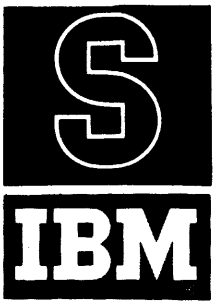


7090 DATA PROCESSING SYSTEM BULLETIN

IBM 7090 PROGRAMMING SYSTEMS
SHARE 7090 9PAC
PART 2: THE FILE PROCESSOR

This is the second part of a reference manual which describes the SHARE 7090 9PAC System; this publication describes the File Processor portion of 9PAC. Other parts of the manual are:

SHARE 7090 9PAC
Part 1: Introduction and General Principles
Form J28-6166

SHARE 7090 9PAC
Part 3: The Reports Generator
Form J28-6168

References in this publication to the other parts of the manual are in terms of part and chapter numbers. An understanding of Part 1 is required to use this part.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

The File Processor exists for two major reasons: file establishment and file maintenance. File establishment consists of the steps required for the original creation of a master file and includes the assembling of information from various sources. File maintenance consists of the activities involved in changing the master file in order to update it and may include making actual changes to the file (insertions, deletions, and modifications), changing the format of the master file, and using handwritten calculations to provide for complex operations.

File Processor Activities

Within the framework of the system there are several activities that control file processing; these are:

1. Selection. This is the determination of whether the prerequisite conditions for making a change are met; it is done by testing either the master file record or the change file record, depending upon the type of change. For example, the programmer may specify that a particular location must contain a 7 for the change to be made.

2. Matching. This is the determination of whether the master file record and the change record sequencing fields are either identical (for modifications or deletions) or are properly related (for insertions). For example, after a change file record has been selected it is necessary to determine that it applies to the current master record, namely, both the change file record and the master file record apply to part number 31314.

3. Processing. This is the actual operation upon the master file data as a result of the determination that it is to be affected by the change. For example, a field of the change record may be added to a field of the master file record.

Types of File Processor Files

To better understand the discussion of the File Processor, at this point the various possible File Processor input/output files will be listed.

Old Master File

This file does not, of course, exist when a file is being established; however, it is always part of the input for file maintenance. This file must have an attached dictionary; all other characteristics are variable (these are described in Part 1, Chapter 2).

New Master File

This may be a newly established file or an updated master file; this file is always part of the output from the File Processor. All of the file characteristics are variable unless the file is to be used subsequently as an old master file, in which case it must have an attached dictionary.

Change Data Input Files

At least one change data input file (change file) is always present for file establishment or file maintenance. It is possible to have several change data input files, in which case they are automatically merged on the basis of sequencing information. The characteristics of this type of file are variable.

Change Report File

This file is optional and may be specified by the programmer to contain all or any of the changes that have been made to the master file. This file must undergo processing by the Reports Generator to produce a meaningful report. The File Processor automatically attaches a dictionary to this type of file; all other characteristics are variable.

System Data Errors

This file may be output as a result of either file establishment or file maintenance. It consists of detected errors, which may be rejected records, sequence errors, etc. The File Processor automatically attaches a dictionary to this type of file; all other characteristics are variable. This file must undergo processing by the Reports Generator to produce a meaningful report.

Note that both the change reports and the system data errors are written on the same tape and their characteristics are specified by the *FILE card for file number 08.

Summary of the Use of File Processor Files

Whenever a File Processor job is for the purpose of file maintenance, there will be an old master file as input and there will be a new master file as output. The instructions which will be followed to produce a new master file will be contained in a parameter deck comprised of packets describing the individual changes. In addition, there will be one or more input files whose function is to provide change data to be incorporated into the new master file. As output, in addition to the new master file, there will be a comprehensive list of error conditions and a list of change reports. The error conditions and the change reports subsequently will be processed by the Reports Generator to indicate in detail what has been accomplished and what has failed to be accomplished due to errors during a given processing cycle. The error conditions that will be recorded are listed later.

Whenever a File Processor job is for the purpose of file establishment, a dummy old master file is created during generation; the description above then applies to file establishment as well as to file maintenance.

CHAPTER 2: BASIC CONCEPTS

File Processor Parameter Decks

A parameter deck is required by the File Processor for the generation of the object program which will actually control the file processing. The parameter deck consists of an *JOB card, several *FILE cards, an *END card, dictionary definition packets (if required), and change packets.

The parameter deck contains the only instructions that the File Processor generator will need to produce an object program. The function served by the packets is to provide a description of the data input, to specify the action to be taken in incorporating the change data into the master file, and to specify any other changes which are to be made to the master file.

Priority Numbers

Use of the File Processor involves the use of several different coding forms. A given problem may use any combinations of these coding forms. The cards which are punched from one coding form are a single packet and will make up a logical section of the object program. Each card has both a major priority number and a minor priority number. Major priority numbers are always the same for each card of a given packet and are used for defining and sequencing the packets. Minor priority numbers are for programmer convenience in sequencing cards within the major priority packets; they are not checked by the File Processor.

Thus, the first packet will have a major priority number and each card of that packet will carry the same major priority number; they may, if desired, also carry a minor priority number. The cards of the second packet will have a higher major priority number than the cards of the first packet. The assignment of priority numbers is the programmer's responsibility, but major priority numbers must:

1. Be two numeric digits.

2. Be in increasing order from the first to last packet. (It is suggested that gaps be left in the numbering sequence for later insertion of additional packets.)

3. Be consistent with the order of input required by the File Processor; that is, dictionary packets must precede all change packets, and dictionary packets must be ordered on the basis of increasing record type.

Packets are executed in the order of major priority number (from lowest to highest) when they affect the same master file record. (When change packets affect different master file records, the priority numbers of the change packets have no significant effect.)

Dictionary Packets

Dictionary packets are prepared from the Dictionary Definition form, which may be used to specify the complete dictionary for file establishment or to specify changes to the dictionary during file maintenance. Since the dictionary controls the actual output format, the Dictionary Definition form must be included in the File Processor job deck for file establishment even if the output file is not to have an attached dictionary.

A separate Dictionary Definition form must be prepared for each separate record type being defined for file establishment or being changed during file maintenance. Dictionary packets used during file maintenance serve to modify the format of the existing file records or to insert or delete additional record types.

The programmer is cautioned that references to the data in other packets of the program deck may also need to be changed when the dictionary is changed.

The cards punched from this form (i.e., the dictionary packets) must precede all other packets and must be in order of increasing record type.

Change Packets

The File Processor provides for three types of change packets: vertical, update, and horizontal.

Vertical Change

A vertical change is a constant change which is made to all records of a given record type which meet the selection criteria (selection is based on information in the master file record). A single vertical change packet may cause action to be taken on one or more fields in a selected record.

Since a vertical change is made consistently to all records of the same record type, all change information is included on the form and no change file is associated with the vertical change.

A sumreset change is a special type of vertical change which controls the clearing or resetting of summaries and summary records. The sumreset function is used when, during file maintenance, it is necessary to clear previous totals and use the cleared fields for the accumulation of a new set of totals summing up the updated file. All sumreset packets are executed, in order of major priority number, before any vertical, update, or horizontal change packets which affect the same master file record type are executed. Sumreset packets may not contain change report requests or hand calculations (discussed later).

Update Change

An update change consists of performing the same type of action on selected records (selection is based on information in the change file record). The change data is contained in records on a change file tape. The priority packet identifies the selection criteria, the file number of the change file,

the fields which must be matched before performing the change, the type of action, the change data field to be used from the change record, and the field to be updated in the master file. The separate change tape contains the data to be used for matching, selection, and processing. One or more records on the change tape may be matched and processed with one record on the master file tape.

Each change file record must contain all of the sequence fields contained in the master file record or in any parent record of the master file record.

Since the change data is in a known format, such that a given field will always be located in the same place within a change record, and since the type of action is the same for each field, updating provides an economical method of file maintenance in that a fixed pattern of instructions may be generated to process this type of change.

The update change file may or may not have a dictionary. If the change file does not contain a dictionary, the file must be in BCD representation but may be blocked or unblocked.

If unblocked, the maximum record length must be stated. If blocked, all logical records must be fixed length and the blocks must be of equal length (with the possible exception of the last block). Terminating padding characters may be used, if desired.

The formats of the records themselves may vary as long as they are described individually by the various major priority packets sharing the change input tape.

All data fields in the update change file without dictionary are referred to by increment and length.

If the change file does have a dictionary, the file characteristics are variable and data fields are referenced by record type and field number.

Horizontal Change

A horizontal change, unlike a vertical or update change, does not always involve the same change action or even the same master record field. Each horizontal change data record specifies the master record field to be changed and may also specify the change action to occur. This type of change requires the use of information from the master file dictionary. Since the master file dictionary is not present at object time, the horizontal change data must be preprocessed to a form which may be utilized at object time. This preprocessing becomes the first phase of a two-phase object program; it is automatic and need not concern the programmer.

The priority packet specifies the change file number, the selection criteria (selection is based on information in the change file record), and the location of the fields in both files to be matched prior to the execution of the change. The change tape contains, in addition to the data needed for matching, the field number of the field to be changed, the actual change value, and may contain the type of change action.

A horizontal change packet describes an unblocked BCD file of change records which are, in some manner, to be incorporated into the master file. For each master file record type to be changed, there must be a separate horizontal change packet. Each change record is related to a particular horizontal change packet by means of the selection process. Only one horizontal change file is permitted for a single File Processor job. Each change file record must contain all of the sequence fields contained in the master file record or in any parent record of the master file record.

There are two types of horizontal change file records; both contain the necessary matching and sequence fields and, in addition:

1. The first type contains field-value pairs; pairs specify both the master file field to be changed and the change value to be used. This type of change is used when the field action is the same throughout the packet and is indicated in the packet as:

R	Replace master file field with change value
+	Add change value to master file field
-	Subtract change value from master file field

2. The second type contains field-action-value triplets; triplets specify the master file field to be changed, the action to be taken with respect to the master file field (using the change value), and the change value.

   Thus, this type of change allows the field action to vary with each triplet. The allowable actions for this type of change are the same as for field-value pairs. The actual field action specified in the change packet itself is P (Programmed).

More than one change may be made with one change card. In this case, no punctuation or spaces are allowed between the pairs or triplets. The format of the change card is:

1. Matching and sequencing information (this may be arranged in any manner but must precede items 2, 3, and 4 below).

2. Four columns for master file field number to be affected by this change.

3. For triplets only, one column for field action; this may be R, +, or -.

4. n columns for change value, where n is the length of the master file field as defined in the dictionary.

Auxiliary Functions

In addition to making changes of the types listed above, the File Processor provides for including hand calculations and requests for change reports in the object program.

Hand Calculations

Occasionally a computation or procedure is desirable which may not be stated in terms of the standard File Processor coding forms. For these situations, the File Processor permits the inclusion of certain macro-instructions and symbolic machine instructions; these are described in a subsequent chapter.

Change Reports

A file may be written which contains every change or every nth change of any specified types. This change report file usually must be processed by the Reports Generator to produce a report which will summarize the changes or check the accuracy of the change procedure. When changes are being made to relatively few records in the file, it may be desirable to write a change report file which includes all changes made to the master file, thus providing a condensed input to the Reports Generator. When changes are made to the master file, it may be desirable to obtain a report of the changes made in order to insure the correctness of the master file after change. However, since the content of the change report varies depending upon the data involved, no attempt has been made to prepare change reports of fixed format and content. Instead, a special form called the Change Report Field Request form is provided so that the programmer may request specified fields from the master file, the change file, and (for horizontal changes only) the master file dictionary, to be written onto the change report output tape. Fields from the master file record being processed, or from any of its parent records, may also be requested.

