space over records through a given number of segment marks defined in the operand of the macro. A subroutine (LMSR2) is included by either macro to perform the tape positioning. The macros have the following format:

Operation	*Operand*	
BLX	IOCSIXG, IOCSRSF	
DC		
	+operation code	(0, 1)
	+number of operations	(2, 5)
	+DTF address for file	(6, 9)
INCL	LMSR2	

The operation for the RDSF or RDSB macros is given in Figure 13.

Block 400, IOCSRSF: A loop is entered that tests the free switches of both channels. This loops and tests until both channels are free. The routine is then initialized, using information (channel, unit, etc.) from the DC generated by the RDSF or RDSB macro-instruction.

Block 401, IOCSLMSTEF: A test is made to determine if the file is input or output. For output files, a branch is made to IOCSLMSGO (block 405), bypassing input operations.

Block 402: The file scheduler for this file is conditioned by increasing the pending counter and decreasing the block count, by the number of areas available.

Block 403: A test is made of the availability switch. If it is off, control passes to IOCSLMSGO (block 405). This indicates that the tape has been positioned properly in preparation for the tape operation indicated by the macro (RDSF or RDSB).

Block 404, IOCSLMSRB: The tape is backspaced and the file scheduler for this file is entered. In the scheduler, a new area is made available, the availability counter is decremented by 1, and the availability switch is turned off when its associated counter becomes zero. A return is made to block 403 to retest the availability switch.

Block 405, IOCSLMSGO: The required tape operation is performed. A ready loop is then entered to allow completion of the operation before continuation of processing.

Block 406: The condition code is tested for an error. If an error occurred, control passes to block 407.

Block 407: A message is typed, indicating that the error occurred. A HP is then performed, allowing the operator to take corrective action.

Block 408: A PC operation is performed to allow interrupt and a return is made to the main program to continue operation.

RDLIN Macro

This section describes operation for the RDLIN (read label, input) macro, given in Figure 14. This macro is used to update the label record in the label DC for input files with new label data from label cards (columns 8 through 40) read from the card reader. One label card updates one label DC.

Block 411, IOCSRDLIN: The label card is read into storage. If an error or EOF condition occurs, an error message is typed, followed by a halt (block 417) to allow the operator to correct the card.

Block 412: During the running of programs using IOCS with tape labels, storage location 0109 must contain the current date. This date will be in the form +YYDDD00000, where YY is the tens and units positions of the year and DDD is the number of the day within the year. The date in word 0109 is compared to the current date in the label card (word 1, positions 0-4). If they are not equal, a branch is made to block 417 to inform the operator of this condition. If they are equal, control is passed to block 413.

Block 413: The channel and unit from the label card (word 1, positions 5-6) are placed into accumulator 1.

Block 414, IOCSLASMOD: A termination card must follow the last label card into the card reader. This card will have 00 punched into the channel-unit columns. Block 414 will recognize this "end" card with a BZ1 and return control to the main program.

Block 415: Blocks 415, 416, 418, and 419 form a processing loop to compare the channel and unit number of the label card with those in the DTF's (word 1, positions 0-1). Block 415 places the channel and unit number of the DTF being tested into accumulator 1. The address of the first DTF is obtained by addressing the symbol IOCSFTBL01. The remaining DTF's are tested by increasing the testing address by 9 (number of words per DTF). If the proper DTF is not found, a message is typed.

Block 416: Recognition of the NOP after the last DTF in storage is accomplished by a BM1. If no branch is made, the DTF is tested by block 418.

Block 417: The message "CH DR ERR" (channel drive error) is typed, indicating that the channel and tape unit number in the label card should be checked by the operator. A halt occurs to allow for correction and rerun.

Block 418: The channel and unit number of the DTF under test is compared to that in the label card record. If they are equal, control is passed to block 420 for processing.

Block 419: The address of the DTF under test is incremented by 9 (number of words per DTF), to alter the testing address to that of the next DTF.

Block 420: The number of exit addresses in the label DC are counted, using information in the label mask. This number (number of words containing exit addresses) is added to LABELINF (word 7, positions 6-9 of the DTF) for this file. The result will be the address for the first word of the label record in the label DC.

Block 421: The new label data words of the card are transferred to the label record of the label DC, one word at a time. If any word is zero, no change in the original data is made for that word. When all changes are completed, control is passed to block 411 to read the next card.

DEOR Macro

The macro DEOR (delay end of reel) is used to prevent normal end-of-reel procedures from occurring on an output file. Records may then be written past the end-of-reel reflector. After the desired number of records have been written beyond the reflector, end-of-reel can be forced using the FEOR (force end of reel) macro. End-of-reel recognition to the main program is given by turning on the electronic switch named in the operand of the DEOR macro. This section describes the operation for the DEOR macro as shown in the flow chart, Figure 15.

Block 425: The electronic switch assigned for the DEOR macro is turned off and its switch code number is placed into the file scheduler's DRDW, positions 0-1. Electronic switches are coded from 0 through 29. This DRDW is the first word of the scheduler and contains the location of the pending and availability switches in positions 2-9.

Block 426, IOCSIDEOR: The branch address of the scheduler exit used when the condition code is not equal to 2 (other than correct length record), is altered to that of the DEOR subroutine, IOCSDEOR.

Block 427, IOCSDEOR: This block is entered from the file scheduler when the condition-code comparison to 2 is unequal. The condition code is compared, in this block, to 5 (EOR). If the comparison is equal, control passes to block 428. If it is unequal, a branch is made to IOCSIRTAIN to take necessary action for the abnormal condition of the previous tape operation.

Block 428: The routine is initialized by placing the electronic switch number and the channel and tape unit into their respective operations to make possible the processing of block 429.

Block 429: The end-of-file indicator is turned off. The electronic switch that was allocated to the DEOR operation is turned on. This operation informs the main program that end of reel was reached.

Block 430: The condition code is altered to 2. A branch is then made to the scheduler to complete the operation as if it were normal.

FEORN Macro

The force end of reel (input) macro is used to start an end-of-reel operation for an input file at any desired time. The FEORN operation must be preceded by a RLSE macro to cancel any partially used area. During the processing of the FEORN macro, end-of-reel procedures are completed, with the exception of trailer label reading and testing, and user's end-of-file procedures. This section describes the operation of the FEORN macro as shown in the flow chart, Figure 16.

Block 440, IOCSFEORI: The LMSR (little macro subroutine) is included by the FEORN macro, and a portion of this routine is used to condition the file scheduler. Block 440 sets the address of the entry IOCSFORCE into the 6-to-9 portion of the BE (availability switch off branch) in the LMSR routine.This provides a return to the FEORN subroutine after processing the file scheduler. A branch is then made to IOCSLMSR.

Block 441, IOCSLMSR: This routine is used primarily to determine the number of available areas in order to set up the file's index words, so that the next GET (if any) causes a read into the proper area. When the availability switch is off, a BE will branch to IOCSFORCE to resume forcing the EOR condition.

Block 442, IOCSFORCE: A test is made of the second word in the EOR routine to check for labels. This word in EOR1 (DIOCS entry) is different from that in the EOR2 routine. Since no labels may be used with EOR2, recognition of EOR2 will cause a branch around the label bypass alterations (block 443) to block 444.

Block 443: The BCX instructions in the EOR routine located at IOCSTEF+5 and IOCSTEF+9 are altered to branch instructions. These branches bypass trailer reading and testing operations.

Block 444: The STD1 operation in the end-of-reel routine that places the user's exit address for end of file is altered to make it a branch to IOCSEOFEX+2. This effectively bypasses the end-of-file exit during the end-of-reel procedure.

Block 445: An exit is placed into the EOR routine at IOCSOPNSW2+4 to effect a return to this subroutine (FEORN) after EOR procedures are completed.

Block 446, IOCSEOR: The EOR routine is entered at IOCSEOR. This routine rewinds the tape as specified by RWDPROC of the DTF. If an alternate tape is specified, the DTF tape addresses will be interchanged and the flip-flop message typed. If labels are specified, the header on the new tape will be read and tested.

Block 447, IOCSRTF: The second word of the EOR routine (IOCSEOR+1) is tested for content. This word in the EOR1 (DIOCS entry) routine is different from that in the EOR2 routine. Alterations were made to bypass trailer operations in the EOR1 only. If EOR2 is recognized a branch is made to block 449 bypassing the restoration procedures for EOR1.

Block 448: The branch instructions bypassing trailer reading and checking in EOR1, altered by block 443, are restored to their original condition (BCX).

Block 449: The instructions common to both EOR1 and EOR2 are restored to normal. Control is then passed to the main program.

FEOR Macro

The macro FEOR (force end of reel, output) is used to start end-of-reel procedures on an output file at any given time. It is unnecessary to perform a RLSE operation before the FEOR macro, as the FEOR routine uses the CLOSE routine to write any incomplete block of records. End-of-reel procedures are performed to write a trailer (if specified), interchange tapes per DTF, and perform header reading and testing as specified. The FEOR operation is shown on the flow chart, Figure 17.

Block 450, IOCSFEORT: The CLOSE routine is initialized by altering the instruction that turns on the CLOSE exit in the end-of-reel routine, insuring that the exit remains a NOP. An exit that returns control to the FEOR subroutine is placed into the end-of-reel routine beyond the programming for trailer processing. The address of the DTF for the file is stored into the word after the BLX to IOCSICLOSE. The BLX and the following two words simulate a CLOSE macro to close one file. Entrance is then made to the CLOSE routine at IOCSICLOSE.

Block 451, IOCSICLOSE: Normal IOCS CLOSE procedures are performed for the file with the usual output file exit to the EOR routine.

Block 452: Normal IOCS trailer procedures are performed for the file.

Block 453, IOCSEOR: The tape is rewound as specified by RWDPROC in the DTF for the file. If an alternate tape is specified, the DTF tape addresses will be interchanged and the flip-flop message typed.

Block 454, IOCSLOPRO: If labels are specified, the header on the new tape is read and tested. A return is made from the stored exit at IOCSOPNSW2+4 to the FEOR subroutine.

Block 455, IOCSFRET: Restoration is made of the altered instructions in the CLOSE and end-of-reel routine. A return is then made to the main program after the FEOR macro.

Unit Record Card Files

This section contains an example of the machine instructions produced by Autocoder using IOCS, whenever an input card file is described by a DUF entry. The coding produced by a GET macro for a card file, and the RDW's for an input area of the card file are shown in Figure 18. A brief description of the coding follows.

When the BLX, produced as a result of the GET-card macro-instruction, is executed, the program goes to the routine produced by the DUF entry on assembly. Entry into the read routine is at location IOCSUGET1. The index word specified in the fourth item in the DUF entry is loaded with the location of the RDW's for the card input area (DA entry, INPT label). The index-word sign is set to plus, since the record may have one RDW (last RDW is always minus).

A card read is executed to the RDW list of the INPT, DA entry. If no end of file or error occurs, a branch to 0+IOCSIXH is performed to return to the next sequential instruction in the user's program.

If the GET macro was a GET TO WORK AREA, the entry is IOCSUGWA1 and the sign of the return branch is set to minus, making it a NOP. The address of the RS instruction is set with the address plus 1 of the BLX generated by the GET. This is the location of the RDW for the TO area in a GET TO macro. For a GET TO WORK AREA instruction, the records are read into the input area and moved to the work area by the RS instruction. Return is then made to the user's program at the next sequential instruction.

If an error occurs in reading, the input area is typed and the machine halts to allow correction of the error. The card, after it has been corrected, can be put back into the card reader, and reading can be resumed by pushing the start key. If an error routine address had been supplied by the DUF entry, a branch to that address would occur after typing the message.

An EOF results in a branch to the address of an end-of-file routine as specified in the DUF entry. If no address for an end-of-file routine is given in the DUF, the program branches to IOCSEJLOOP in IOCS.