A separate Change Report Field Request card is prepared for each field of any record associated with the priority packet for which change report records are to be created; these cards will be punched and appended to the priority packets and must have the same major priority numbers. Change report records will be written only if the priority packets contain these cards. The change report records for all priority packets are written onto a single output unit. These records will constitute a file with dictionary (automatically created); the record types will be the same as the major priority number of the packets in which they are created, with minus zoning over the first digit. The change report record consists of the various fields specified on the Change Report Field Request cards. These fields contain the data in the field before and/or after the change as follows:

1. If field action is Insert, the record is written after change.

2. If field action is Delete, the record is written before change.

3. If field action is Modify, the record is written before and after change.

4. If a T is entered in the leftmost digit of a Frequency of Change Report field on an update or vertical change form, only the "after" change record is written.

An exception to the above occurs for horizontal changes, in which case only a single record is written and it may consist of both the "before" and "after" field value, as specified by the programmer.

The following information will be useful in making Reports Generator or future File Processor references to the change report file:

1. The first field of each change report record will contain the record type; this field is entered in the change report file dictionary with field number 0001.

2. The second field of vertical and update change report records contains one of the following record action codes: I, D, E, O, F, or T; F and T represent FROM and TO when M action (or O code resulting in M action) has been specified. The record action codes will be described later. This field is entered in the change report file dictionary with field number ACTR.

Error Reports

Certain error conditions and irregularities may be sensed during the execution of the File Processor object program either by validity tests built into each program or by editing which is built into hand calculations.

In case such an error or irregularity is sensed, an error will be written on the change report output unit. The record type of an error record will be the major priority number of the packet in which the error was detected, with plus zoning added to its leading character.

Like change report records, each type of error record will have a dictionary. Following is a description of the error record format, which will be useful in making Reports Generator references to the error records:

Field Number	Description
0001	Record type.
0002	Field number of master file field upon which rejection was based. This is filled in if rejection was due to failure of the minimum-maximum test, or improper field action on a "replace only" field, or was due to the ERRFLD macro-instruction. This is set to zero if rejection was based upon a change file sequence error or failure to match with master file identification.
0003	The reason for the rejection in a 12 BCD character description. If the rejection was caused by a REJECT or ERRFLD macro-instruction, the reason will come from the address field of the macro-instruction.
0004	The value that a field would have had if action had not been stopped by failure to meet a minimum-maximum test. If this field is greater than 6 characters, only the six high-order digits will be given.
0005	The entire rejected change record, except for vertical changes. For vertical changes, the master file identification is provided, since there is no change record.

An error record will be written for any of the following reasons:

Error	Description
UNMATCHED	Modifying or deleting a non-existent record.
SEQ ERROR	Horizontal or update change records are out of sequence on the change input unit.
INSERT EQUAL	Attempting to insert a record already present.
SF MIN TEST	Data fails to pass the minimum test on the basis of the dictionary.
SF MAX TEST	Data fails to pass the maximum test on the basis of the dictionary.
CHG MIN TEST	Data fails to pass the minimum test on the basis of the parameter packet.
CHG MAX TEST	Data fails to pass the maximum test on the basis of the parameter packet.
FLD# INVALID	Horizontal change contains invalid field number.
IS AN ID FLD	Attempting to change a sequence field with a horizontal change.
ACTF INVALID	Invalid field action specified by a horizontal triplet.
CANNOT ACCUM	Horizontal packet attempts to accumulate to a "replace only" field.
BCD-BIN ERR	Attempting to add a non-numeric BCD character in a field to a binary field.
NO SUM FIELD	Using the SUM macro-instruction in a hand calculation for a field not specified for summarization by the dictionary.
NO PARENT	Attempting to insert an offspring record for which no parent exists.

CHAPTER 3: THE FILE PROCESSOR CODING FORMS

The purpose of this chapter is to describe the function performed by each section of the coding forms which, after card punching, constitute the priority packets. The detailed parameter entries are described in a later chapter.

On all coding forms, fields are referred to by record type and field number if the file contains a dictionary; otherwise, by increment and length. Increment is defined to be the number of characters or bits within the record which precede the described field; e.g., the increment of the first character is 0. Length is the number of characters or bits comprising the field. If the character B is placed in column 20 of the *JOB card, all references to increments, lengths, or sizes must be in bits; if any other character, or a blank, appears in column 20, the dimensions must be in characters (or card columns).

The Dictionary Definition Form

The following discussion explains many of the uses of the Dictionary Definition form. The usual procedure when initiating a new application is to first generate a program to establish the new master file and then generate a second program which will perform the routine file processing. At this point, a binary deck is often punched, thus eliminating the need for regeneration on subsequent runs. The parameter deck used during file establishment will consist of dictionary change packets, one for each record type, followed by vertical and update or horizontal change packets specifying the manner in which the change data will be used to build the file. The parameter deck used for file maintenance will usually consist of the same vertical, update, and horizontal change packets used during file establishment, with minor changes. The dictionary definition packets, of course, will not be used.

If, at some later time, it is desired to modify the contents of the dictionary, and thus the format of the file, it will be necessary to generate a new object program. This object program will serve the one-time purpose of modifying the dictionary and the format of the file itself.

The parameter deck used for this purpose will consist of dictionary change cards for only those records and fields for which changes are desired. These are followed by the normal vertical, update, and horizontal change packets. It will usually be necessary to make some changes in these packets, since they will now be referring to a file with a new format. All references to the data by these packets should refer to the data in its ultimate format as described by the new output master file dictionary. The conversion of each data record from the old to the new format is automatically accomplished at object program time before any other packet makes reference to it.

After a dictionary change, it will be necessary to again generate an object program without the dictionary change packets for subsequent routine processing of the file.

When establishing a new file, there must be a dictionary packet for each record type. A dictionary packet consists of a packet header, which supplies

record specifications that serve to build a dictionary E-record. Following the packet header are dictionary function cards, which supply field specifications and which serve to build dictionary F-records.

When deleting a record type, the packet consists of the packet header alone. When a parent record type is deleted, its offspring are automatically deleted from the dictionary during generation and from the master file text during processing.

When modifying or inserting a record type, the packet need contain only the packet header and cards for those fields being changed. Record format may be modified by the insertion and deletion of fields or by changing the format of existing fields. Other generic information such as parent code, mode, etc., may also be changed. The only things that cannot be changed are the record type numbers and the field numbers of existing records and fields; thus, it may be desirable to leave gaps in the numbering sequence to provide for later insertions.

It should be noted that modifications to one record type may sometimes require modifications to other record type descriptions in the dictionary and usually require modifications to related update, vertical, and horizontal packets.

Dictionary Packet Cards

A dictionary packet contains a header card, which specifies the record type, the record action code, and the parent code. The record action code specifies the type of change to be made to the dictionary. All changes made to the dictionary will subsequently be reflected in the master file after processing. The parent code is optional and will be described later.

Following the header card will normally be a series of DICT and SUMM function cards. Each DICT function card refers to a field within the record. It specifies to the File Processor the field length, mode, allowable range, field action code and, for fields which will be used in sequencing the file (see below), a sequence level.

For those binary fields that are to be summed, a SUMM function card stating the record type and field number which is to contain the sum must immediately follow the DICT function card describing the field being summed. If one field is to be summed into more than one summary field, a SUMM card must be included for each summary field. The record type containing the summary field is referred to as the summary record type. Summary record types are physically the same as any other record type and may themselves be summed into other summary records. They differ only with regard to sequencing, which will be described below.

Record Sequencing

Sequencing information must be thought of in terms of the entire file, not in terms of individual records in the file. Therefore, when making entries of sequencing level in a DICT function card for one record type, all other entries for all record types in the file must be considered.

Each record in the master file must be uniquely sequenced. To prevent equality where it might otherwise exist, record type is always suffixed to the other sequencing information.

When there are two or more record types in a file, application logic frequently requires one type to have more sequencing information than the others. This is permissible, however, the File Processor will operate, and the programmer must code, on the basis that sequence level designations for any field are relative to the file as a whole. That is, the primary sequence field must always be designated as sequence level 01, the next most significant field must always be designated as sequence level 02. The least significant sequence field (record type) must also be designated by the same sequence level designation — which will always be numerically one greater than any other sequencing level designation in the file. For any sequencing level not specified, the File Processor will assume its contents to be the lowest possible value (blanks for commercial sequence, zeros for 709 sequence).

When different types of records are logically related in a group or family relationship, the primary sequencing fields will normally contain identical information. In most cases, it is desirable to eliminate the carrying of this redundant information in the file, thus decreasing processing time. The File Processor allows the primary sequencing information to be omitted from a record (no DICT cards for such fields) if the information can be obtained from a higher level record (lower numerical record type). The record type from which the information can be obtained is termed a parent and its designation must be entered in the parent code space in the packet header. This parent record may itself obtain some of its high-order sequencing information from a higher level parent.

When different types of records do not fall into a single logical group, the sequencing data may or may not be identical, depending on the data involved. For a given level, the sequence field lengths do not need to be equal; the File Processor will pad the contents of the shorter field with low padding during the sequencing operations (not in the file itself). Thus, depending on the data, it is possible that records of different logical groups will be merged.

Placement of Summary Records

For summary record types, the File Processor will assume the highest possible value for those sequencing levels between the lowest level specified for the summary record and the lowest level specified for the file (record type). Summary records will occur whenever the file sequencing information changes on any of the fields specified as sequencing fields for the summary record type. To obtain the desired summary records, the programmer must specify, for each summary record type, the sequence levels which, when their contents change, are to cause summaries to be taken. This may be specified either with DICT cards or with a parent code. To obtain a single summary for the entire file, the programmer need not specify any sequence fields other than record type.

The method of summary record placement described above may give rise to the insertion of some redundant summary records when a group or family relationship exists. These redundant summary records are placed between records which do not contain fields for all sequence levels and records which contain additional sequence fields. While these redundant summary records do not affect the accuracy of the file, they do slow the processing speed and they may be difficult to handle when using the file as input to a reporting operation. To eliminate these redundant records, it is suggested that at least the lowest level sequence field (excluding record type) be carried in the summary record (i.e., not obtained from a parent), and that a vertical deletion packet, with selection based on this field containing the lowest possible value, be used to delete the redundant records. (Vertical packets are discussed later in this manual.)

Examples of Dictionary Action Codes

The following examples will become clearer as the discussion of the File Processor progresses.

1. File establishment. A dictionary packet with I (Insert) record action and I (Insert) field action on a series of DICT and SUMM function cards is included for each type of record to be created in the output file. These packets must be placed immediately after the *END card and before the necessary update, horizontal, and vertical packets describing the input data.

   The first field of each record must contain the record type code. This must be specified in the dictionary packet as a two character (or twelve bit) unpacked BCD field with field number 0001 and must be given the lowest sequence level designation in the file.

2. Inserting a new record type in an existing file. This is handled exactly the same as file establishment.

3. Deleting an existing record type. Only the header card is used and the record action is listed as D. (If a parent record type is deleted, its offspring are automatically deleted.)

4. Removal of summary records. When all summary action related to a given summary record type has been stopped as a consequence of field mode changes, field deletions, record deletions, or the cessation of summarization, the summary record type must be removed from the file.

5. Modifying an existing record type. The modification of one record type will sometimes require modifications to other record type definitions in the dictionary and usually requires changes to related vertical, update, and horizontal packets.

   a. Modifications to fields. The modifications listed below may be used in any combination and are accomplished with record action M (Modify), field action M, and insertion of the new value in the appropriate place:

Parent code
Sequence level
Allowable range
Replace only code

b. To insert a field. Use record action M and field action I. If the inserted field is to be a sequencing field, it may be necessary to adjust the sequence level of fields in this and other records. During the execution of the dictionary change object program, a MTCH card referencing a blank change field value must be used to provide matching equality, since the system will automatically insert blanks as a part of the reading process. Also, a PUT macro-instruction (see Hand Calculations) must be used to move the actual value from the change record to the master file record. With subsequent routine processing, the PUT macro-instruction and the temporary MTCH card must be replaced by a valid MTCH card.

c. To delete a field. Use record action M with field action D (Delete). If the field is being summed, the summary is stopped, but it may be necessary to delete the field in the summary record type (perhaps the entire record type). If the field is a sequencing field, it may be necessary to adjust the sequence level of higher level sequence fields in this and other records.

d. To change the length of the field. Use record action M with field action R or L. The new length may be longer or shorter than it was originally.

Field Action Code	Mode	Change	Action
R	BCD	Shorten	Keep left-justified, truncate on right.
R	BCD	Lengthen	Keep left-justified, blanks inserted on right.
R	BIN	Shorten	Divide by a power of 10.
R	BIN	Lengthen	Multiply by a power of 10.
L	BCD	Shorten	Keep right-justified, truncate on left.
L	BCD	Lengthen	Keep right-justified, blanks inserted on left.
L	BIN	Shorten	No action.
L	BIN	Lengthen	No action.

e. To stop the summing of a field. Use M record action and D field action on a SUMM card. Note also, some change may be needed in the summary record type.

f. To start a binary field to be summed. Use M record action on header card, and I field action on a SUMM card. Note also, some change may be needed in the summary record type. This can be used with a newly inserted field or an existing field.

Elements of File Processor Change Packet Coding Forms

The following is a general description of the sections and fields of the various File Processor change packet coding forms which are used for update, vertical, and horizontal changes. These forms are described in terms of their three major subdivisions: header cards, selection cards, and processing cards.

Header Cards

FUNCTION

Each of the coding forms contains heading information which describes the function to be performed. These functions were discussed previously; the function is usually pre-printed on the coding form.

MASTER FILE RECORD TYPE

This is the numeric designation of the master file record type. In all instances, a packet pertains to only one record type; thus, a separate packet is required for each record type to be operated upon.

RECORD ACTION CODE

This code indicates the action to be taken with respect to the specified record type. The action may be one of the following:

Action	Explanation
D	Delete with summary records automatically adjusted.
E	Delete without changing summary records.
I	Insert.
M	Modify.
O	Insert or modify. This has a built-in test; it will automatically add the record if it is not present or modify the record if it is present.

In each case, the action specified occurs only if the selection criteria is satisfied. With D and E action codes, only deletion of the highest level parent record (lowest parent record type) causes deletion of offspring records; if it is desired to delete an intermediate parent and its offspring, this may be accomplished through a hand calculation (see "Examples of Hand Calculations" in Chapter 5).

FREQUENCY OF CHANGE REPORT

The Frequency of Change Report field only has an effect if the Change Report Field Request form is also used. When used, the frequency field is a four column field, as follows:

1. If the first column is a T, only the "after" change report is written.

2. If the first column is not T, then the change record may also be written before change, in accordance with the previous discussion of change reports.

3. The last three columns in this field indicate the frequency of writing the change records (either the "after" change or both the "before" and "after" change); a number, n, designates that the first and every nth change, thereafter, is to be written.

4. If the field is not coded, a change report will be written each time a change occurs.

SPECIAL ENTRIES FOR UPDATE PACKETS

The header card for update changes must specify the file number of the file which contains the change data to be used by the packet. If the change file does not have a dictionary, the logical record length must be stated (regardless of whether the records are blocked or unblocked). If the change file contains a dictionary, the dictionary file number must be stated (if the change file does not have a dictionary, this field must be blank to signal the File Processor that the change file does not contain a dictionary). For a change file with a dictionary, the record type of the change record to be used with this packet must be stated.

SPECIAL ENTRIES FOR HORIZONTAL PACKETS

The header card for horizontal changes must indicate the file number of the file which contains the processed horizontal data. The field action must be specified; if it is specified as P, this indicates that the field action is included in the horizontal data (triplets). If it is not P, the field action is consistent for all changes (pairs). The increment of the first pair or triplet within the change record must be stated; this, in effect, is the length of the preceding sequence, matching, etc., information.

Selection Cards

Selection is part of the process of determining whether the prerequisite conditions for making a change are met; it is augmented by matching, which is described below. Selection varies depending upon the type of change to be made.

In the case of vertical changes, selection is the determination of whether the proper master file record is present.

In the case of update or horizontal changes, selection is the determination of whether the proper change record is present.

Certain selection is automatically provided as a result of information supplied on the packet header card. In the case of vertical changes, it is the master file record type. In the case of update changes where the change file has a dictionary, it is the change file record type.

The SEL and NOT functions may be used for further selection of change records. If no SEL or NOT instructions are present, all change records on the indicated file will be accepted for that packet. If there is more than one packet using the same input file, only the last may be void of SEL and NOT functions. (This is sometimes useful to detect any change data not selected by preceding packets because of card punching or other errors.) Fields for SEL and NOT are specified in terms of record type and field number if the change file contains a dictionary, in bits or characters if the file does not contain a dictionary. Partial fields may be referenced when using an update change file with dictionary.

SEL

If the field specified by the SEL card matches the change selection value or range, the record is accepted.

NOT

If the field specified by the NOT card matches the change selection value or range, the record is rejected.

USING SEL AND NOT

SEL and NOT may be used separately or together in any combination. When SEL and NOT apply to the same field:

1. Two or more SELs are combined in an "or" fashion.

2. Two or more NOTs are combined in an "or" fashion.

3. SELs and NOTs are combined in an "and" fashion.

When SEL and NOT apply to different fields:

1. SELs and NOTs which apply to the same field must be grouped together.

2. Each group which applies to the same field is combined as above.

3. The groups are combined in an "and" fashion.

Example:

The following example applies to an update change without dictionary:

Function	Change Field Increment	Change Field Length	Selection Value or Range
SEL	000	001	A-E
NOT	000	001	C
SEL	000	001	R-T
SEL	010	003	008

This would cause the change record to be selected for operation by this packet if the first column contains A, B, D, E, R, S, or T and columns 11, 12, and 13 contain 008.

Processing Section

The third section on the change form is the processing section. In the case of a vertical change, the actual value to be inserted is specified by the VAL function; this value replaces the existing value in the selected fields. There are five other functions which are used for processing update and horizontal changes: MTCH, FLD, SEQn, SELF, and NOTF.

MTCH

MTCH compares the sequence fields of the change record with the sequence fields of the master record and takes action based on the relationship of the sequence fields, as follows:

1. If the change data is high, all vertical packets which apply to the master file record type are executed, the record is written, and a new master file record is read and compared against this change record.

2. If the change data equals the master file data, the master file is modified as indicated by the change file and a new change file record is read for another comparison with the master file record.

3. If the change data is lower than the master file data, the master file is backspaced one record, the change record is inserted, and the next change file record is read and compared against the current master file record (the one just inserted).

A MTCH card must be present for every sequence field contained in the record (or in any parent of the record) except the record type. The change record associated with the change packet must contain a field for matching each sequence field used to identify the master file record (except record type).

Example:

Suppose a personnel file consists of two types of records: record type 10, which is sequenced on department number, and record type 20, which is sequenced on department number and man number. If no parent-offspring relationship exists, the portion of the dictionary associated with the sequencing fields might be:

Function	Record Type	Field Number	Sequence Level	Description
DICT	10	0001	03	Record type
DICT	10	0005	01	Department number
DICT	20	0001	03	Record type
DICT	20	0004	01	Department number
DICT	20	0012	02	Man number

The MTCH cards needed for any packet intended to update a man record (record type 20) would be:

Function	Record Type	Field Number	
MTCH	20	0004	etc.
MTCH	20	0012	etc.

If, however, record type 20 had specified that record type 10 was its parent, the dictionary entries might be:

Function	Record Type	Field Number	Sequence Level	Parent Code	Description
DICT	10	0001	03		Record type
DICT	10	0005	01		Department number
DICT	20	0001	03	10	Record type
DICT	20	0012	02		Man number

The MTCH cards needed to update a man record would be:

Function	Record Type	Field Number	
MTCH	10	0005	etc.
MTCH	20	0012	etc.

MTCH cards may be used with non-sequencing fields for additional checking.

Example:

A file is maintained using man number as a sequencing field; a packet is to be written which will change pay rate. Before making the change, it is desired to check the name field, which is not a sequencing field, to be sure there was no error in card punching the man number.

In the above example, a MTCH card must be used on the sequencing field for man number and may be used on the non-sequencing field for name. Note that these non-sequencing MTCH cards are meaningful only when equality is required; if equality does not exist, an unmatched error record is written and the change action does not occur.

SEQn

All incoming change data is automatically sequence checked on the basic file sequence fields, as specified in the MTCH function lines. If it is desired to augment this automatic feature, that is, to sequence check the change data on additional fields, the SEQ function will be used, followed by a single digit, n, specifying the order in which additional testing is to be performed. The digits (n's) are assigned in consecutive order beginning with 1, and the cards punched for the SEQn function must appear adjacent to one another in this order in the priority packet. SEQn cards must immediately follow the MTCH cards.

FLD

This function specifies the field action to be taken after the record has been selected and matched; this may be one of the following:

1. Replace. The field in the master file is replaced by the field in the change file.

2. Add (or subtract). The change field is added to (or subtracted from) the binary master file field.

The FLD card in an update without dictionary packet that applies to a binary master file field may include an allowable range for changes; if the change field does not fall within this range, no change will be made, instead, an error report record will be written.

Selective Fielding

Provision is also made for examining the data and determining whether each FLD card is to be done. The operation codes used are SELF and NOTF. Although these cards are intermixed with FLD cards, they follow the same logic and format as SEL and NOT cards; SELF and NOTF cards must be grouped together preceding the single FLD card to which they apply.

SELF

If the selection value or range is met, the action specified by the FLD card is done.

NOTF

If the selection value or range is met, the action specified by the FLD card is not done.