An example of a DUF entry, with GET and DA associated with it, is shown on the following page.

```
     LABEL          OPERATION          OPERAND                                  COMMENTS

     CARDFILE01     DUF            CARDFILE01, 1, 1, INPT, 28 SW3
          DUF1                     FILENAME, 1, SYNC, RDW, IXWORD,                NO SPOOL
                                   EOF ADD, ERRADD
     IOCSUGWA1      MSM            *+6    ENTRY FOR GET CARDFILE TO WA
     IOCSUGET1      XL             28,INPT     ENTRY FOR GET CARDFILE
                    MSP            28
     MACRO00111     UR             1,INPT
                    B              *+8
                    B              MACRO00112
                    B              0+IOCSIXH
                    MSP            *-1
                    ZAA            0 (6,9)+IOCSIXH
                    STD1           * (6,9)+1
                    RS             28,0
                    B              1+IOCSIXH
                    TYP            INPT
                    B              *+1
                    HP             0
                    B              MACRO00111
     MACRO00112     NOP            0
                    B              SW3
                    B              IOCSEJLOOP
     IOCSCRDW1      EQU            28,X

                                   USER'S PROGRAM

     GETCARD        GET            CARDFILE01                                   Macro Inst.
X*                                 GET CARDFILE01                               Produced by
                                                                                Autocoder
X    GETCARD        BLX            IOCSIXH,IOCSUGET1
                    .
                    .
                    .
     INPT           DA             2,RDW,0
X
     FIELD1                        0,9                                          Input Card Area
                                                                                for
     FIELD5                        150, 159                                     2 Cards
```