Example:

Suppose that time cards for a company contain sequencing information in columns 1-9 and then are organized as follows:

Columns	Contents
10-12	Straight time hours
20-25	Expense dollars
40	A code as follows:

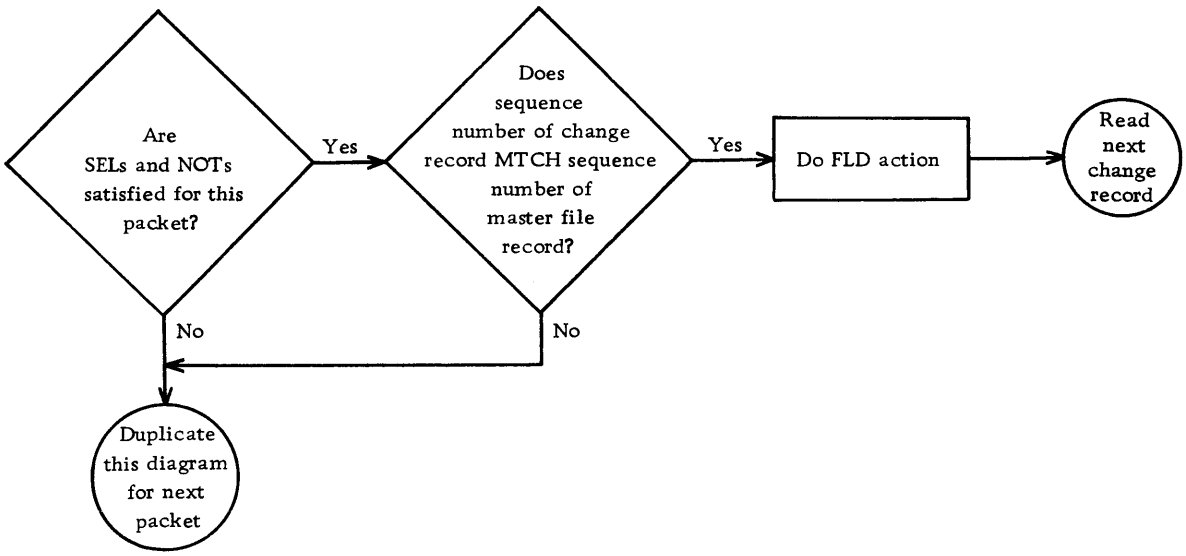
b	No action is to be taken based on information in columns 41-80
1	Columns 41-55 contain a new name
2	Columns 41-59 contain a new address

If card punching instructions on expense (columns 20-25) were such that zeros were not to be punched, the fielding section might be:

Function	Increment	Length	Action	Selection Value	Description
FLD	009	003	+		Add hours.
NOTF	019	006		bbbbbb	Skip next FLD card if field is all blank.
FLD	019	006	+		Add expense dollars.
SELF	039	001		1	Do next FLD if column 40 contains a 1.
FLD	040	015	R		Replace name.
SELF	039	001		2	Do next FLD if column 40 contains a 2.
FLD	040	019	R		Replace address.

The Sequence of Operations

The following flow chart briefly summarizes the process by which the change packet causes or does not cause action to be taken:



Auxiliary File Processor Forms

Two additional coding forms are available for use with the File Processor. These are the Change Report Field Request form and the standard SHARE 709 symbolic coding form; the latter is discussed in the chapter on hand calculations.

The various fields of the Change Report Field Request form are described in Chapter 4; cards prepared from this form must have the same major priority number as the packet in which they are included. These cards may appear anywhere in the packet following the header card but must be together and must not be between two cards of the same function (e.g., between two MTCH cards).

CHAPTER 4: FORMAT OF FILE PROCESSOR CODING FORMS

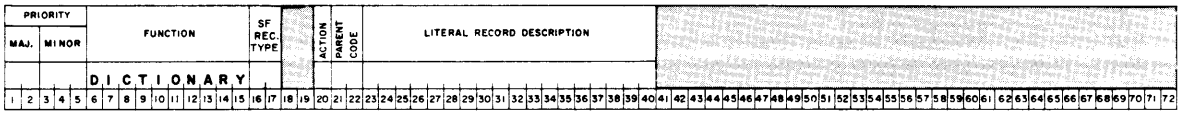
Following is a description of each of the fields of the various File Processor coding forms. Priority numbers are indicated on all File Processor forms except hand calculations, and they will not be further discussed in relation to the following coding information. For a detailed discussion of priority numbers see page 3.

When coding 9PAC forms, columns 73-80 are never used. Unless otherwise specified, all columns except comments columns must contain leading zeros.

Dictionary Definition Form

The Dictionary Definition form is shown in Figure 1.

Header Card



Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-15	Function	Must contain DICTIONARY.
16-17	Master File Record Type	This is the record type of the output record being defined by the packet.
18-19		Not Used.
20	Record Action	Must contain one of the following codes to describe the action to be taken: I Insert D Delete M Modify
21-22	Parent Code	Must be blank if no parent-offspring relationship is to exist; otherwise, must contain the record type of the parent record.
23-40	Literal Record Description	May contain any alphameric comments.
41-72		Not Used.

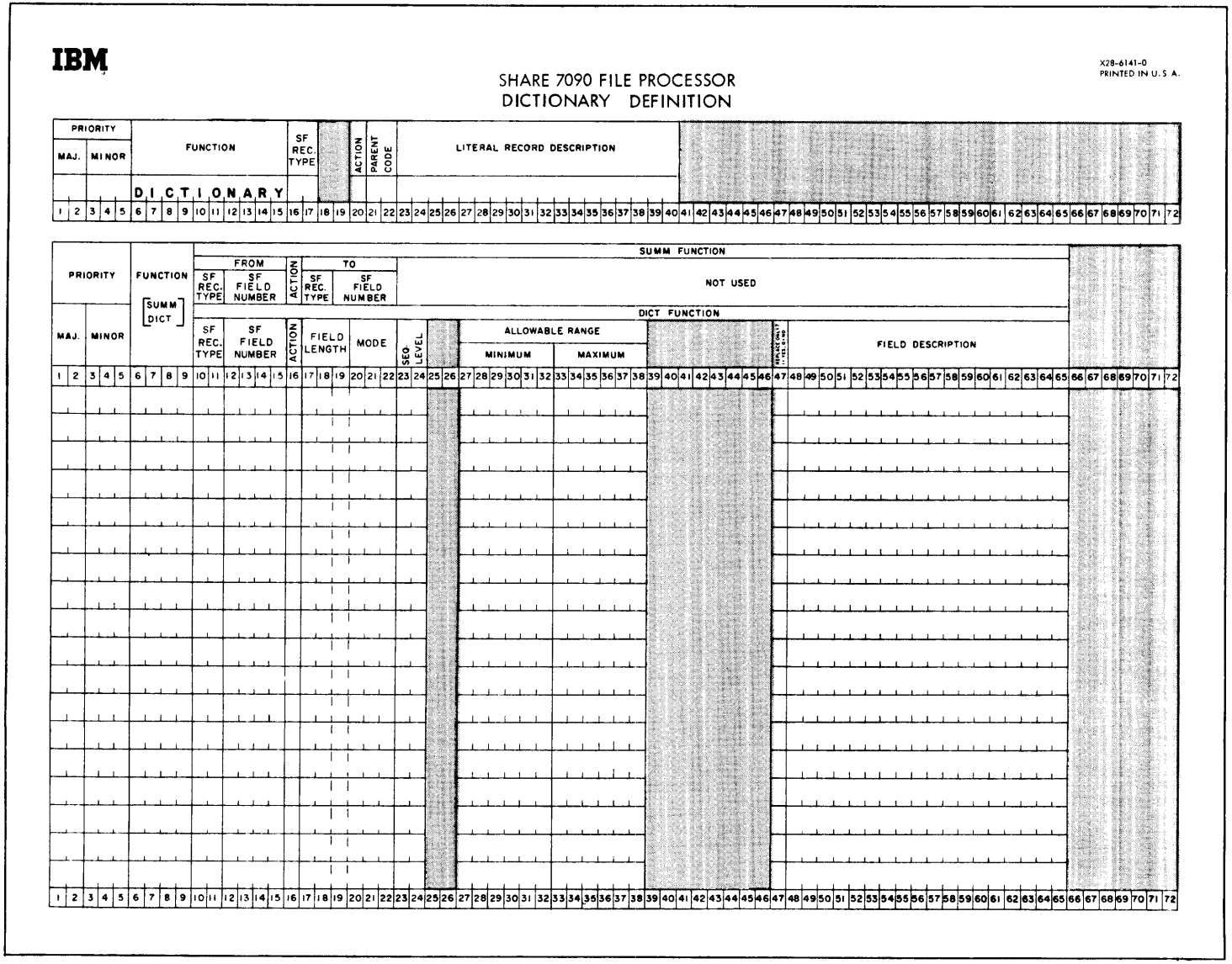
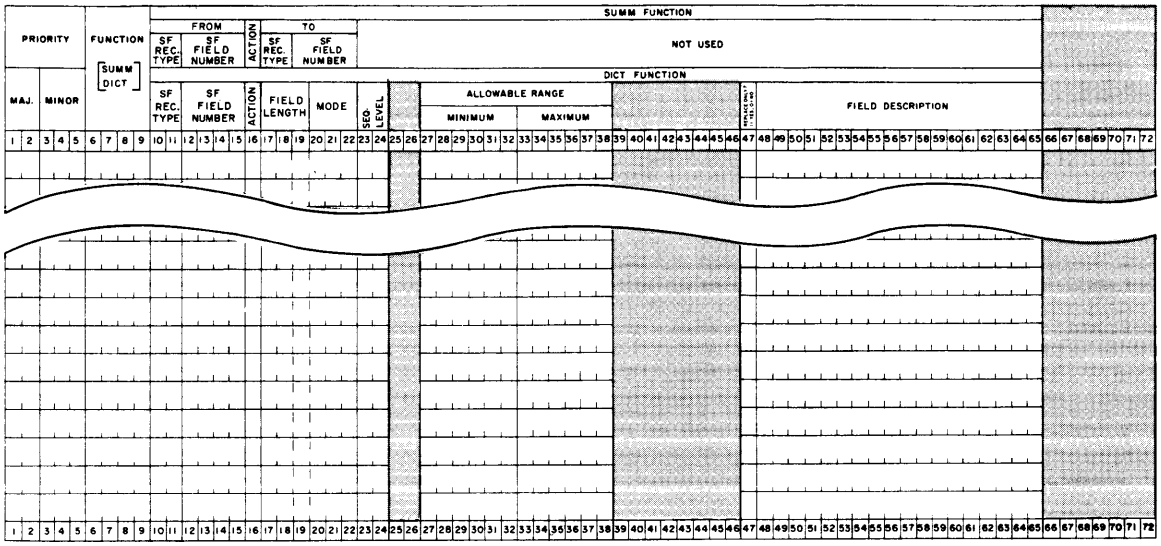


Figure 1. The File Processor Dictionary Definition form

Function Card

The center portion of the Dictionary Definition form may be used to prepare either dictionary function cards or summary function cards; the format of these cards will be discussed in turn.



DICTIONARY FUNCTION CARD

Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-9	Function	Must contain DICT.
10-11	Master File Record Type	This is the record type of the master file record which is: 1. Being created during file establishment. 2. Being changed during file maintenance.
12-15	Master File Field Number	This is the field number within the record type. These field numbers must be in increasing magnitude within a record type; it is suggested that gaps be left in the numbering sequence for later insertions.
16	Field Action	Must contain one of the following codes representing the action to be taken relative to the field designated by columns 10-15: I Insert D Delete M Modify R Change length of field on right L Change length of field on left
17-19	Master File Field Length	These columns specify the size of the output file field. If the method of field specification (column 20 of the *JOB card) is B, this entry is in bits; otherwise, this entry is in characters.

Columns/Contents		Description

Table of Allowable Entries			
BCD Fields		Binary Fields	
Characters	Bits	Characters	Bits
1	6	1	5
2	12	2	8
3	18	3	11
.	.	4	15
.	.	5	18
.	.	6	21
n	6n	7	25
.	.	8	28
.	.	9	31
.	.	10	35
166	996	-	-

20-22 Mode

Must contain one of the following codes to describe the data field being defined:

100 Unpacked BCD
110 Packed BCD
001 Binary

If the field is to have any arithmetic operations performed on it, it must be binary. A non-arithmetic field that will be referenced frequently should be unpacked BCD (in which case it will be right-justified) for maximum computer speed. Large amounts of seldomly referenced data should be packed BCD to save tape space.

23-24 Sequence Level

If the field being defined is not a sequence field, these columns must be blank. If the field is a sequence field, these columns must contain a sequence level number. The primary sequence field of a record type must be designated as 01, the secondary, 02, etc. If the record type is an offspring record, the sequence level starts after the last level to be obtained from the parent record. Note: A binary field may not be a sequence field.

25-26

Not Used.

27-38 Allowable Range

This is the designation of the minimum and maximum values which a numeric (binary) field may assume. These columns may all be blank, or either minimum or maximum may be coded separately. Entries must be right-justified with leading zeros and may have a minus over-punch in the units position; entries must be numeric. If the field size is greater than six characters (21 bits), minimum-maximum values correspond to the high order 6 characters. If a change is attempted which would result in exceeding the stated allowable range, the change will not be made and an error report will be written.

39-46

Not Used.

Columns/Contents		Description
47	Replace Only Code	A 1 in this column will permit the field to be changed by replacement only; this must be used for BCD fields. A 0 allows the field to be changed by addition, subtraction, or replacement. This field must not be blank.
48-65	Field Description	May contain any alphameric comments.
66-72		Not Used.

SUMMARY FUNCTION CARD

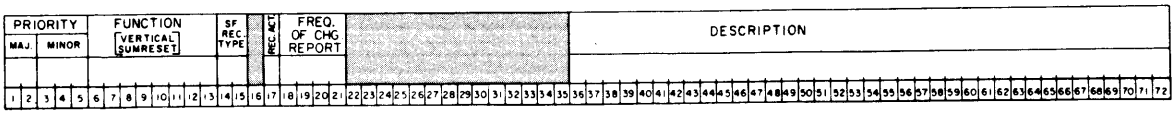
This card is used to specify a field which is to be summed and the field into which it is to be summed. The SUMM cards for a field must immediately follow the DICT card for the field being summed.

Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-9	Function	Must contain SUMM.
10-15	Master File Record Type and Field Number	Must contain the output master file record type and field number of the field to be summed.
16	Action Code	Must contain one of the following codes to indicate action to be taken: I Insert D Delete
17-22	Master File Record Type and Field Number	Must contain the summary record type and field number into which the sum will be placed. This record type must follow (and thus have a higher number than) the record type being summed.
23-72		Not Used.

Vertical Change Definition Form

This form is used either to make a consistent change to all records of a given record type which meet the selection criteria, or for the SUMRESET function. The Vertical Change Definition form is shown in Figure 2.

Header Card



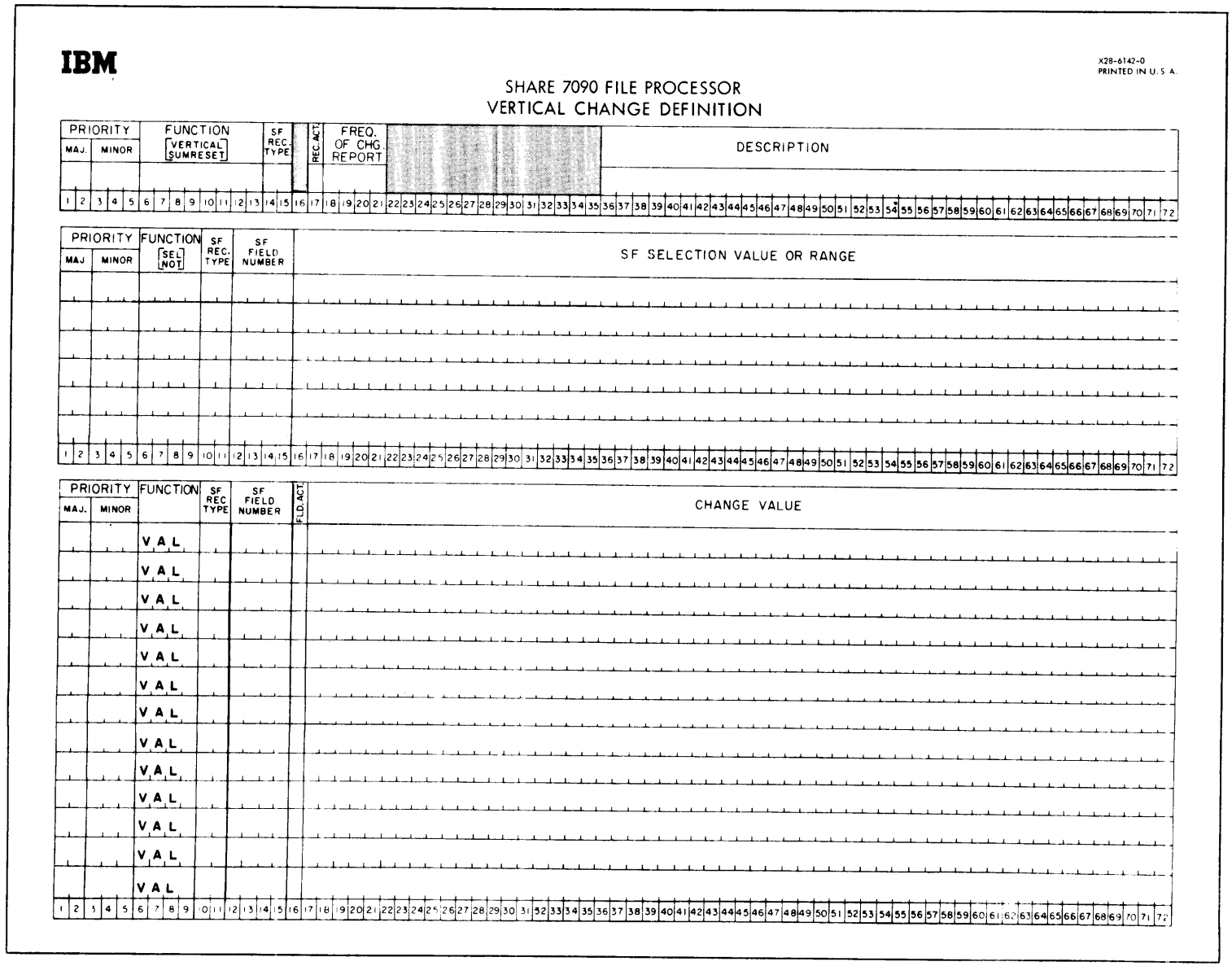
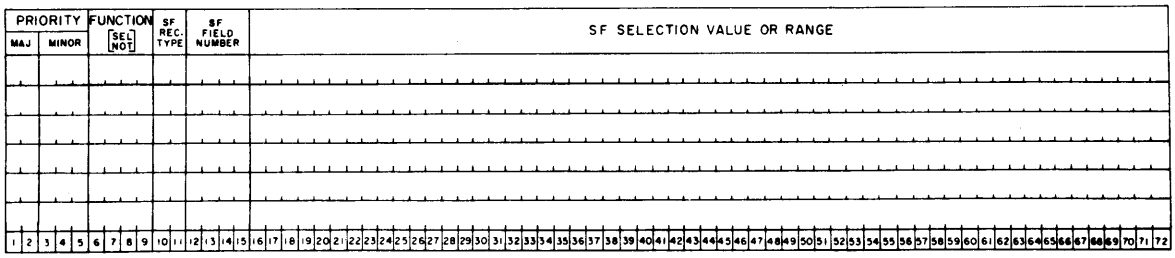


Figure 2. The Vertical Change Definition form

Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-13	Function	Must contain VERTICAL or SUMRESET.
14-15	Master File Record Type	Must contain the input master file record type to be affected by this change.
16		Not Used.
17	Record Action	Must contain one of the following codes if the record is to be deleted: D Delete with summaries adjusted. E Delete with summaries not adjusted. For all other vertical changes, this column must be blank.
18-21	Frequency of Change Report	Applicable only if there are Change Report Field Request cards in this vertical packet. A T in column 18 indicates only "after" change records are to be written; columns 19-21 may contain a reporting frequency, n.
22-35		Not Used.
36-72	Description	May contain any alphameric comments.

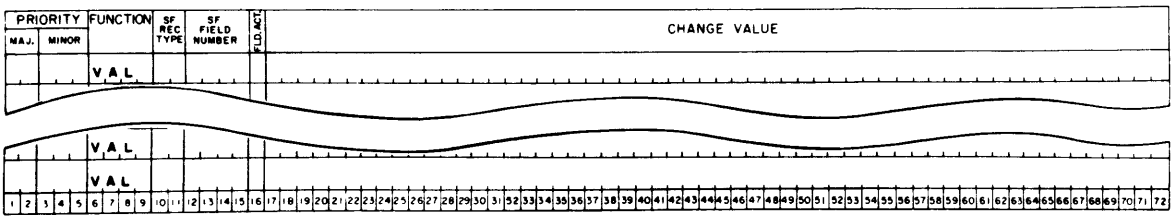
Selection



Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-9	Function	May contain one of the following selection instructions, written left-justified: SEL Select NOT Not Select
10-15	Master File Record Type and Field Number	This is the record type and field number used to determine which records in the master file are to be selected. The entry in columns 10-11 may be different than the entry in columns 14-15 of the packet header if the selection is made based upon data in a lower numbered record type.

Columns/Contents		Description
16-72	Master File Selection Value or Range	The specific value or range of values to be used for selection are entered in these columns. Each entry must be equal in length to the length of the field defined in the dictionary. The minimum value must precede the maximum value and be separated from it by a minus sign (11-punch).

Processing Cards



Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-9	Function	Must contain VAL, left-justified in the field.
10-15	Master File Record Type and Field Number	This is the record type and field number of the field to be changed.
16	Field Action	Must contain one of the following codes to indicate the change activity to take place: R — Replace the current entry in the field by the value specified in the change field. + — Add the value specified in the change field to the current entry in the master file field being changed. Valid for binary fields only. - — Subtract the values specified in the change field from the current entry in the master file field. Valid for binary fields only. S — Change the entry in the master file field to blanks if the field is BCD or to zeros if the field is binary and adjust the summary records, if applicable. T — Same as S, except that summaries are not affected.
17-72	Change Value	For R, +, and - field action, this is the actual value to be processed into the master file. The entry may be signed or unsigned, must be left-justified, and must be the same length as the master file field defined in the dictionary. This field must be blank for S and T field action.

Update Change Definition

The Update Change Definition form is shown in Figure 3.

IBM

X28-6143-0
PRINTED IN U.S.A.