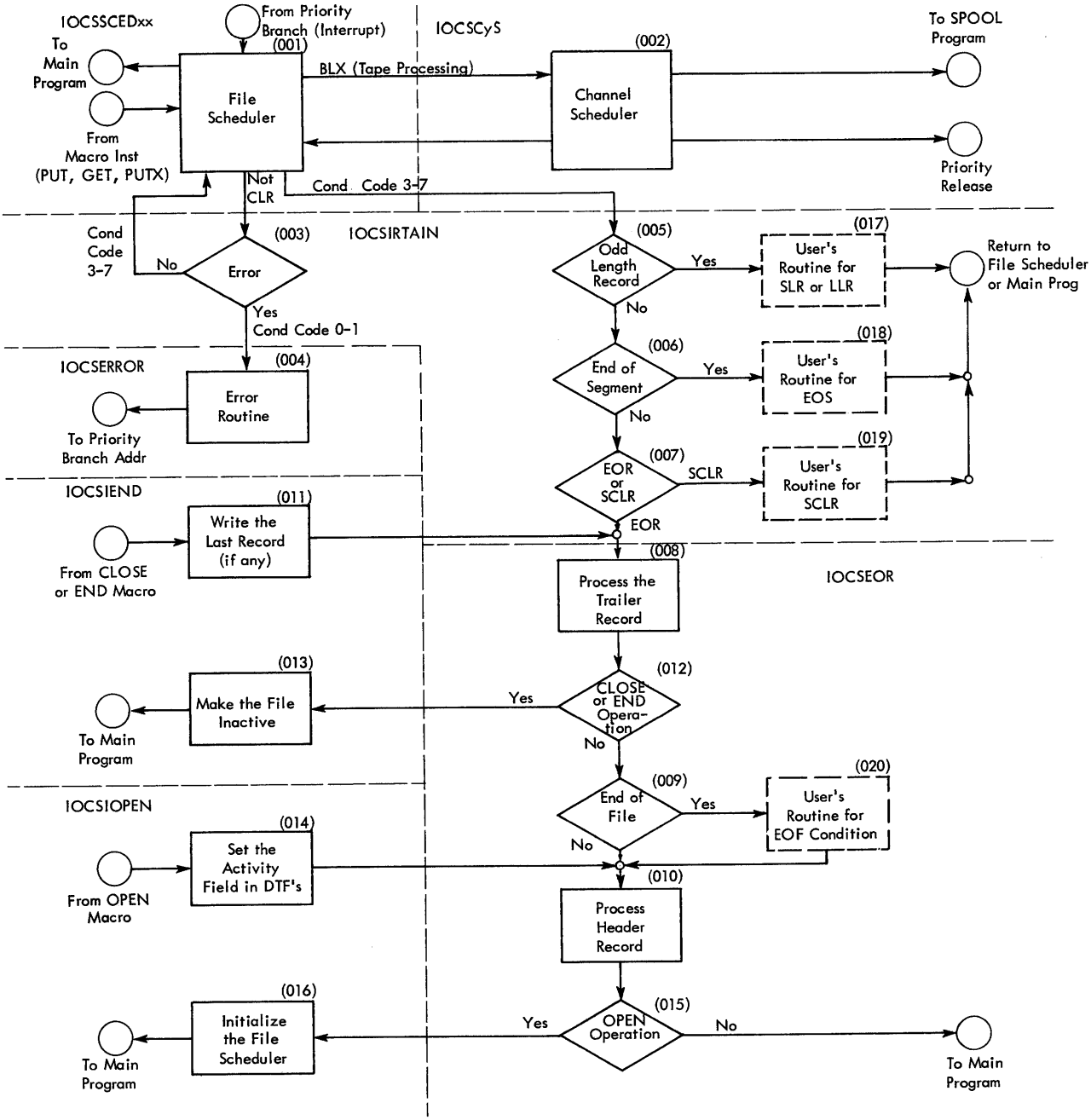


Figure 1. Correlation between IOCS Routines

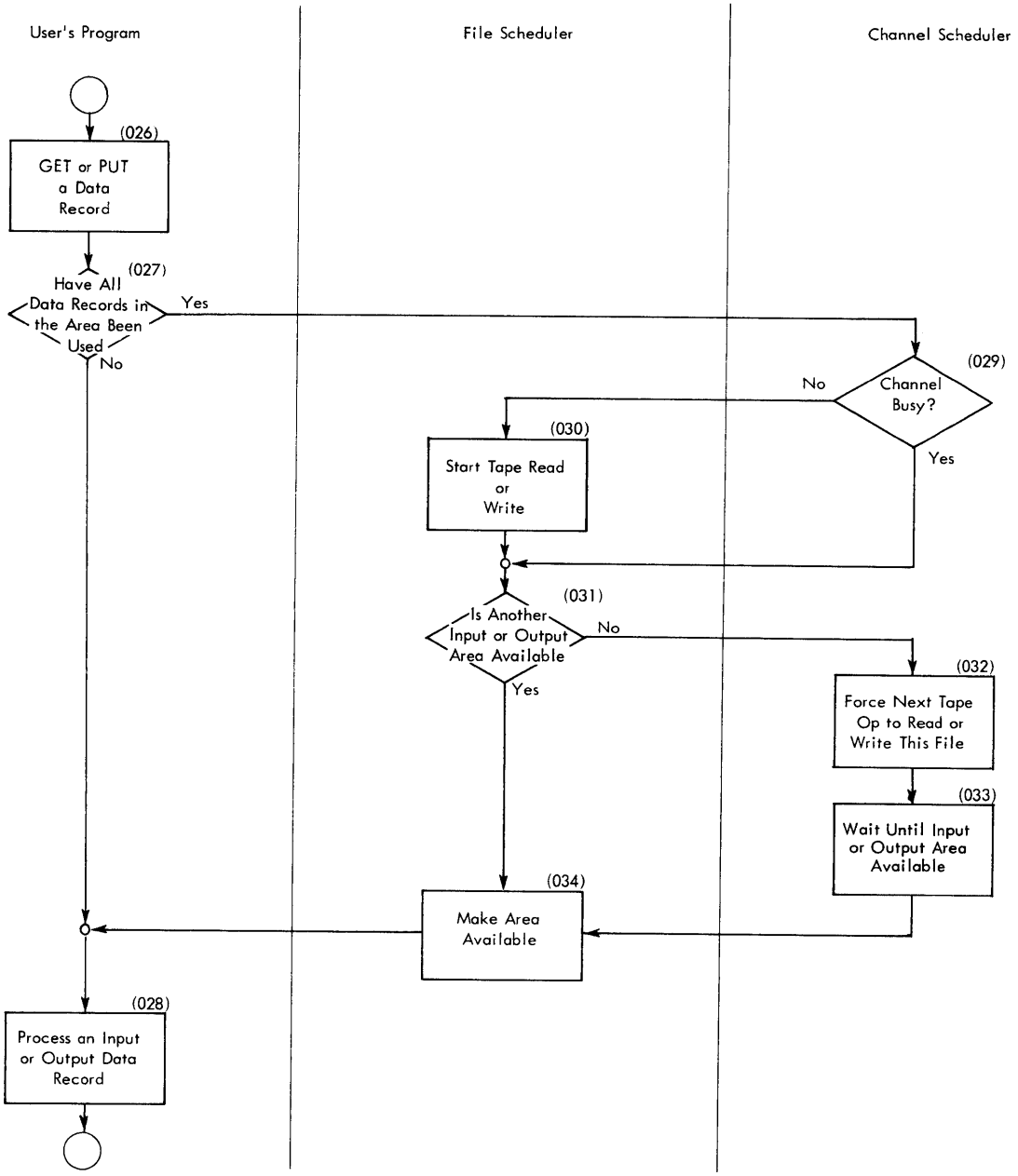


Figure 2. General Operation of Schedulers

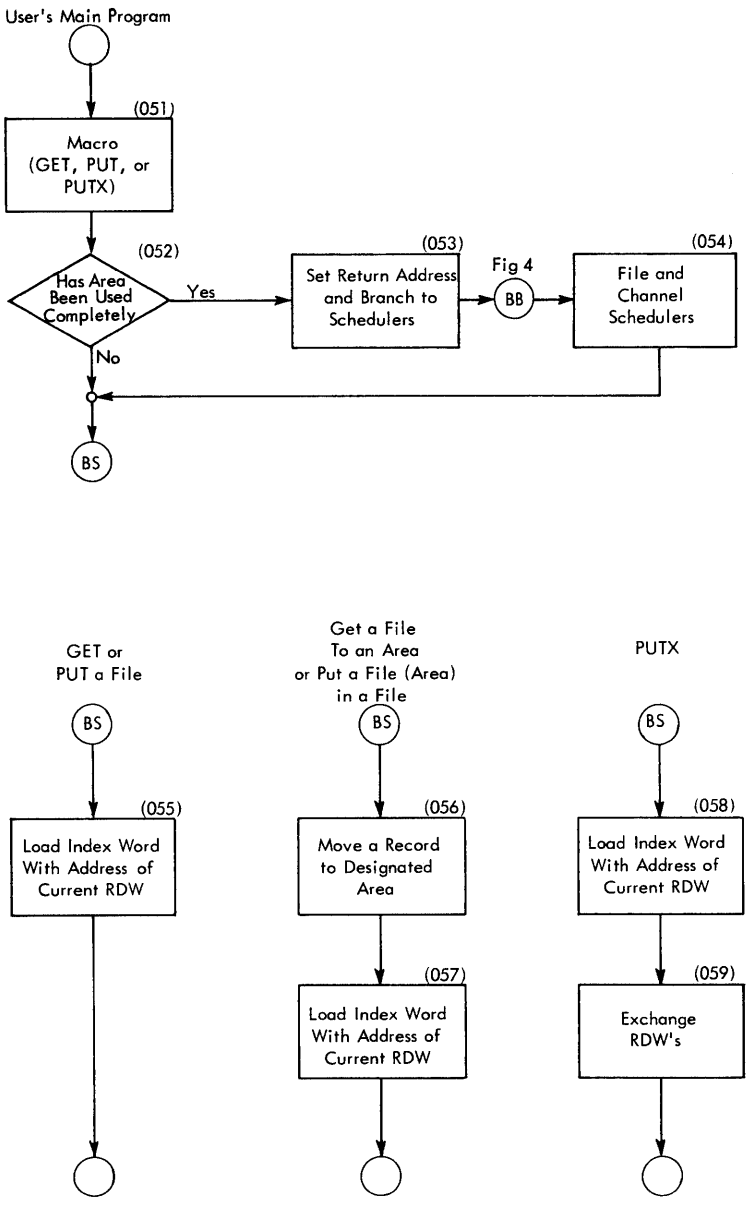


Figure 3. GET, PUT, and PUTX Macros

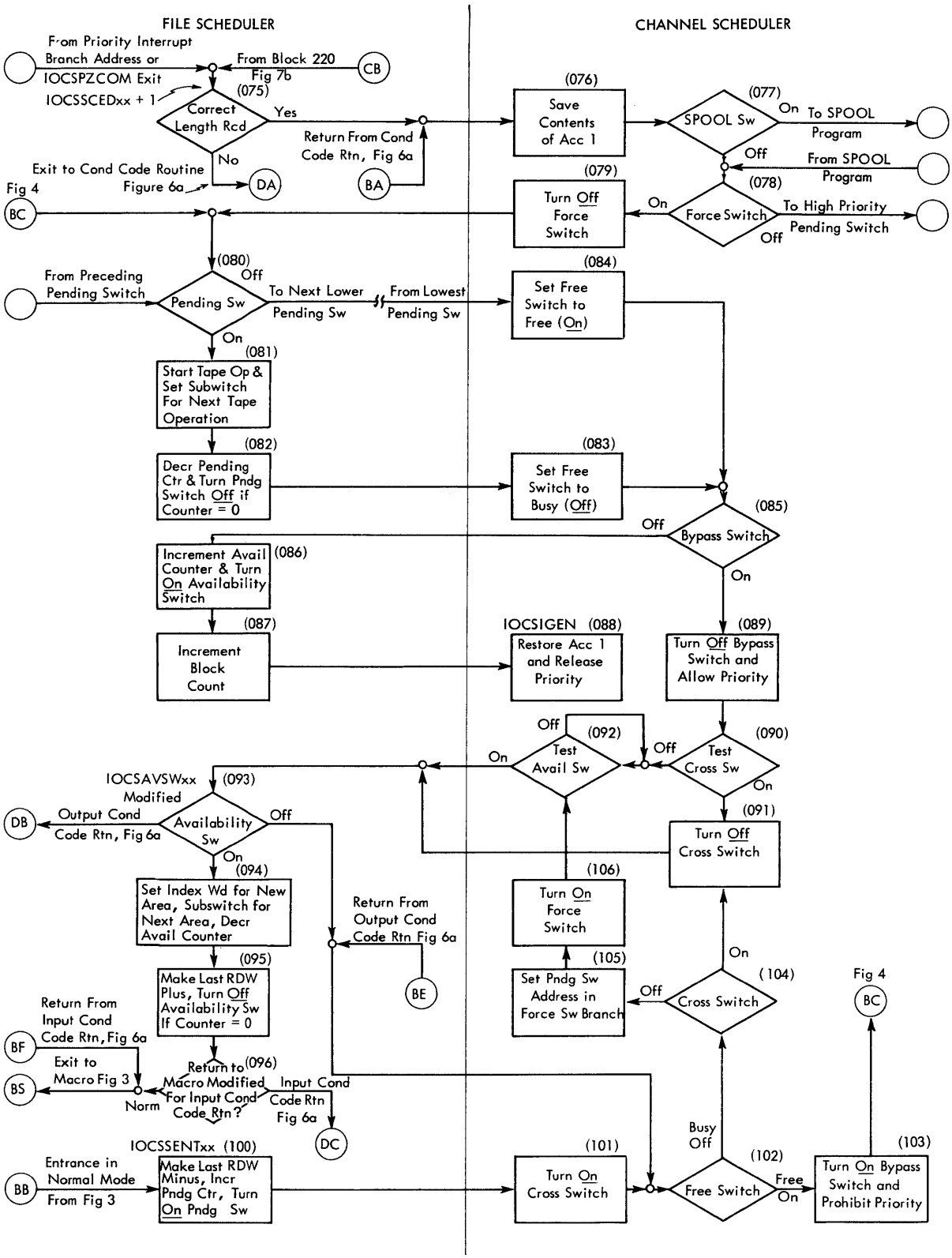


Figure 4. Channel and File Schedulers

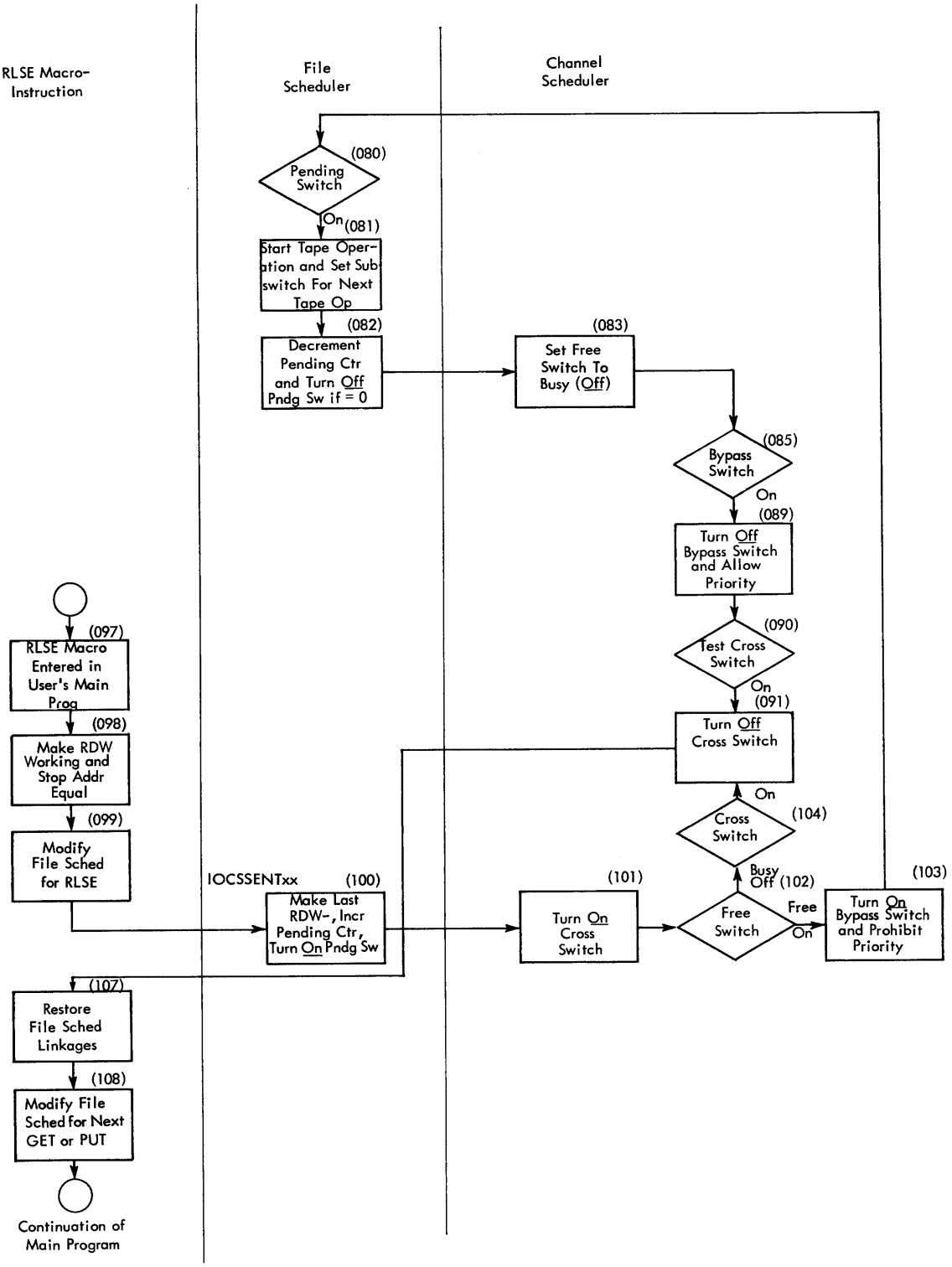


Figure 5. RLSE Macro

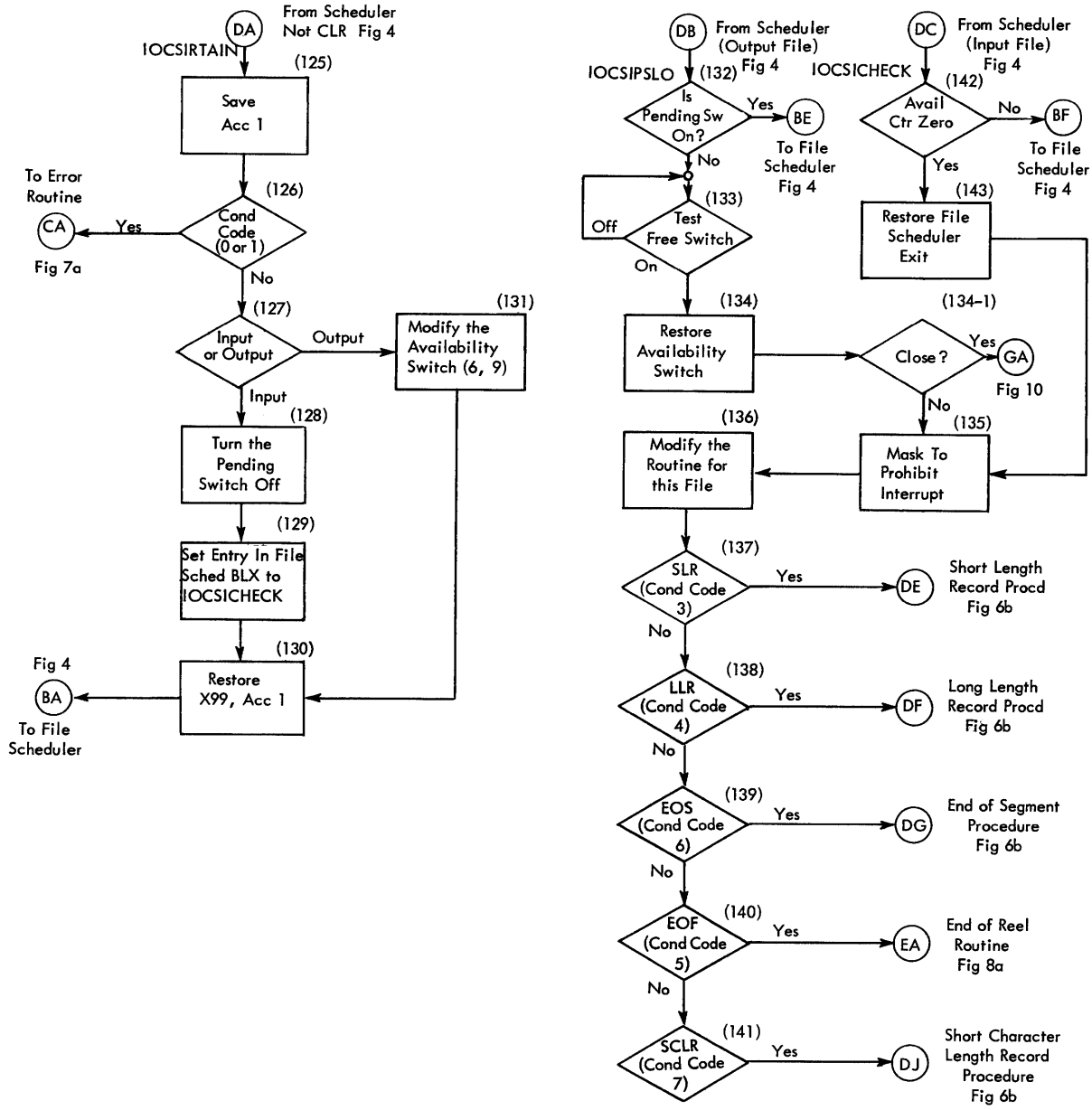


Figure 6a. Condition Code Routine

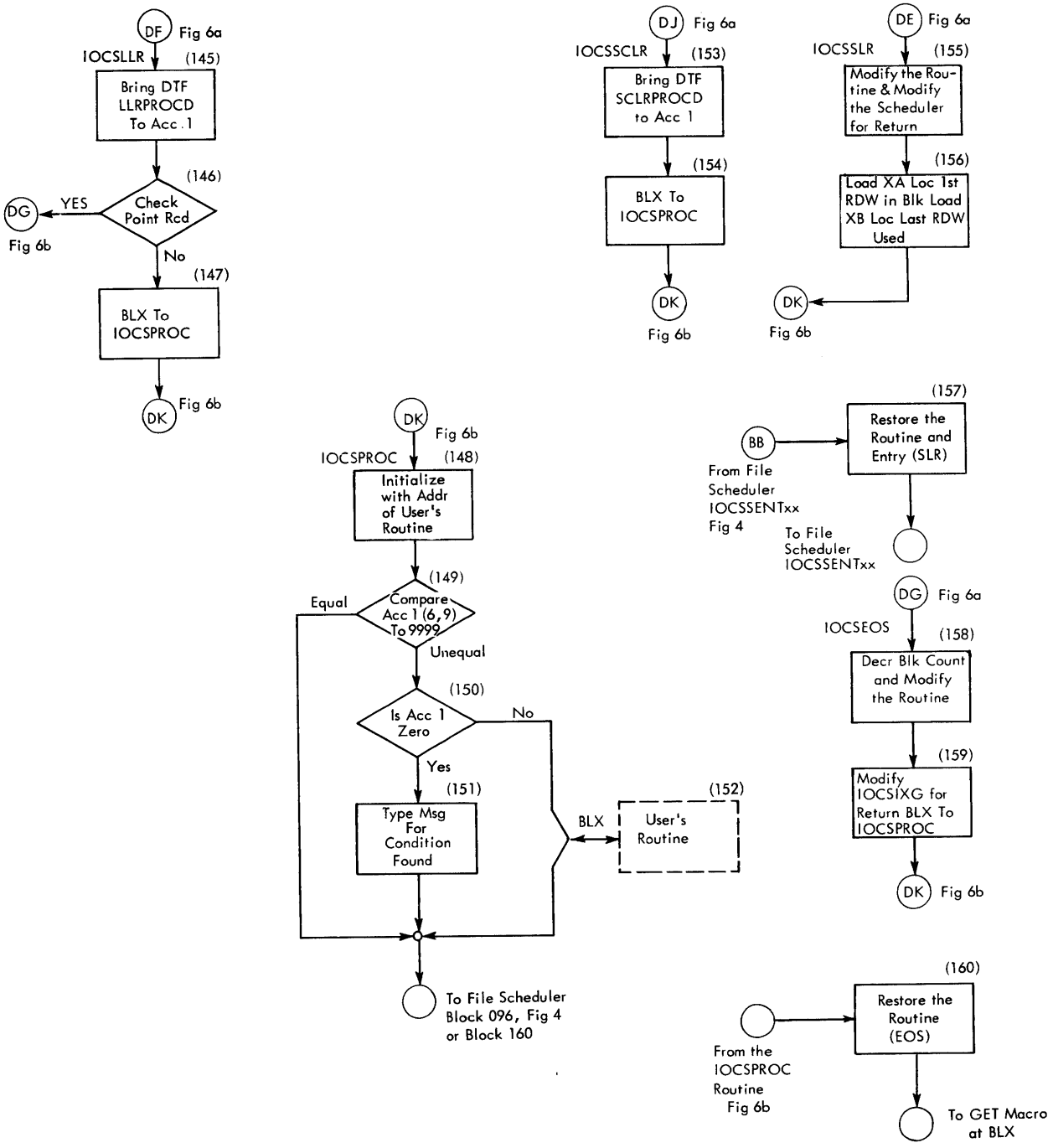


Figure 6b. Condition Code Routine

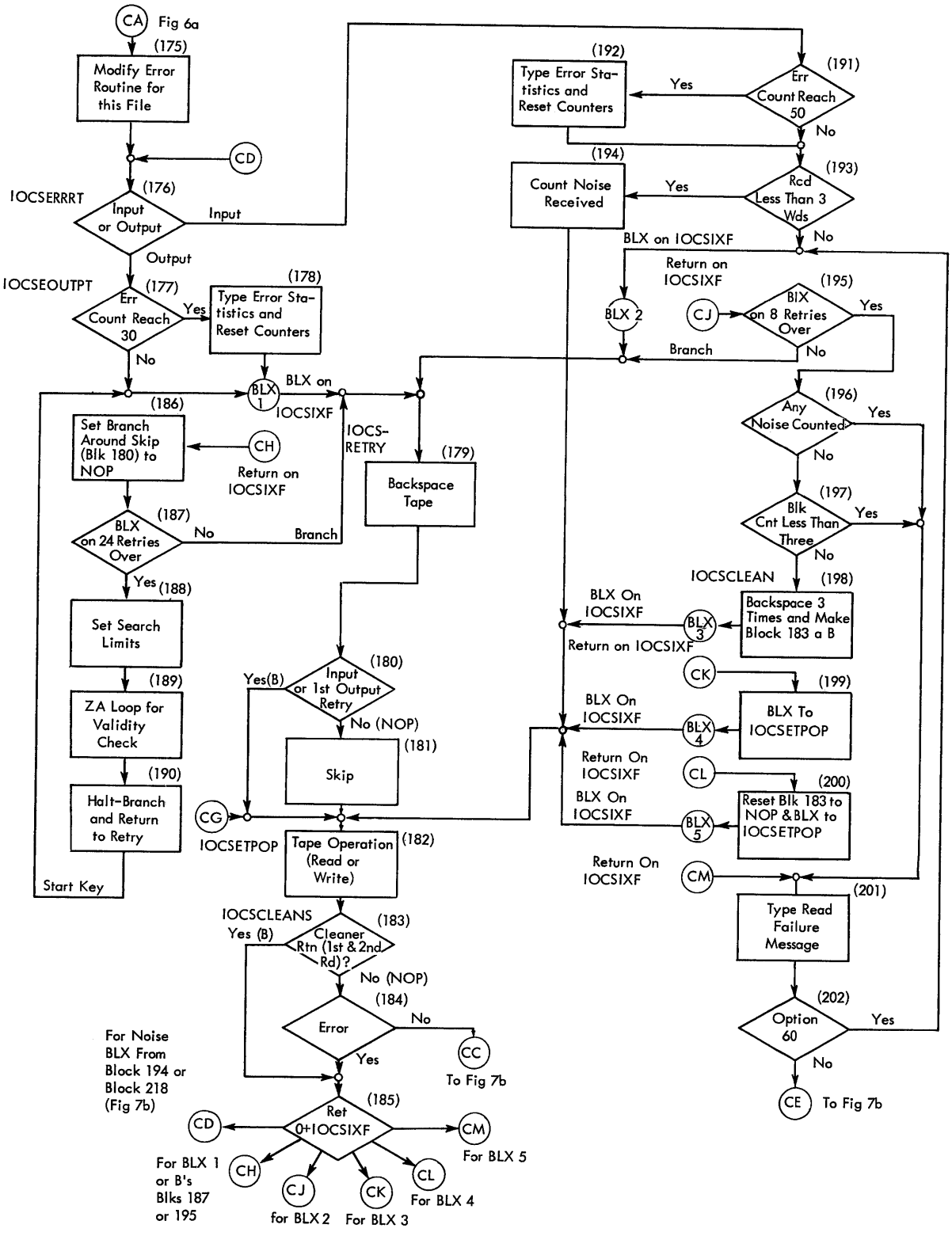


Figure 7a. Error Routine

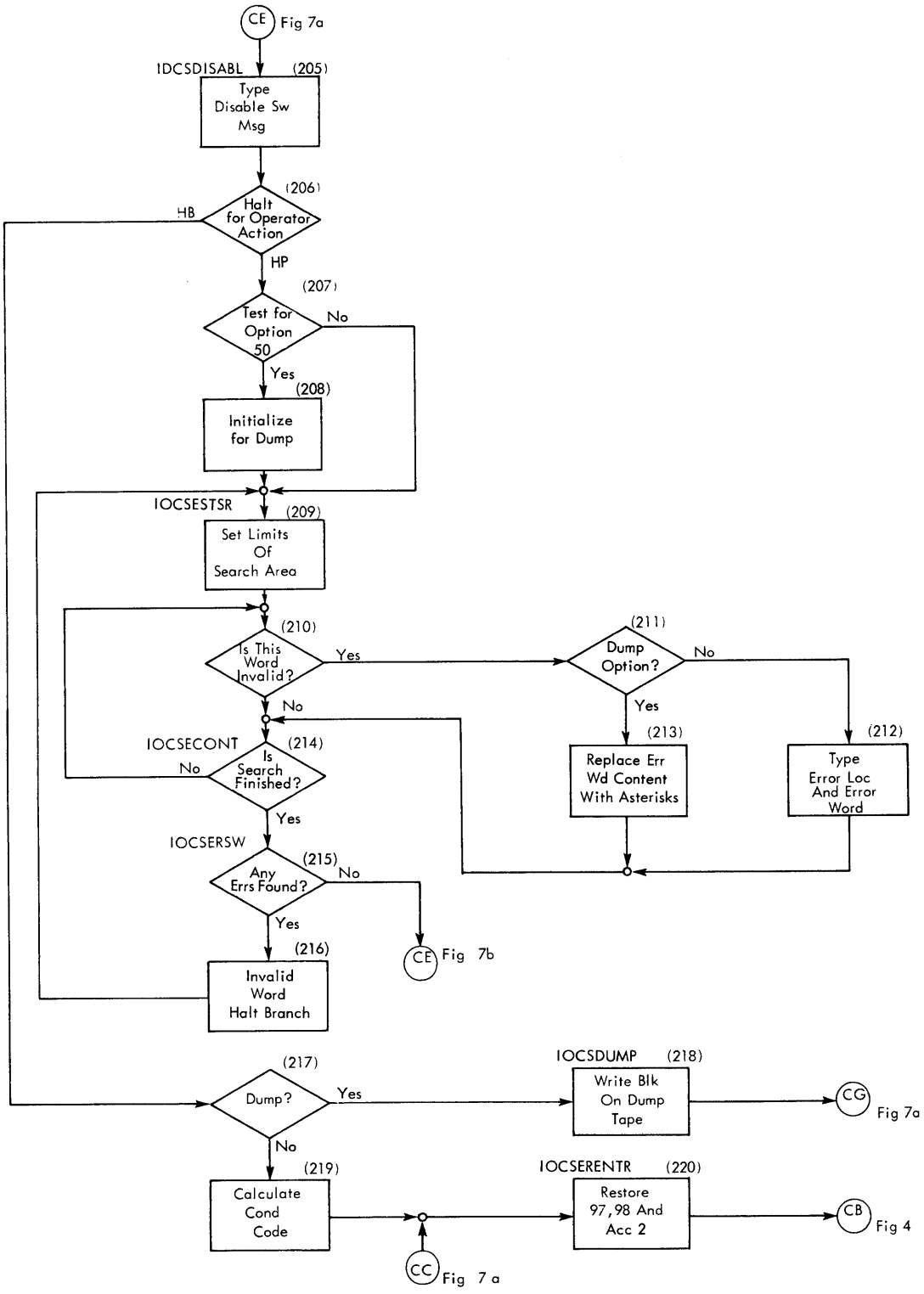


Figure 7b. Error Routine

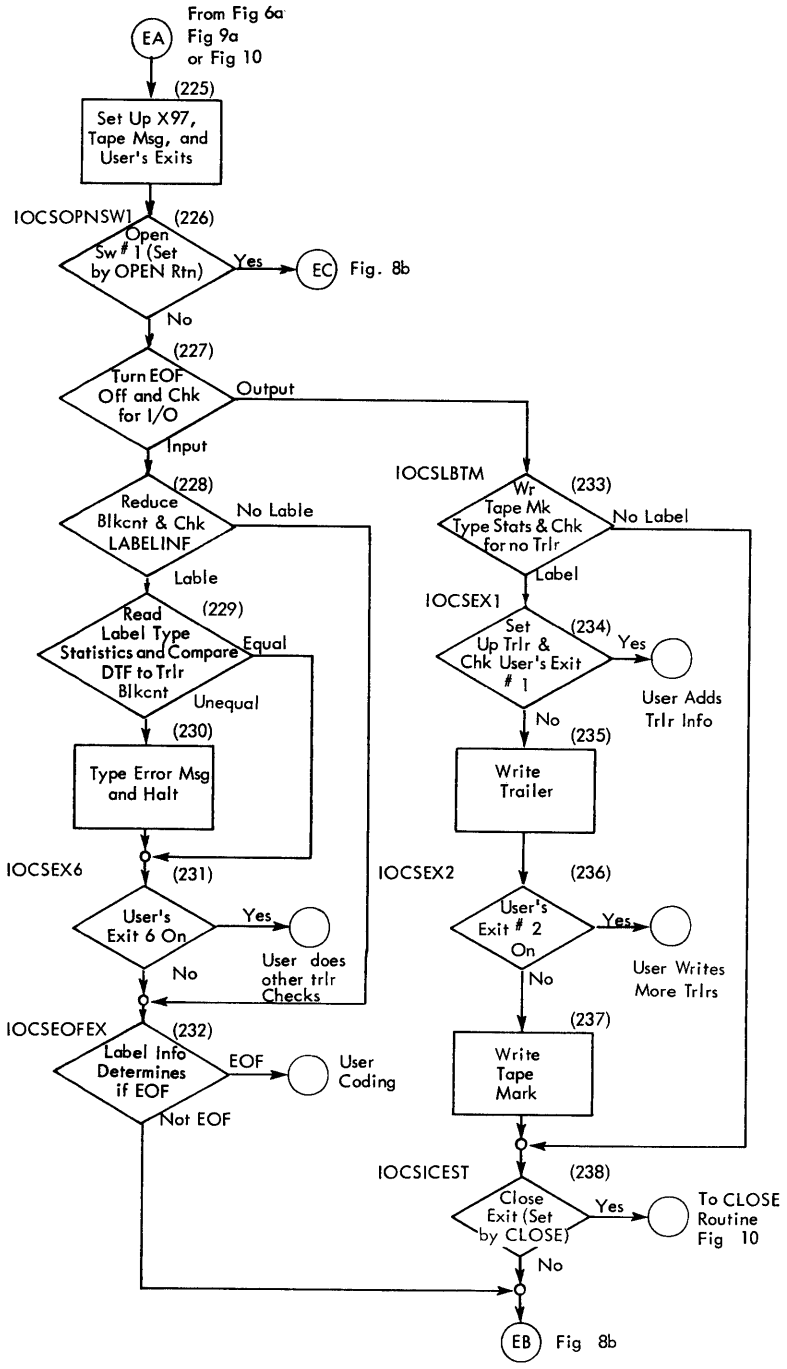