SHARE 7090 FILE PROCESSOR
UPDATE CHANGE DEFINITION

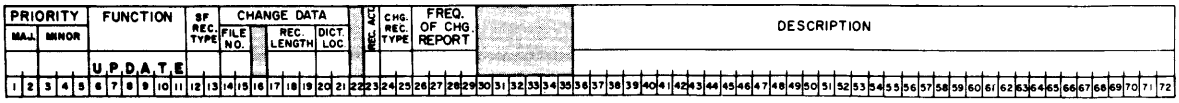
PRIORITY MAJ.	PRIORITY MINOR	FUNCTION	SF REC. TYPE	CHANGE DATA: FILE NO.	CHANGE DATA: REC. LENGTH	CHANGE DATA: DICT. LOC.	REC. ACT.	CHG. REC. TYPE	FREQ. OF CHG. REPORT		DESCRIPTION
		U,P,D,A,T,E									
1 2 3 4 5		6 7 8 9	10 11	12 13	14 15 16	17 18 19	20 21	22	23 24 25	26 27 28 29	30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

PRIORITY MAJ.	PRIORITY MINOR	FUNCTION [SEL NOT]	CHG. LOCATION	PARTIAL FIELD	CHANGE SELECTION VALUE OR RANGE
1 2 3 4 5		6 7 8 9	10 11 12 13 14 15	16 17 18 19 20 21	22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

PRIORITY MAJ.	PRIORITY MINOR	FUNCTION [MTCH, FLD SEQn]	SF REC. TYPE	SF FIELD NUMBER	CHG. FIELD INCR	CHG. FIELD LENGTH	FLD ACT	ALLOWABLE RANGE: PARTIAL FIELD		COMMENTS
1 2 3 4 5		6 7 8 9	10 11	12 13 14 15	16 17 18 19	20 21	22	23 24 25 26 27 28	29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	

Figure 3. The Update Change Definition form

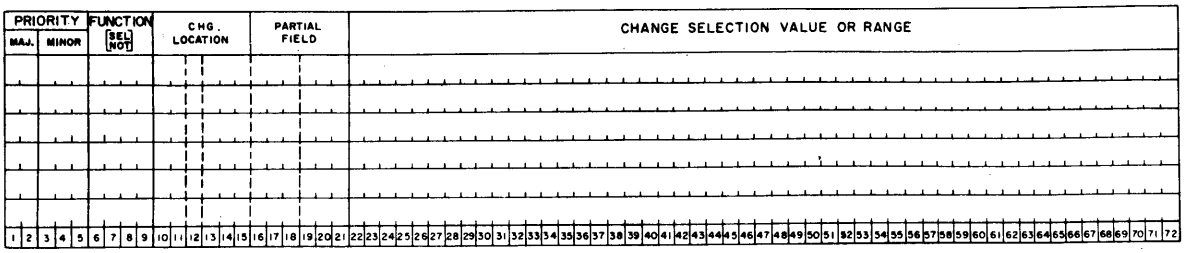
Header Card



Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-11	Function	Must contain UPDATE.
12-13	Master File Record Type	Must contain the master file record type to be affected by this change.
14-15	Change File Number	Must contain the file number of the change file to be used by this packet; this is the association of the packet with a file described on an *FILE card.
16		Not Used.
17-19	Change Data Record Length	Must contain the logical record length, in 7090 machine words, for an update change without dictionary.
20-21	Dictionary File Number	This is the number of the file which contains the dictionary; it must be the same as columns 14-15 for an update change with dictionary. It must be blank for an update change without dictionary.
22		Not Used.
23	Record Action	This is a code which specifies the manner in which the master file records are to be affected by this change, and must be one of the following: I Insert change record into master file. D Delete master file record corresponding to change record and adjust summaries accordingly. E Same as D, but do not adjust summaries. M Modify master file records according to subsequent update packet instructions. O Action is the same as M if change record matches source record; otherwise, action is same as I.
24-25	Change Record Type	For an update change with dictionary, this field must contain the record type of the change record to be used by this packet; otherwise, these columns must be blank.

Columns/Contents		Description
26-29	Frequency of Change Report	Applicable only if there are Change Report Field Request cards in the packet. A T in column 26 indicates only "after" change records are to be written; columns 27-29 may contain a reporting frequency, n.
30-35		Not Used.
36-72	Description	May contain any alphameric comments.

Selection Criteria



Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-9	Function	Must contain one of the following codes to indicate selection: SEL Select NOT Not Select
10-15	Change Field Location	Must contain the record type (columns 10-11) and field number (columns 12-15) of the change field if the change file contains a dictionary; must contain the increment (columns 10-12) and length (columns 13-15) of the change field if the change file does not contain a dictionary.
16-21	Partial Field	For an update change with dictionary, may contain the increment and length within the field of a partial field which is to be used for selection.
22-72	Change Selection Value or Range	May contain the specific value or range of values to be used for selection. Entries must be the same length as the change field. Minimum values must precede maximum values and be separated from maximum values by a minus sign (11-punch).

Processing Cards

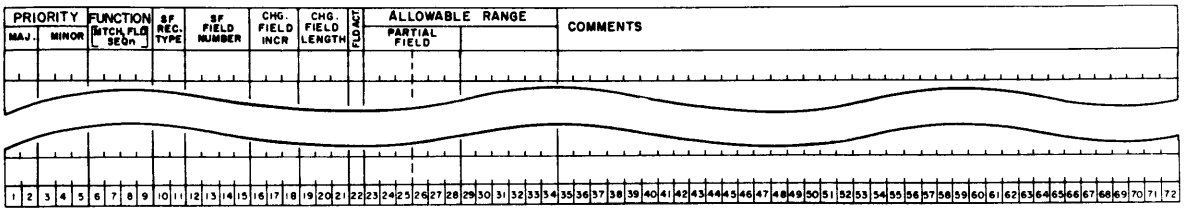
The final portion of the form is used for:

1. Selective fielding (SELF and NOTF functions). These cards are prepared in the same manner and format as SEL and NOT cards except that the terminating F is added and the cards are located in the processing section preceding the FLD card to which they apply.

2. Matching (MTCH function). The sequencing fields of the change file record are matched against the sequencing fields of the master file record.

3. Processing (FLD function). These cards specify how the change record is to affect the master file record.

These cards will each be dealt with separately.



MTCH AND SEQn FUNCTION

Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-9	Function	Must contain MTCH or SEQn. A MTCH card must be present for each sequence field of the master file (even though, if the record is an offspring record, the sequencing fields may not all be contained in the master file record). MTCH cards may also be used for non-sequence fields.
10-15	Master File Record Type and Field Number	Must contain the master file record type (columns 10-11) and field number (columns 12-15) of the master file field to be matched (MTCH) or checked for sequencing (SEQn).
16-21	Change Field	This field is used to specify the change field which is to be matched or checked for sequencing. For an update change without dictionary, it is stated in terms of increment (columns 16-18) and length (columns 19-21); for an update change with dictionary, it is stated in terms of record type (columns 16-17) and field number (columns 18-21).
22		Not Used.
23-28	Partial Field	For an update change with dictionary, this field may be used to specify a partial field (within the record type and field number) which is to be used for matching or sequencing; it is stated in terms of increment (columns 23-25) and length (columns 26-28).
29-34		Not Used.
35-72	Comments	May contain any alphameric comments.

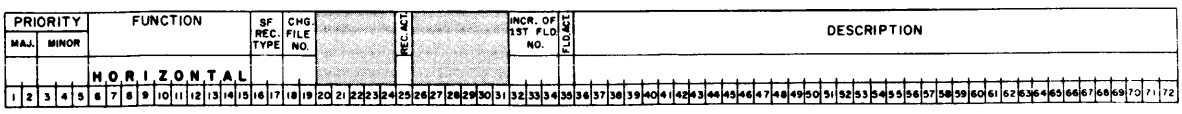
FLD FUNCTION

Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-9	Function	Must contain FLD.
10-15	Master File Record Type and Field Number	Must contain the master file record type (columns 10-11) and field number (columns 12-15) of the master file field to be changed.
16-21	Change Field	This field is used to specify the change record field to be used in making the change. For an update change without dictionary, it is specified in terms of increment (columns 16-18) and length (columns 19-21); for an update change with dictionary, it is specified in terms of change file record type (columns 16-17) and field number (columns 18-21).
22	Field Action	Must contain one of the following codes to indicate how the change field is to be incorporated into the master file field: R Replace master file field with change field. + Add change field to master field, which must be binary. - Subtract change field from master field, which must be binary.
23-28	For Update Change With Dictionary	May contain the increment (columns 23-25) and length (columns 26-28) within the change file record type and field number of a partial field which is to be used in making the change.
29-34	For Update Change With Dictionary	Not Used.
23-34	For Update Change Without Dictionary	May contain the minimum-maximum test range on the change value itself. The minimum value must precede the maximum value. Note that this testing may be used to supplement the allowable-range testing on the master file which is specified in the master file dictionary.
35-72	Comments	May contain any alphameric comments.

Horizontal Change Definition Form

The Horizontal Change Definition form is shown in Figure 4.

Header Card



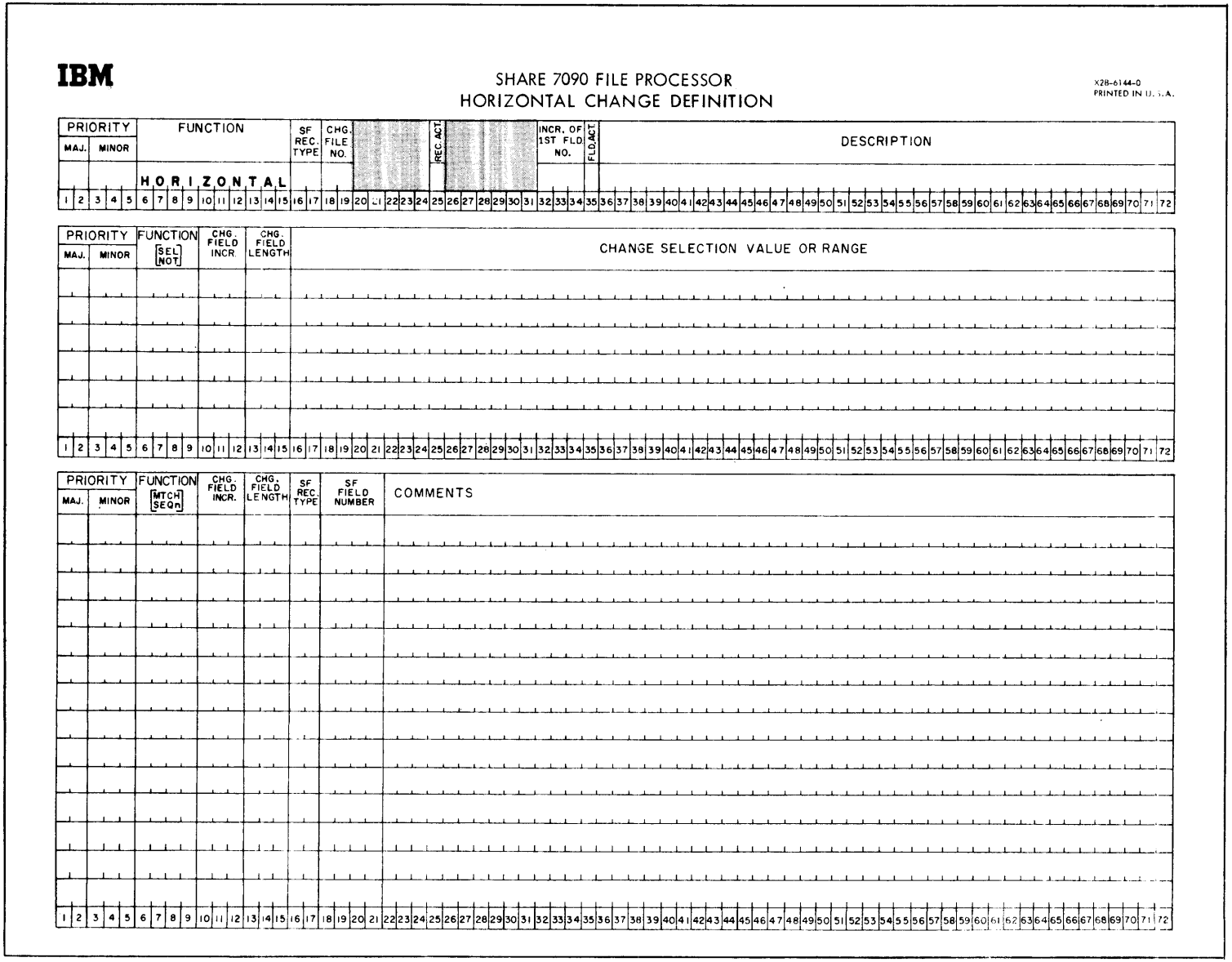
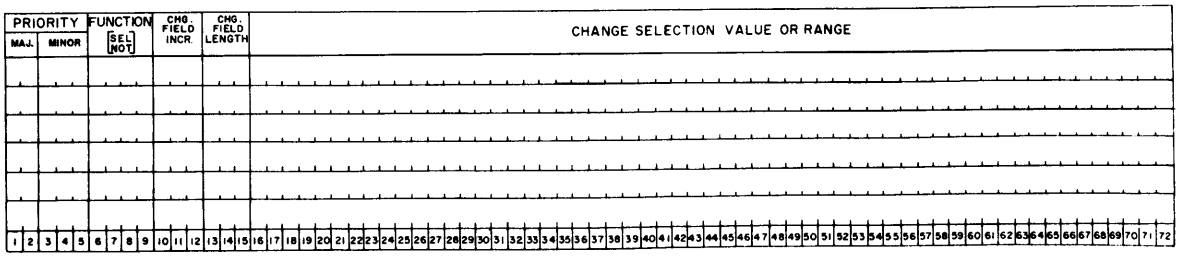