Figure 8a. End of Reel Routine

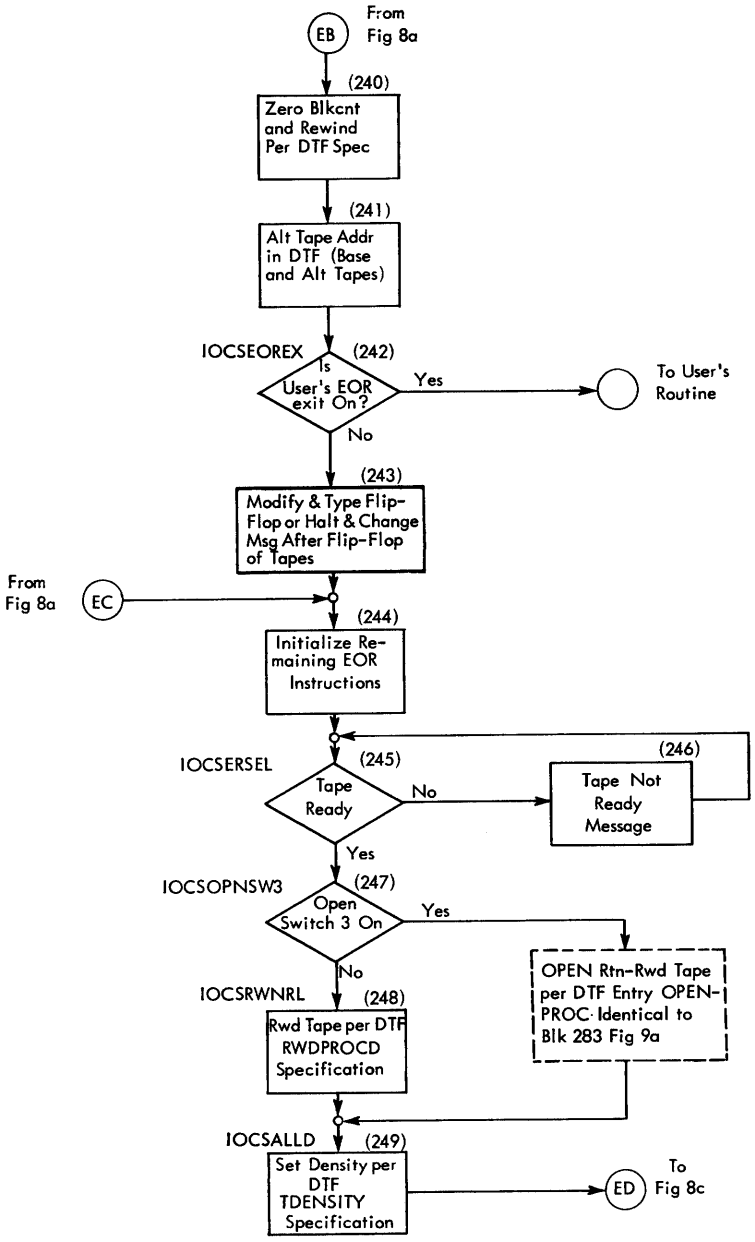


Figure 8b. End of Reel Routine

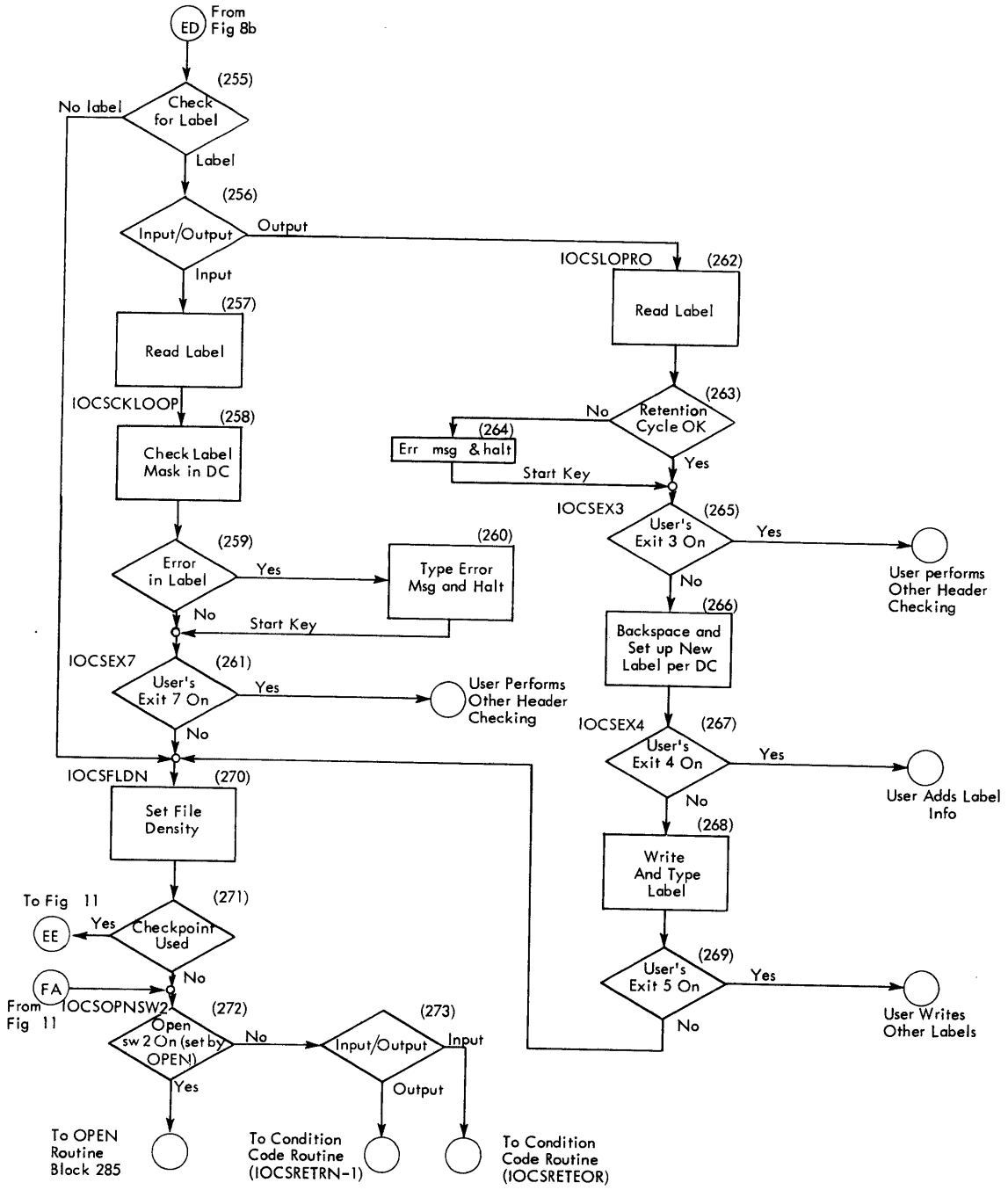


Figure 8c. End of Reel Routine

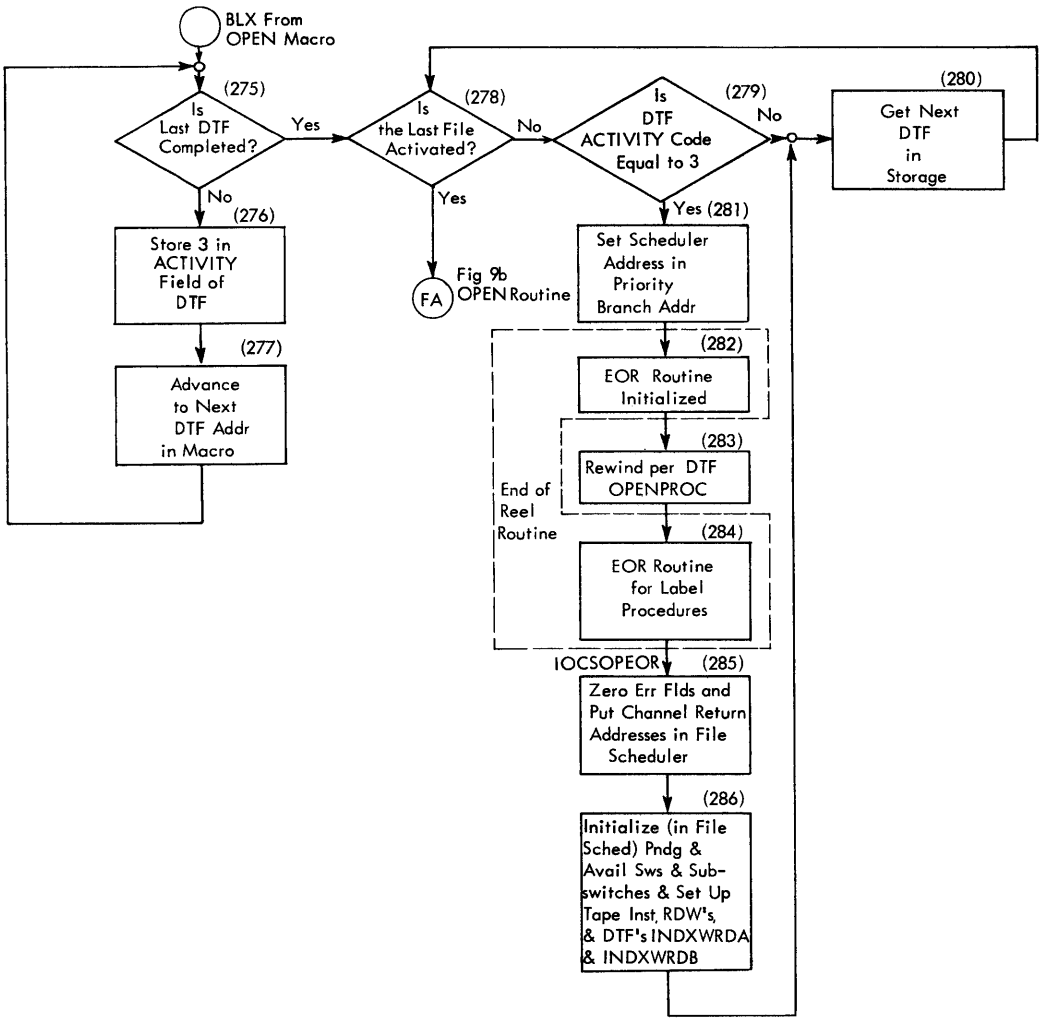


Figure 9a. Open Routine

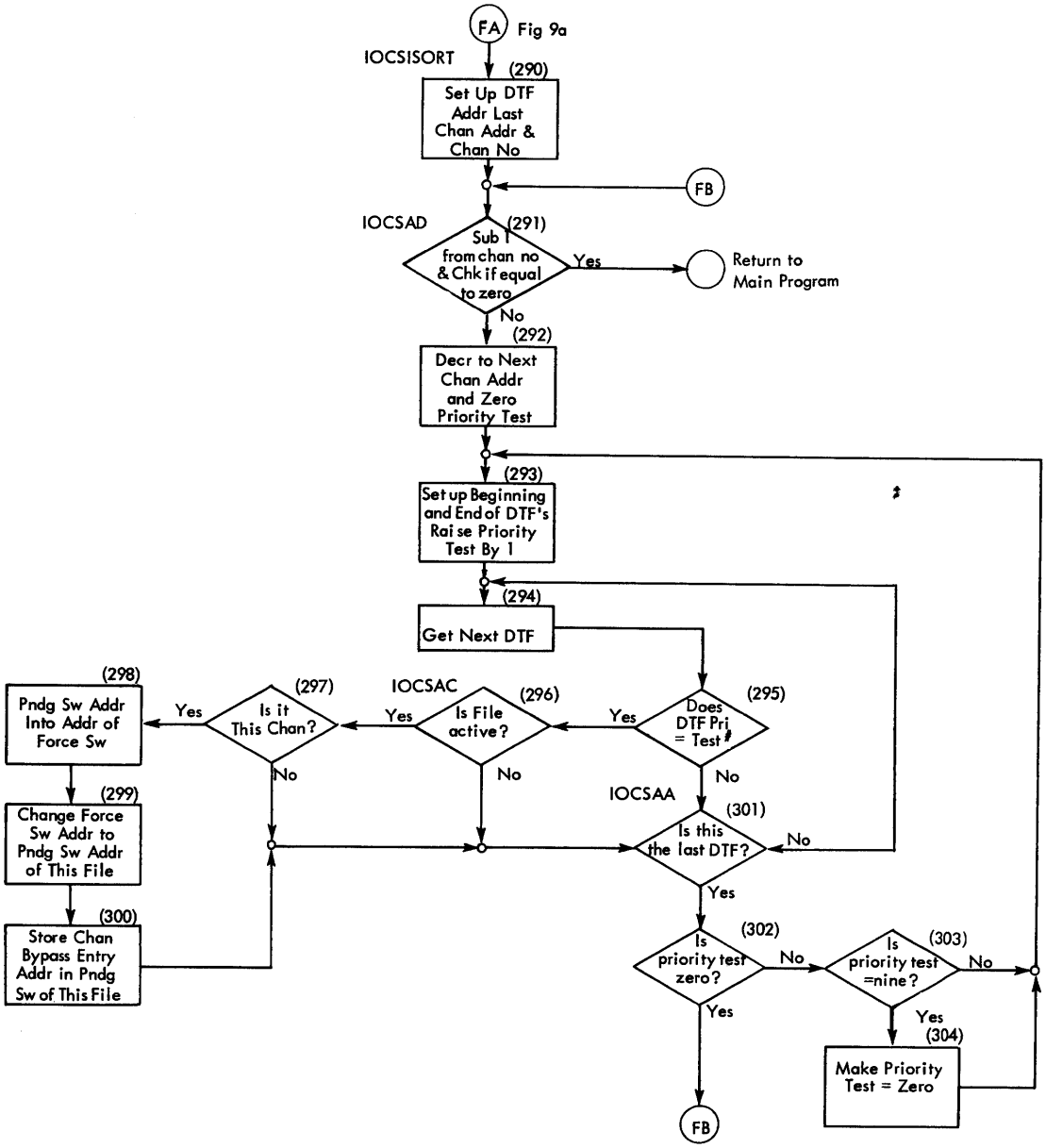


Figure 9b. Open Routine

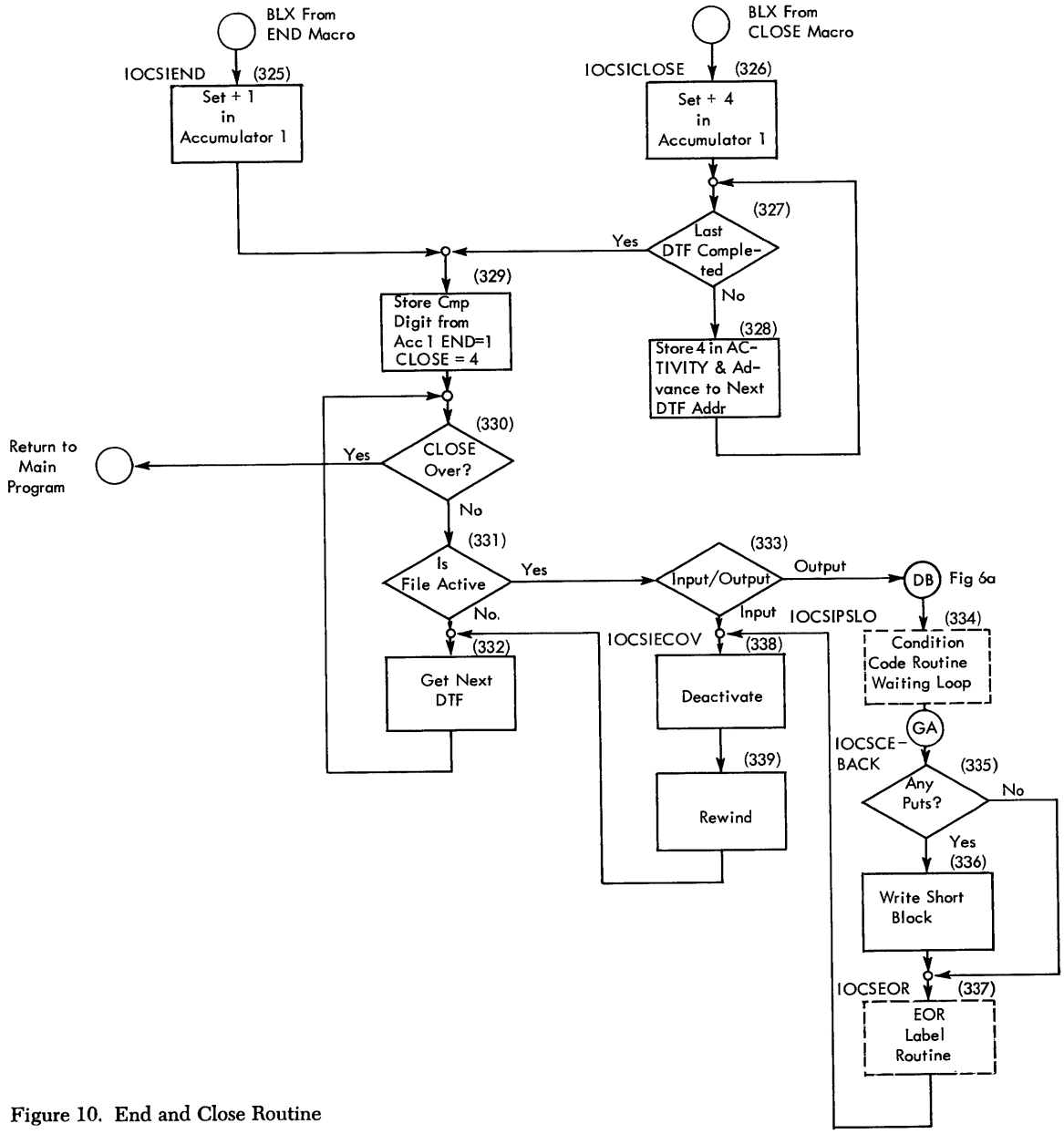
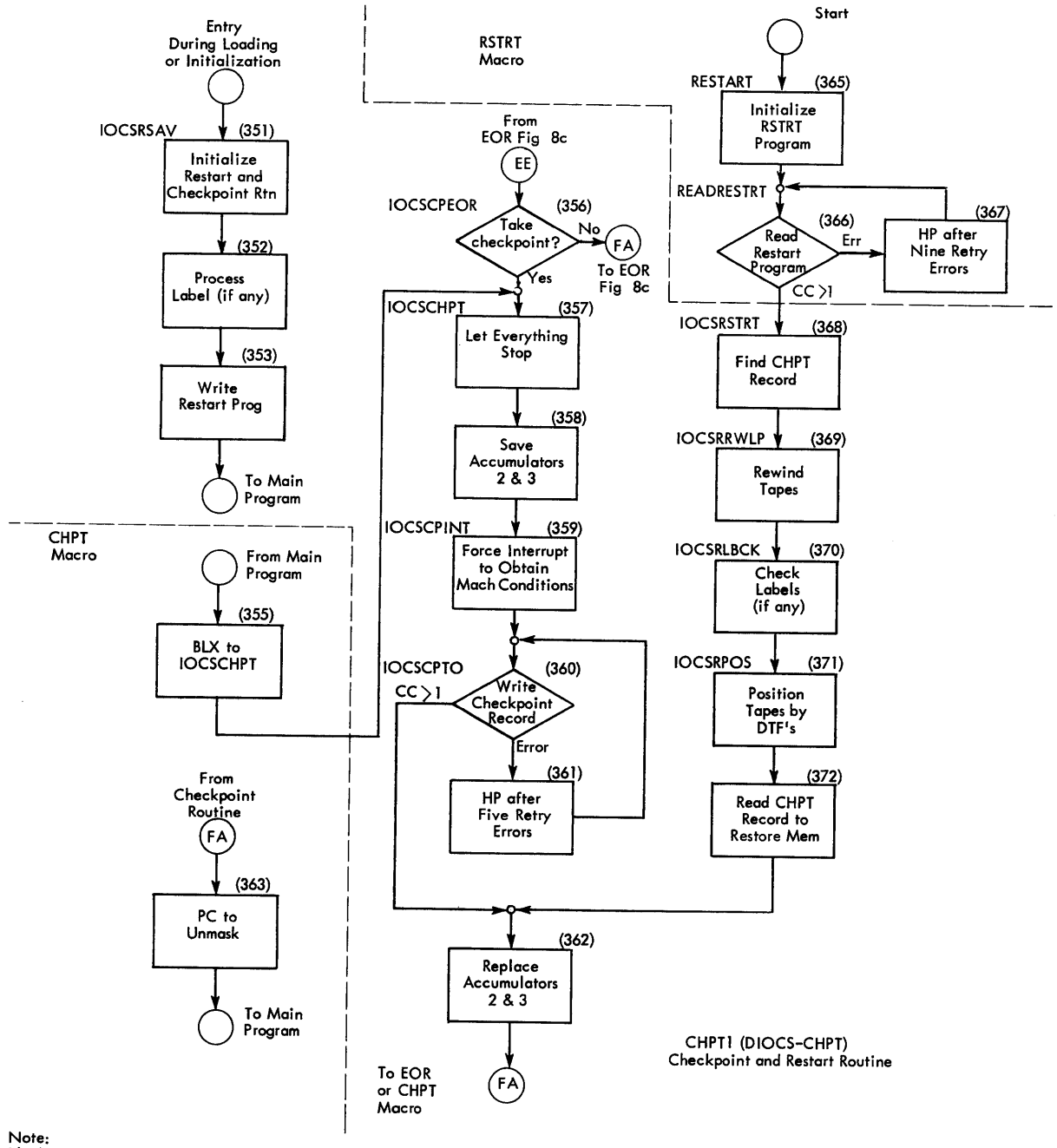


Figure 10. End and Close Routine



Note:
Blocks 351-353 and 368-372 may be overlaid with programming and may not be found in storage during the running of the program.
Blocks 365-367 are read into storage during a restart and bring the restart program into memory (Blocks 368-372). Blocks 365-367 will be overlaid when memory is restored.