Figure 4. The Horizontal Change Definition form

Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-15	Function	Must contain HORIZONTAL.
16-17	Master File Record Type	Must contain the record type of the master file record to be affected by this change.
18-19	Change File Number	Must contain the number of the change file which contains the processed horizontal change data, as described by an *FILE card.
20-24		Not Used.
25	Record Action	Must be one of the following codes designating the record action to be taken: I Insert change record. D Delete master record with summaries adjusted. E Delete master record with summaries not adjusted. M Modify master records according to field action code.
26-31		Not Used.
32-34	Increment of First Field Number	This is the increment of the first pair (field-value) or triplet (field-action-value) field contained on the change record.
35	Field Action	Must contain a code indicating the field action to be taken: R Replace master file field with change data value. + Add change value to master file field. - Subtract change value from master file field. P As specified with change data; this code must be used with, and only with, triplets.
36-72	Description	May contain any alphameric comments.

Selection Criteria

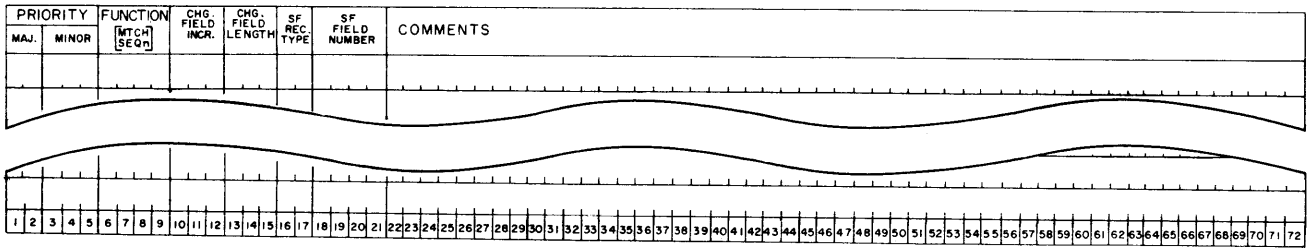


Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.

Columns/Contents		Description
6-9	Function	Must contain one of the following codes to indicate selection: SEL Select NOT Not Select
10-15	Change Field Location	Must contain the increment (columns 10-12) and length (columns 13-15) of the change field.
16-72	Change Selection Value or Range	Must contain the specific value or range of values to be used for selection. Entries must be the same length as the change file field being tested. Minimum values must precede maximum values and be separated from them by a minus sign (11-punch).

Processing Specifications

The final portion of this form is used for matching (MTCH). The sequencing fields of the change file are matched against the sequencing fields of the master file.



Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-9	Function	Must contain MTCH or SEQn. A MTCH card must be present for each sequence field of the master file record (even though, if the record is an offspring record, the sequence fields may not all be contained in the master file record). MTCH cards may also be used for non-sequence fields.
10-15	Change Field	Must contain the change field to be matched (MTCH) or checked for sequencing (SEQn). This is stated in terms of increment (columns 10-12) and length (columns 13-15).
16-21	Master File Record Type and Field Number	Must contain the master file record type (columns 16-17) and field number (columns 18-21) of the master file field to be matched or checked for sequencing.
22-72	Comments	May contain any alphameric comments.

Change Report Field Request Form

The Change Report Field Request form is shown in Figure 5.

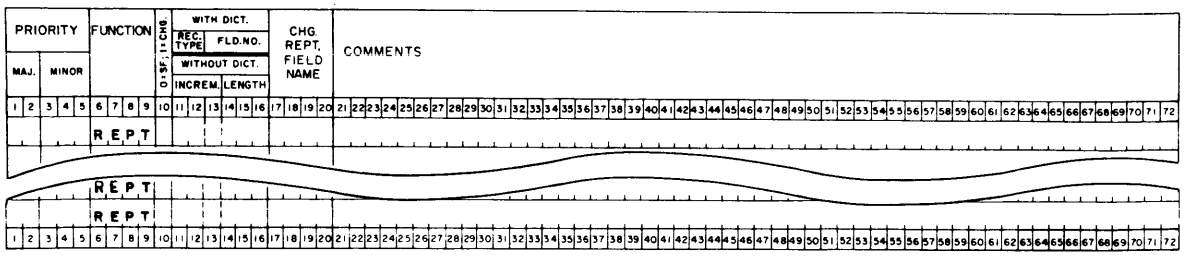
IBM

SHARE 7090 FILE PROCESSOR
CHANGE REPORT FIELD REQUEST

X28-6145-0
PRINTED IN U.S.A.

PRIORITY		FUNCTION	0=SF, 1=CHG	WITH DICT.		CHG. REPT. FIELD NAME	COMMENTS
MAJ.	MINOR			REC. TYPE	FLD.NO.		
				WITHOUT DICT.			
				INCREM.	LENGTH		
1 2 3 4 5		6 7 8 9	10	11 12 13	14 15 16	17 18 19 20	21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
		REPT					
1 2 3 4 5		6 7 8 9	10	11 12 13	14 15 16	17 18 19 20	21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

Figure 5. The Change Report Field Request form



Columns/Contents		Description
1-5	Priority Number	Must contain the priority number of the card.
6-9	Function	Must contain REPT.
10	Origin of Field Being Requested	Must contain one of the following codes to designate where the field being requested is to come from: 0 Master file 1 Change file
11-16	Location of Field Being Reported	If the field being reported is in a file which contains a dictionary, this is specified in terms of record type (columns 11-12) and field number (columns 13-16). If the field being reported is in a file which does not contain a dictionary, this field is stated in terms of increment (columns 11-13) and length (columns 14-16).
17-20	Change Report Field Number	Must contain a literal field designation by which the field being reported may later be referenced. Certain literals have been reserved for use in horizontal packets only, as follows:

Literal Field Number	Included in Change Report
FROM	Master field before change
TO	Master field after change
FLD#	Number of field changed
DSC	Description of field changed
ACTR	Record action specified
ACTF	Field action specified
TYPE	Record type of field changed

21-72	Comments	May contain any alphameric comments.

CHAPTER 5: HAND CALCULATIONS

The basic automatic properties of the File Processor can be augmented by the insertion of handwritten programs called hand calculations.

Hand calculations are coded in the SCAT language (see the SOS Reference Manual), in accordance with the rules, and using the additional language facilities, given in this chapter. The instructions themselves are written on the standard SHARE 709 symbolic coding form; each line must have a C in column 1.

Location of Hand Calculations

Hand calculations may be located in three positions of the generated object program; the actual location is dependent upon the location of the hand calculation statements in the source program priority packets:

1. Pre-calculations. If the cards are placed immediately following the packet header, the calculations will be positioned in the object program after any selection, matching, or pre-reporting and before any FLD or VAL action of the packet.

2. Post-calculations. If the cards do not immediately follow the packet header and do not satisfy the conditions in item 3 below, the calculations will be positioned in the object program after the FLD or VAL action and before any post-reporting of the packet.

3. Edit calculations (applicable to horizontal or update changes only). If the cards do not immediately follow the packet header and the first card has EDIT in columns 2-5, the calculation will be an edit calculation; it will be positioned immediately after the reading of any record from the file specified in the packet header.

Note that while pre-calculations and post-calculations are associated with a particular priority packet, edit calculations are associated with a particular input file; therefore, only one edit calculation may appear in all the packets having the same input file, and it is in no way affected by the choice of which of those packets it is included in.

Calculations are always entered at the first instruction following the associated storage locations (see below) and exit is always after the last physical instruction in the calculation is executed. The last instruction must be in the path of flow and must not be a transfer or test-type instruction.

Location Symbols

With the exception of location symbols for associated storage, described below, location symbols for File Processor calculations are formed in accordance with the normal SCAT rules, modified as follows:

1. They may not be more than five characters in length.

2. They must be written left-justified, beginning in column 2.

3. They must not begin with a C. (For exceptions to this rule, see "Associated Storage Symbols," below.)

4. They must not be one of the special File Processor hand calculation symbols (see "File Processor Symbols," below).

5. The C in column 1 is not considered to be part of a symbol; it is a flag only.

Associated Storage Symbols

Associated storage is a block which contains constants, temporary storage locations, masks, and other related out-of-line references. Associated storage locations must be created in a block at the beginning of the calculations and must have symbols formed in accordance with the following:

1. The associated storage definitions must all precede any other type of instruction included in the calculations.

2. Each instruction defining an associated storage location must have a unique location symbol which must be of the form:

   Cx

   where x is a four digit number.

3. The address of each instruction defining an associated storage location must be (1) numeric, (2) a system symbol (see below), or (3) a literal (for BCI only).

Examples:

```
C0001     PZE     11
C0099     BCI     2,SAMPLEbENTRY
C0002     DEC     1
```

In addition, the following rules apply:

1. Each hand calculation is limited to 800 words of "C" constants.

   Expressions of the nature

```
C0005     BSS     100
C0200     BCI     8,yy...yy
```

   count against the allowable "C" constants (in these examples, 100 and 8 words respectively).

2. Each hand calculation is limited to 400 symbols including associated storage symbols.

Addresses

Symbolic addresses may be either a system symbol (see below), a 1-5 character location symbol, or an associated storage symbol.

Relative addressing is permitted by the File Processor, but it should not be used around macro-instructions, since these expand into several actual instructions in the object program.

A symbolic address in one calculation will refer to an instruction or data word in another calculation if it is formed in accordance with the following:

wxy + k

where w indicates that the reference is to an instruction or data word as follows:

$\bar{0}$ Reference is to a location following the C-symbol block.
$-\bar{0}$ Reference is to a location within the C-symbol block.