Figure 11. Checkpoint and Restart Routines

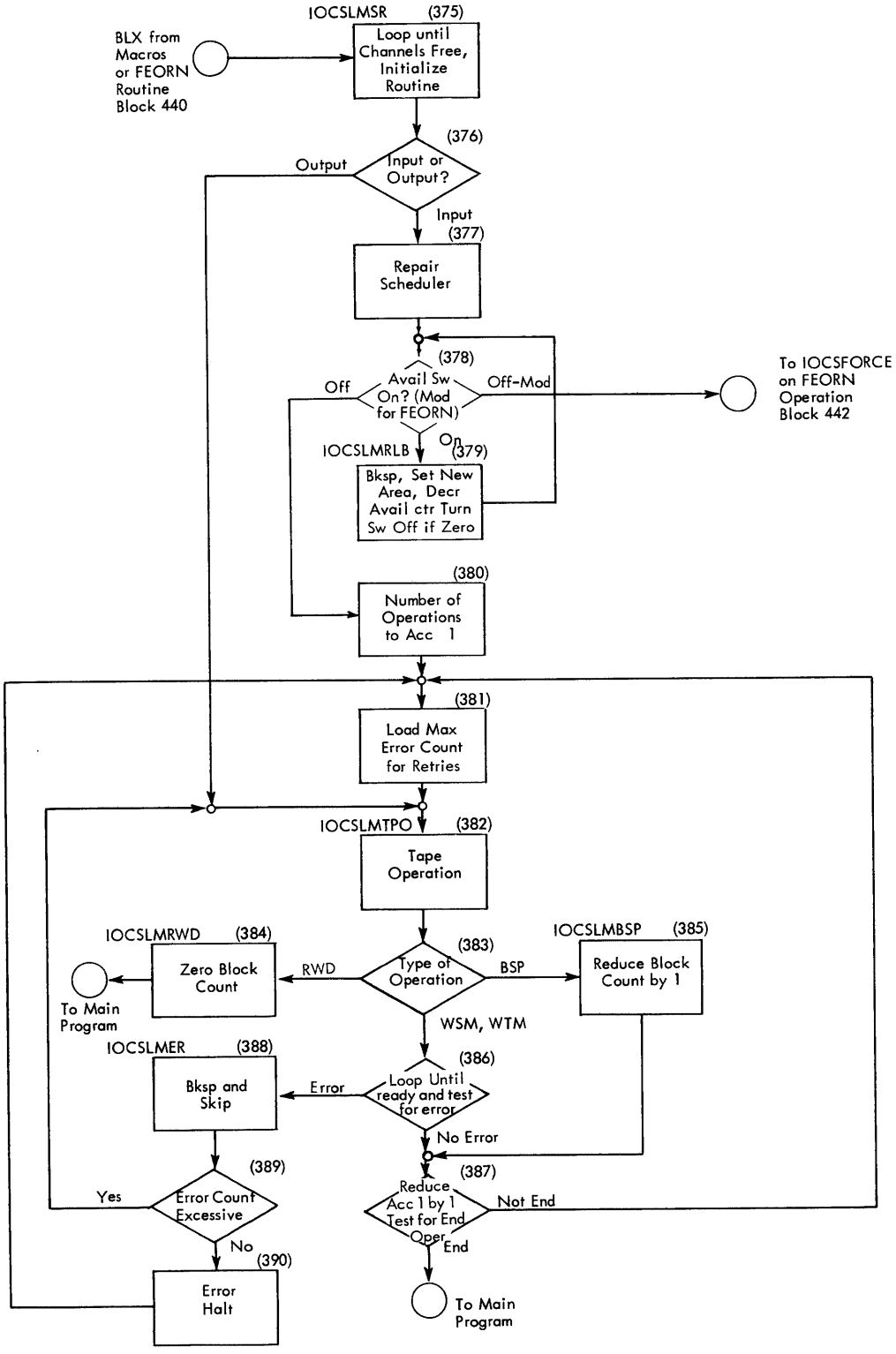


Figure 12. BSP, RWD, WSM and WTM Macros

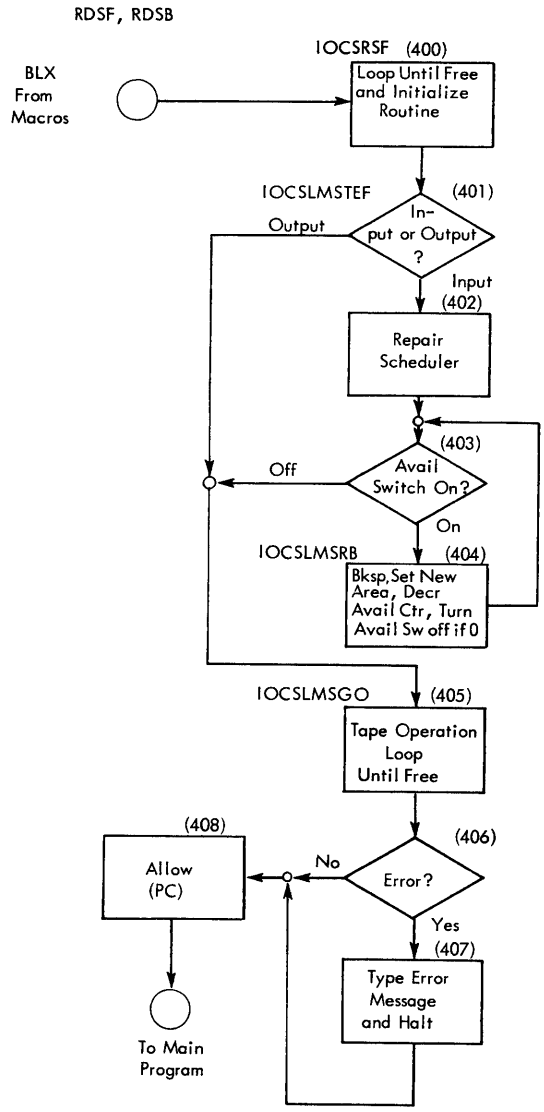


Figure 13. RDSF and RDSB Macros

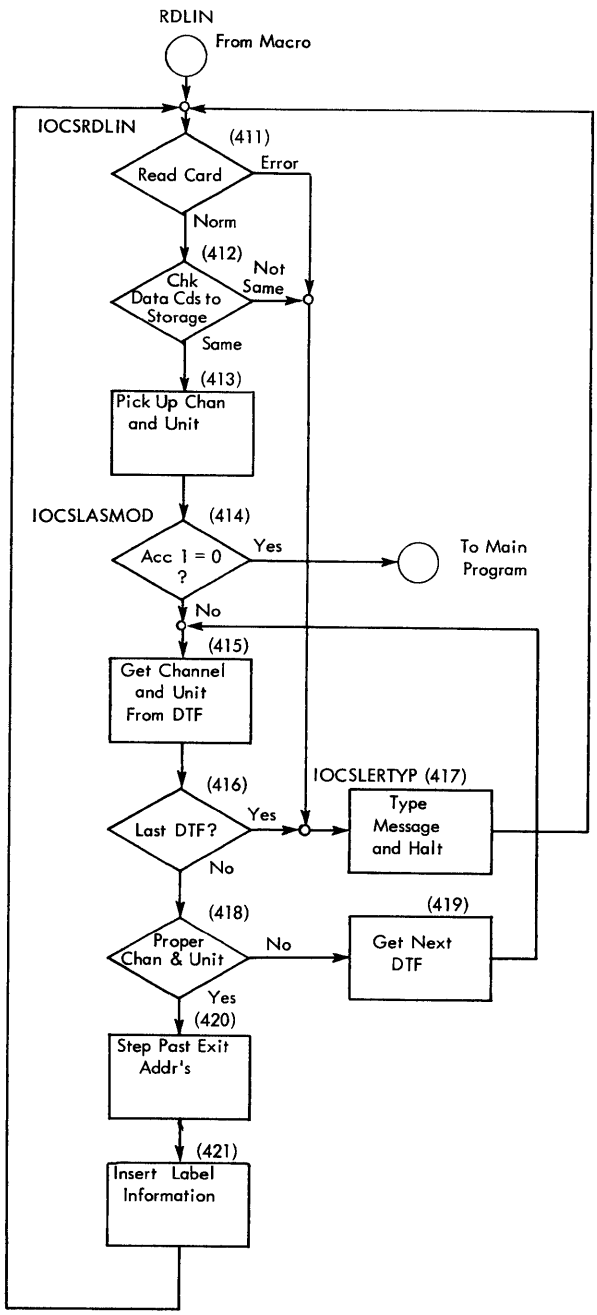


Figure 14. RDLIN Macro

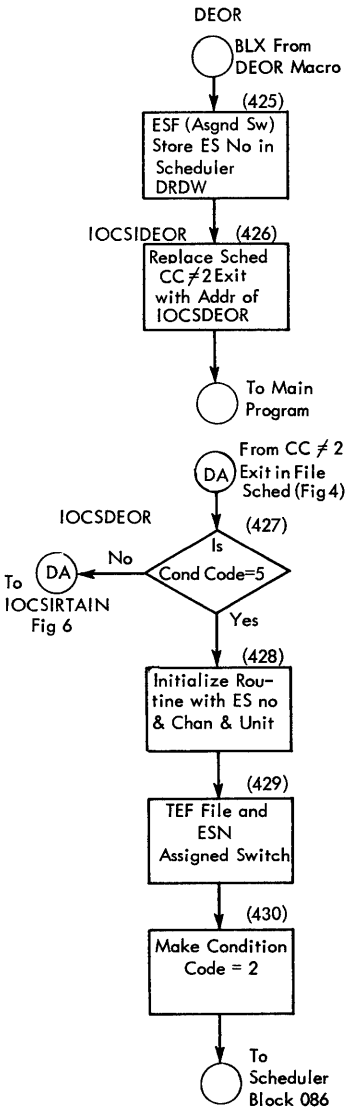


Figure 15. DEOR Macro

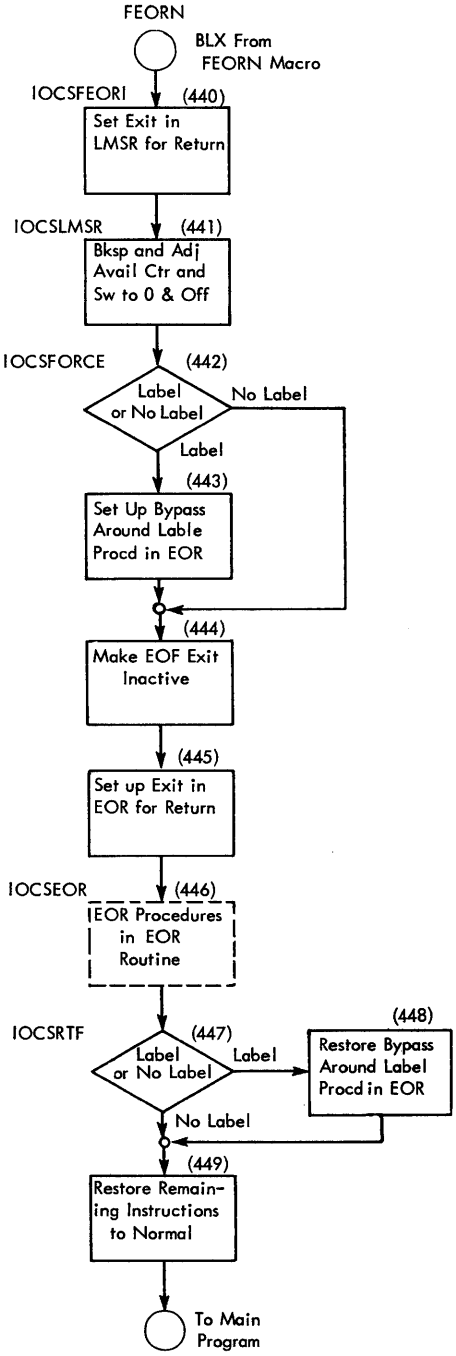


Figure 16. FEORN Macro

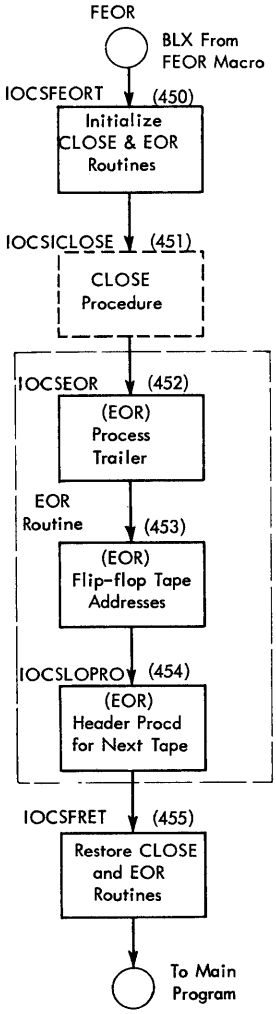
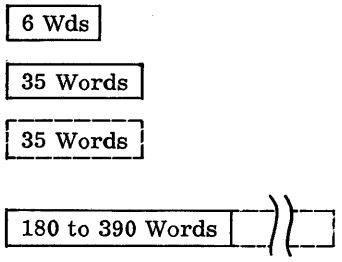


Figure 17. FEOR Macro

Memory Locations

Storage Map

These routines will be in consecutive positions of storage with the exceptions of the sections in dotted lines under given conditions. Their length and option will be determined by specifications in the DIOCS entry.



@ IOCSIGEN = General Scheduler

@ IOCSC1S = Channel Scheduler 1

@ IOCSC2S = Channel Scheduler 2 if two channels are specified on assembly.

@ IOCSEOR = End-of-reel routine. Smaller version will be used if no labels are specified on assembly of program.

@ IOCSCPEOR = Checkpoint and restart routine. The checkpoint portion (first part) will be in storage at the option of the programmer at assembly. The restart portion is overlaid in storage after it is placed on the checkpoint tape as the first record.

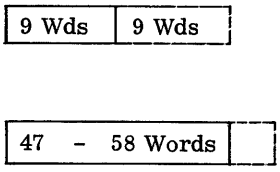
@ IOCSIOPEN = OPEN routine. The programmer has the option of using it and overlaying other instructions over it after usage. It may be placed on tape for re-use.

@ IOCSIRTAIN = Condition code routine.

@ IOCSERROR = Error routine. Smaller version will be used if invalid alphameric characters are not to be checked, as specified on assembly of program.

@ IOCSIEND = END and CLOSE routine.

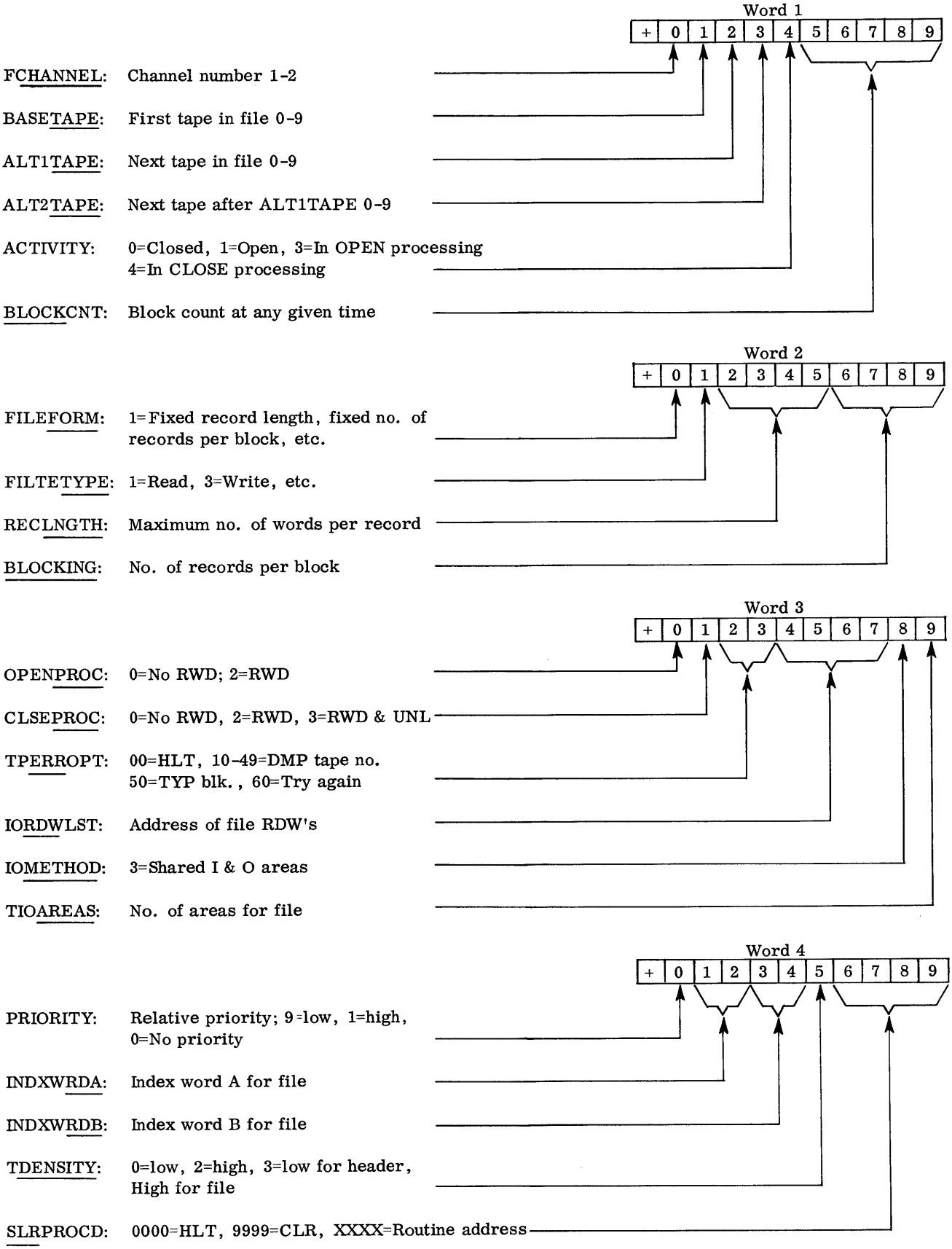
The following section of storage is located by the programmer at assembly. It may be anywhere in storage provided that it precedes the main program during assembly.

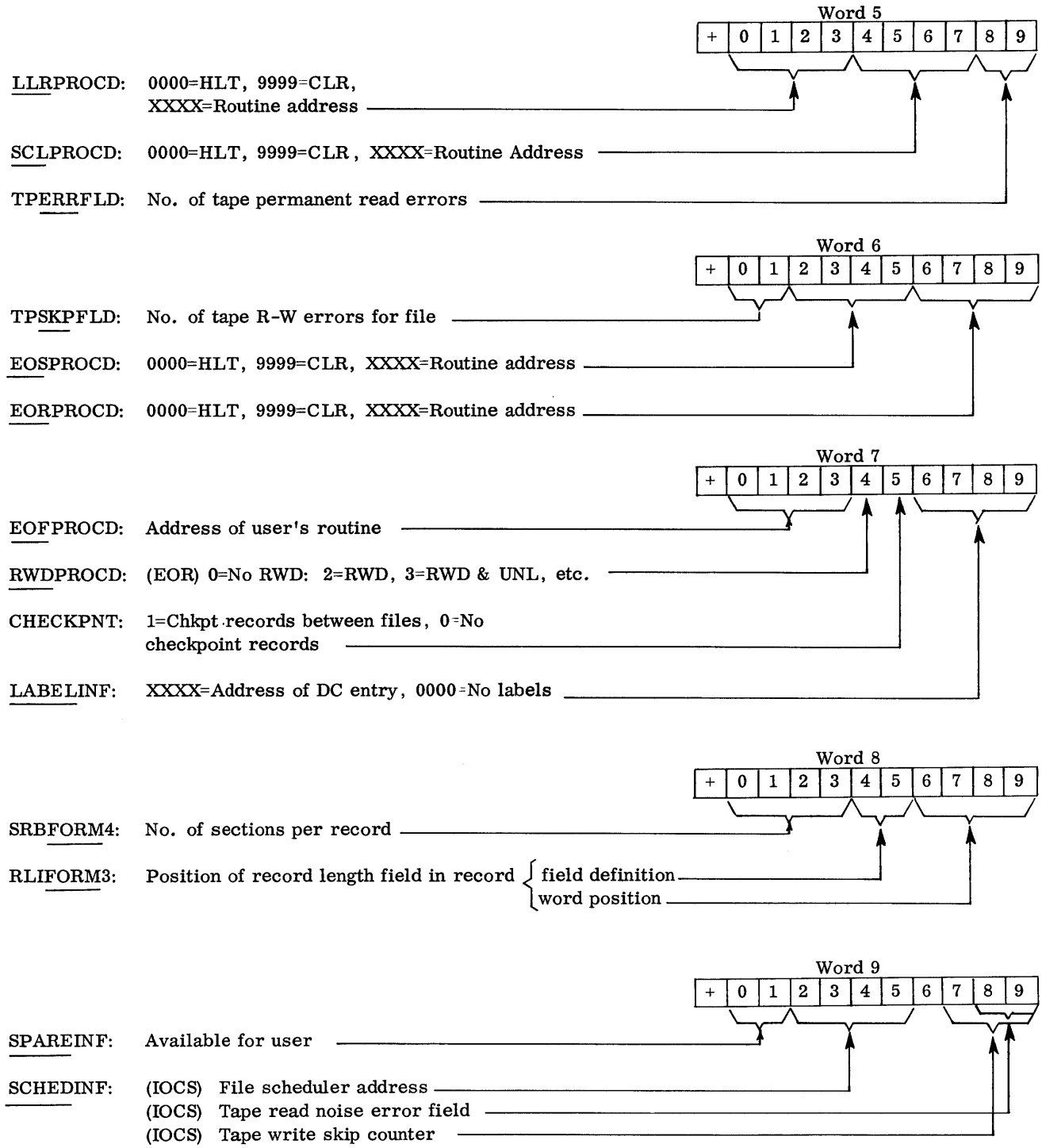


@ TAPEFILExx = DTF for file (specifications). There will be one nine-word area for each file needed in the program. The DTF's must be in sequence.

@ IOCSSCED01 = First file scheduler. Its length is determined by the DTF specifications at assembly such as FILEFORM, FILETYPE, and TIOAREAS. There will be one file scheduler for each DTF in storage. Each scheduler's length will be between 47 and 58 words.

DTF Contents




Word 5
+ 0 1 2 3 4 5 6 7 8 9
LLRPROCD: 0000=HLT, 9999=CLR, XXXX=Routine address
SCLPROCD: 0000=HLT, 9999=CLR, XXXX=Routine Address
TPERRFLD: No. of tape permanent read errors
Word 6
+ 0 1 2 3 4 5 6 7 8 9
TPSKPFLD: No. of tape R-W errors for file
EOSPROCD: 0000=HLT, 9999=CLR, XXXX=Routine address
EORPROCD: 0000=HLT, 9999=CLR, XXXX=Routine address
Word 7
+ 0 1 2 3 4 5 6 7 8 9
EOFPROCD: Address of user's routine
RWDPROCD: (EOR) 0=No RWD: 2=RWD, 3=RWD & UNL, etc.
CHECKPNT: 1=Chkpt records between files, 0=No checkpoint records
LABELINF: XXXX=Address of DC entry, 0000=No labels
Word 8
+ 0 1 2 3 4 5 6 7 8 9
SRBFORM4: No. of sections per record
RLIFORM3: Position of record length field in record
field definition
word position
Word 9
+ 0 1 2 3 4 5 6 7 8 9
SPAREINF: Available for user
SCHEDINF: (IOCS) File scheduler address
(IOCS) Tape read noise error field
(IOCS) Tape write skip counter

Special Operational Situations

This section will help to analyze a machine or program malfunction more quickly. For the benefit of the user, an index giving the actual storage locations of the routines and main sections of the IOCS program is generated in the listing. It may be found following the CLOSE routine.

If more than one file uses the same priority (DTF priority) IOCS will not process properly. The last file opened having the duplicate priority will be the only file with this priority to operate correctly. GET or PUT operations on the remaining files cause processing to hang in the channel scheduler "force" loop. This situation occurs because the OPEN routine places into storage location 015n (n is DTF's PRIORITY), the file scheduler entry address for the file. Opening a file overlays the scheduler address of a previously opened file having the same priority. When this occurs, an interrupt for this file branches to the wrong file scheduler.

If the 7070 were to "hang up," leaving a channel busy with no tape selected, it might be difficult to determine which tape unit was in use at that time. With IOCS using multiple-file operation, this information might be determined by comparing the initial and final status words for tape units on the busy channel. If a tape unit had not completed its operation, the RDW address in positions 6-9 of the initial status word would not be for the same area as that in positions 6-9 of the final status word. This comparison might determine which tape unit had been operating at the time of the machine hang-up.

In the main program, tape operations on files handled by IOCS should be accomplished by the use of IOCS little macros (WSM, BSP, etc.). IOCS is operating ahead of the main program for input files and behind on output files. For this reason, adjustments must be made internally in IOCS for unexpected tape operation. The little macros perform these adjustments as well as the operation wanted.

Index Words

IOCSIXF: Located in the IOCS listing as an EQU after the DIOCS entry. It is used in priority mode primarily. It will also hold the address of the DTF word which is being operated on, during OPEN, END, and CLOSE operations.

IOCSIXG: Located in the IOCS listing as an EQU after the DIOCS entry. It is used in normal mode primarily. It also holds the DRDW (pending and availability switch address) of the file scheduler being operated on during OPEN, and the availability switch address during END and CLOSE operation.

IOCSIXH: Located in the IOCS listing as an EQU after the DIOCS entry. It is used during a GET or PUT with unit record files.

X97: Holds the file scheduler entry address during OPEN, and the DTF address during EOR operations.

X98: Holds the pending switch address in 2-5 and the availability switch address in 6-9 during OPEN.

X99: Holds the availability switch address in 2-5 and the pending switch address in 6-9 during OPEN.

INDXWRDA and INDXWRDB: These words hold information pertaining to a tape file (two words for each file). Their location may be determined from the DTF for the file, word 4, positions 1 and 2 for INDXWRDA, and positions 3 and 4 for INDXWRDB. In condition code 3(SLR), the non-indexing portion of INDXWRDA contains the location of the first RDW, and INDXWRDB contains the location of the last RDW used as the program branches to the user's routine as determined by the DTF entry (SLRPROCD).

DTF Fields

ACTIVITY: DTF word 1, position 4 contains 0 if the file is closed, contains 3 if the file is in the process of being opened, contains 1 if the file has been opened, or is in the process of being closed by an END operation, and contains 4 if it is in the process of being closed by a CLOSE operation.

BLOCKCNT: DTF word 1, positions 5-9 contains the block count for the file at any given time. It is zeroed during the OPEN and EOR operation.

TPERRFLD: DTF word 5, positions 8-9 contains the number of permanent read errors (records which would not read correctly after ten tries) at any given time.

TPSKPFLD: DTF word 6, positions 0-1 contains the number of read or write errors for the file at any given time. When the count reaches 30 for an output file or 50 for an input file, the counter is zeroed and a statistical type-out is made. DTF word 9, positions 7-9 contains the number of write skips at any given time for an output file. DTF word 9, positions 8-9 contains the read noise count at any given time for an input file.

File Scheduler

The following switches and counters will be found in the file schedulers. All locations are given with respect to the labels in the schedulers. The last two characters in the label, stated as xx, are the file number.

Availability Switch, IOCSAVSWXX: On = B, Off = NOP.

Availability Counter, IOCSAVSWXX, Position 5: Contains the number of areas available (0-3).

First Availability Subswitch (area switch), IOCSAVSWXX+5: On = B, Off = NOP.

Second Availability Subswitch (area switch), IOCSAVSWXX+8: On = B, Off = NOP.

Pending Switch, IOCSSCEDXX+9 (except on form 3, two-area, input which is located at IOCSSCEDXX+12): Off = B, On = NOP.

Pending Counter, located at pending switch address, position 5: Contains the number of areas pending (0-3).

First Pending Subswitch (area switch), IOCSSCEDXX+10 (except on form 3, two-area, input which is located at IOCSSCEDXX+12): On = B, Off = NOP.

Second Pending Subswitch (area switch), IOCSSCEDXX+13 (except on form 3, two-area, input which is located at IOCSSCEDXX+16): On = B, Off = NOP.

Channel Scheduler

The following switches are located in the channel scheduler. Their locations are given with respect to the label of the first instruction of the scheduler, IOCSCYS (y being the channel number).

Bypass Switch, IOCSCYS+10: On = NOP, Off = B.

Cross Switch, IOCSCYS+27: On = B, Off = NOP.

Force Switch, IOCSCYS+3: On = NOP, Off = B.

Free Switch, IOCSCYS+23: On = NOP, Off = B.

SPOOL Switch, IOCSCYS+2: On = electronic switch 29 or 30 on, Off = electronic switch 29 or 30 off. (Electronic switch 29 is used for channel 2, 30 for channel 1.)

Holding Loops

Channel Scheduler: A loop is IOCSCYS+15, 16, and 17, consisting of a CSP, BE, and B to the CSP that loops until the CSP detects that the availability switch was turned on for this file. This loop is entered on a force operation, indicating that the main program cannot proceed until an area is made available to it for operation after a GET or PUT.

Condition Code Routine: Two waiting loops are included to allow all pending operations for the file to be completed. This permits previously processed records to be handled by IOCS before an operation such as a condition code 3-7 can be performed for a specific file. One loop that tests the pending switch is located at IOCSIPSLO and is used for output files. The other loop, which tests the availability counter, is at IOCSICHECK and is used for input files.

SPOOL: A branch-to-itself loop in the CLOSE routine used to keep the machine in run status so that SPOOL operations will continue after end of job if the programmer desires. This loop is located at IOCSEJLOOP.

Temporary Storage Words

Located in the OPEN *routine:*

IOCSHACC2: Holds accumulator 2.
IOCSHACC3: Holds accumulator 3.
IOCSHIX98: Holds index word 98.

Located in the general scheduler:

IOCSIGEN+3: Holds accumulator 1.

Appendix

Glossary

AVAILABILITY COUNTER, FILE SCHEDULER: This counter is contained in digit position 5 of the availability switch. For an input file, it indicates the number of areas that have been read into and are ready for use. For an output file, it indicates the number of areas that have been written on tape and are ready to be used. When it is zeroed, the availability switch is set off.

AVAILABILITY SWITCH, FILE SCHEDULER: B is on, NOP is off. This switch is set on whenever at least one area is ready for processing, (i.e., whenever at least one area is available). It is turned off when the availability counter is zeroed.

BLOCK: One or more data records grouped to form one continuous record which will be written or read from tape, from or to storage.

BYPASS SWITCH, CHANNEL SCHEDULER: NOP is on, B is off. Part of the logic of the schedulers is the same for both the normal and priority mode; this switch differentiates between the modes when it is necessary to separate the logic, by being turned on and off only in the normal mode.

CLR (Correct Length Record): Condition code 2 after completion of a tape operation.

CROSS SWITCH, CHANNEL SCHEDULER: B is on, NOP is off. If a new area is not immediately available for processing, the cross switch causes the force routine branch to be set in the force switch so that the needed area will be "forced" to be made available at the interrupt after the tape operation.

DA (Define Area): The declarative operation used to reserve and define any portion of storage as a data area.

DATA RECORD: A number of words of information grouped in a known manner which will be used as data for a given operation.

DC (Define Constant): The declarative operation used to enter numeric, alphameric, and address constants into the object program by an assembly.

DTF (Define Tape File): Nine words containing all information pertinent to a given tape file. This information is in a given order so that it may be referred to by IOCS.

DUF (Define Unit-Record File): A source language statement which, on assembly, causes generation of instructions that will perform a unit record operation for the object program.

EOF (End of File): The logical end of an organized collection of information directed toward some purpose. For multiple-reel files, it is recognized at the end of the last reel.

EOR (End of Reel): The end of all records on a single tape. The trailer on labeled input tapes contains information defining end of reel. EOR on unlabeled input tapes must be recognized by a user's routine. The user is able to identify an end of reel by recognizing the last record of a reel. EOR for output files is normally indicated by recognition of the end-of-tape reflector.

EOS (End of Segment): Recognition of a segment mark on input files, producing condition code 6 in the final status word for that file.

EQU (Equate): The declarative operation used to identify a symbol to an actual address, a symbolic address, a component (channel, tape unit, etc.), an index word, or an electronic switch, during assembly.

FORCE SWITCH, CHANNEL SCHEDULER: NOP is on, B is off. If an area is needed immediately, this switch is set to branch to the scheduler of the input or output file in question (via the pending switch) and will cause, or force, an area to be available.

FREE SWITCH, CHANNEL SCHEDULER: B is off, NOP is on. This switch is set off (busy) in the channel scheduler if a tape operation is started. It is set on (free) only when no areas are waiting for a tape input or output operation, i.e., when no pending switches are on, after an interrupt.

IOCS (Input-Output Control System): A program developed to handle all necessary unit record or tape input and output procedures to relieve programmers of duplicating their efforts for most programs they write.

IXA, IXB (INDXWRDA, INDXWRDB): Also equated to IOCSFIXA and IOCSFIXB. Index words assigned to each DTF for the use of IOCS and the user's program to manipulate records.

LABEL: A record or records written on tape containing identifying information concerning the file on the tape. For specifications, see IOCS Bulletin.

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 statements. 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.

PENDING COUNTER, FILE SCHEDULER: This counter is located at position 5 of the pending switch and is incremented each time an area becomes ready for a tape operation. For input files, it indicates the number of areas that have been processed and are waiting to be read into; for output files it indicates the number of areas that have been filled with processed records and not yet written on tape. This counter is decremented each time a tape operation for the file is started. When zeroed, the pending switch is set off.

PENDING SWITCH, FILE SCHEDULER: NOP is on, B is off. This switch is set on whenever the file scheduler is entered in the normal mode. It indicates that at least one area of a file has been processed and is now ready for a tape operation.

RLSE (Release): A macro which ends usage of the block presently being used and makes the next area available to the main program.

SCLR (Short Character Length Record): A condition which occurs when the number of alphameric characters read in a record is not an even multiple of five.

SLR (Short Length Record): A condition which occurs when a record being read is shorter than the read-in area described by the RDW's of the operation.

SPOOL SWITCH, CHANNEL SCHEDULER: This switch tests one of the SPOOL electronic switches (i.e., 29 or 30) to determine whether to branch to the SPOOL program or continue in the main program.

Abbreviations

Acc	accumulator
Addr	address
Adj	adjust
Alt	alternate
Asgned	assigned
Avail	availability
Blk	block
Blks	blocks
Blkcnt	block-count
Bksp	backspace
Chan	channel
Chk	check
Chkpt	checkpoint
Cond	condition
Ctr	counter
Decr	decrement
Err	error
ES	electronic switch
Fig	Figure
Fld	field
Hdr	header
Incr	increment
Info	information
Init	initialize
Inst	instruction
Loc	location
LP	load point
Msg	message
No	number
Op	operation
Pndg	pending
Pos	position
Procd	procedure
Prog	program
Rcd	record
Rd	read
Rtn	routine
Rwd	rewind
Sched	scheduler
Spec	specify
Sub	subtract
Sw	switch
Trlr	trailer
Via	by way of
Wd	word
Wr	write

date

(9/61: 5M-FW-60)

IBM 7070 IOCS Printed in U.S.A. C28-6119

IBM

International Business Machines Corporation

Data Processing Division

112 East Post Road, White Plains, New York