Note: $\bar{0}$ represents a 0-punch and an 11-punch.

x is the two digit major priority code of the packet which contains the calculation being referenced.

y is one of the following codes to indicate the type of calculation being referenced (edit calculations may not be referenced):

J Pre-calculation
P Post-calculation

k is the increment of the word being referenced within the associated storage block if w is $-\bar{0}$ or within the instruction block following the associated storage block if w is $\bar{0}$.

Example:

Assume that it is desired to use the value defined in packet 05, which is the 25th word in the pre-calculation for that packet and which was defined as

```
C0053   OCT   777
```

The required instruction would be

```
ANA     -0̄05J+24
```

Tag and Decrement

Tags and decrements are formed in accordance with SCAT rules except that (1) no symbols may appear in either the tag or decrement; (2) if a tag is present, there must be an address; and (3) if a decrement is present, there must be both an address and a tag.

Operations

File Processor calculations may include any of the operation codes given in Appendix D, with the following restrictions:

1. OCT is limited to 1 word.

2. DEC is limited to 1 word.

3. BCI is limited to 8 words.

In addition, the File Processor provides the following macro-instructions: GET, PUT, SUM, REJECT, IGNORE, ERRFLD, BKSP. These are discussed below.

The BCD and Binary Buffers

The GET, PUT, and SUM macro-instructions interpret the mode of the master file field and operate out of a buffer of that mode. The binary buffer, BINBF, is one word. The BCD buffer, BCDBF, is 14 words and is stored backwards in core storage. The first word of the BCD buffer is BCDBF; the second word is BCDBF-1; the 14th word is BCDBF-13. However, the last word of the field is placed in word 1 (BCDBF); the next to last word of the field is placed in word 2 (BCDBF-1); etc.

The contents of the binary and BCD buffers are not saved outside of the hand calculation in which they are used.

Example:

Symbolic Location	BCDBF-3	BCDBF-2	BCDBF-1	BCDBF
Sample Contents	bbJOHN	bPAULb	JONESb	&bCObb

The File Processor Macro-Instructions

GET

General Form	Examples
"GET xy" where x is a two-digit record type and y is a four-digit field number.	GET 100011 GET 300012 GET 330018

The GET instruction obtains the field referred to in the address field and places it in the BCD or binary buffer, according as the field is BCD or binary, respectively. GET may not be used in pre-calculations with I action.

PUT

General Form	Examples
"PUT xy" where x is a two-digit record type, and y is a four-digit field number.	PUT 100011 PUT 300012 PUT 330018

The PUT macro-instruction may be used only with post-calculations; its effect is to obtain the field referenced by its address portion from the buffer and place it into the master file. The PUT macro-instruction will also check the dictionary for the field and, if the field is indicated to be summed, it will automatically be added to the appropriate summary fields.

PUT may refer only to the current master file record type. PUT may not be used in pre-calculations with I action.

SUM

General Form	Examples
"SUM xy" where x is a two-digit record type and y is a four-digit field number.	SUM 100011 SUM 300012 SUM 330018

The SUM macro-instruction adds the contents of the binary buffer to all summary fields defined (by the dictionary) for the field specified by the address portion of the SUM macro-instruction. SUM may refer only to the current master file record type.

SUM may not be used in a pre-calculation with I action.

REJECT

General Form	Examples
"REJECT x" where x is 1-12 alphameric characters describing the reason for record rejection.	REJECT BAD-REC T REJECT BAD TRAN

The REJECT macro-instruction is used to create an error record when an input change is in error, as determined by a hand calculation packet. When this instruction is used in a pre-calculation, the change will not take place; when used in a post-calculation, the change will take place; in both cases, the error report will be written on tape and the next record will be read and examined.

The REJECT macro-instruction may not be used within a vertical change packet.

IGNORE

General Form	Example
"IGNORE"	IGNORE

The IGNORE macro-instruction is used when certain change records are to be by-passed but are not to appear in error records on the error tape.

When used in a pre-calculation, the IGNORE macro-instruction will cause the change record not to be operated upon at all; when used in a post-calculation, further instructions are not performed.

ERRFLD

General Form	Example
"ERRFLD x" where x is a 0-12 alphameric description of the reason for the error.	ERRFLD BAD-FLD-NO

The ERRFLD macro-instruction is used when processing horizontal changes; it rejects the field currently being processed.

BKSP

General Form	Example
"BKSP"	BKSP

The BKSP macro-instruction, which may only appear in a horizontal or update packet, inhibits reading the next change record. Thus, when exit is made from the priority packet, the same change record is available for use by the next change packet. The selection fields must be changed to prevent re-entry to the same priority packet and, thus, looping.

File Processor Subroutines

The following subroutines are available for use in File Processor hand calculations.

TOBIN (BCD TO BIN)

1. Store the leftmost BCD word, if more than one word, in DBBCL.

2. Store the rightmost BCD word in DBBCR.

3. Execute the following calling sequence:

```
TSX     TOBIN,4
                Error return
                Normal return
```

The error return will be made if:

a. The field contains all blanks.
b. The field contains a blank between two numeric BCD characters.
c. The field contains any BCD alphabetic characters (except the final character, which may be a numeric character with a minus overpunch).

4. The binary word will be located at DBBIN.

CVBIN (BCD TO BIN)

1. Place the BCD words in the two rightmost words of the BCD buffer (BCDBF and BCDBF-1).

2. Execute the following calling sequence:

```
TSX     CVBIN,4
                Error return
                Normal return
```

The error return will be made for the same reasons listed for the TOBIN subroutine.

3. The converted binary word will be located in BINBF.

TOBCD (BIN TO BCD)

1. Store the binary word in BDBIN.

2. Execute the following calling sequence:

```
TSX     TOBCD,4
```

3. The BCD words will be right-justified in BDBCL and BDBCR.

CVBCD (BIN TO BCD)

1. Place the binary word in BINBF.

2. Execute the following calling sequence:

```
TSX     CVBCD,4
```

3. The BCD words will be right-justified in BCDBF and BCDBF-1.

709 TO COMMERCIAL COLLATING SEQUENCE CONVERSION

1. Place BCD characters in AC, right-justified.

2. Execute the following calling sequence:

```
TSX     TO702,4
PZE     x       where x is the number of decimal digits, 1-6, placed
                in the AC.
```

3. The converted word will be located in the AC.

COMMERCIAL TO 709 COLLATING SEQUENCE CONVERSION

1. Place the BCD word to be converted in the AC, right-justified.

2. Execute the following calling sequence:

```
TSX     TO709,4
PZE     x       where x is the number of decimal digits, 1-6, placed
                in the AC.
```

3. The converted word will be located in the AC.

TOPRT (PRINT A MESSAGE ON-LINE)

1. Execute the following calling sequence:

```
TSX     TOPRT,4
op      x,0,y   where x is the location of the first word to be printed;
                y is the number of words to be printed; and op is
                PZE to print the line, PON to double-space and print
                the line, PTW to print the line as a heading line on
                the current page and on each succeeding page, or
                PTH to eject the current page (the address must be
                blank). If the prefix is minus, e.g., MZE, the
                next instruction will be interpreted as an extended
                calling sequence. The last word in the calling
                sequence must have a positive prefix.
```

FLIST (PRINT A MESSAGE ON SYSTEM LISTING TAPE)

1. Execute the following calling sequence:

```
TSX     FLIST,4
op      x,0,y   where op, x, and y are the same as for TOPRT.
```

File Processor Symbols

The following location symbols are available for use in hand calculations:

INLOC	Contains the address of the first word of the current master file input record.
SFLOC	Contains the address of the first word of the current master file record in the output buffer.
MRGLO	Contains the address of the first word of the current change record in the input buffer. This is the record last chosen from all change input units, by virtue of lowest sequence, for current processing. (In an edit calculation, this will not be the record just read on that particular input unit. The complement address, automatically set in Index Register 1, should be used instead.)
DICLO	Contains the address of the first word of the field dictionary involved with the current horizontal field change.
VALLO	Contains the address of the leftmost word of value associated with the current horizontal field change. If the master file field to be affected is binary, the change value will be binary. If the master file field is BCD, the change value will be unpacked BCD.
CHLOC	Contains the address of the first word of the current change (or error) report in the change file output buffer.
DSWCH	Delete switch. STZ to close. STL to open. When open, the current master file record will be deleted.
ESWCH	Delete without summary reduction. STZ will cause summary reduction. STL will prevent summary reduction.
ISWCH	Used for inserts with field values. If the ISWCH is ON (non-zero), the status is pre-insert. If it is OFF (zero), the status is pre-field change.
PSWCH	Opened (non-zero) after deleting a parent record type and closed (zero) before writing next parent record, i.e., it is used for deletion of offspring records when a parent record is deleted.
ZERO	This has the value 0.
ONE	This has the value 1.
TWO	This has the value 2.
THREE	This has the value 3.

FOUR This has the value 4.

FIVE This has the value 5.

SIX This has the value 6.

SEVEN This has the value 7.

EIGHT This has the value 8.

NINE This has the value 9.

TEN This has the value 10.

BULOC This is a table of buffer locations for each file. BULOC-n, where n is a file number, contains, in its address, the first word of the buffer for file n.

BUSZE This is a table of buffer size and count. BUSZE-n, where n is a file number, contains, in its address, the number and, in its decrement, the effective size of the buffers for file n.

FILn where n is a two digit file number. The address contains the first word of the file control block for file n.

The following IOCS symbols may also be used:

IOCS	CLOSE	COPY	BSF
DEFINE	OPEN	REW	CKPT
JOIN	READ	WEF	STASH
ATTACH	WRITE	BSR	MWR

Index Registers

All index registers may be used by the hand calculations programmer. They are reset to their proper value automatically upon exit from the calculation. For convenience, they are set on entry as follows:

Pre- or Post-calculations:

IR1 Contains the complement of the address of the first word of the change record.

IR2 Contains the complement of the address of the first word of the master file record.

IR4 Contains no usable information.

Edit calculations:

IR1 Contains the complement of the address of the first word of the change record.

IR2 Contains no usable information.

IR4 Contains no usable information.

Limitations On Hand Calculations

1. Each hand calculation is limited to 2040 instructions. GET, PUT, and similar macro-instructions are expanded to several instructions. Expressions of the nature

```
        STORE   BSS     100
```

or

```
        RMRK    BCI     8,xx...xx
```

count against the 2040 instructions (in these examples, 100 and 8 words respectively). There must be an executable instruction immediately following the "C" constants.

SHARE 709 Symbolic Coding Form

The SHARE 709 Symbolic Coding form is shown in Figure 6. This form is used according to the normal SCAT rules, with the additional requirements that each line must contain a C in column 1 and the variable field must begin in column 16.

Examples of Hand Calculations

Given below are examples of four common uses of File Processor hand calculations: uphill insertion, intermediate parent deletion, reading an auxiliary file, and including subroutines.

Uphill Insertion

Normally a parent record is inserted and then the offspring records are inserted; however, it is sometimes desirable to insert an offspring record and have it cause the parent records to be inserted. This is known as uphill insertion and may be accomplished by using hand calculations.

In uphill insertion, the change record has a dual purpose:

1. Modify or insert a detail record.

2. Insert the parent records of the detail record if they are not present in the master file.

Uphill insertion can be accomplished using either horizontal or update change input. Following is an example using update change input.

Suppose a file has three levels of records, i.e., three levels of sequencing information. Level 3 records will be updated from the change records. Level 1 and level 2 records will be inserted as skeletons only, and they must have indicative information supplied on subsequent runs, that is, on a delayed basis.

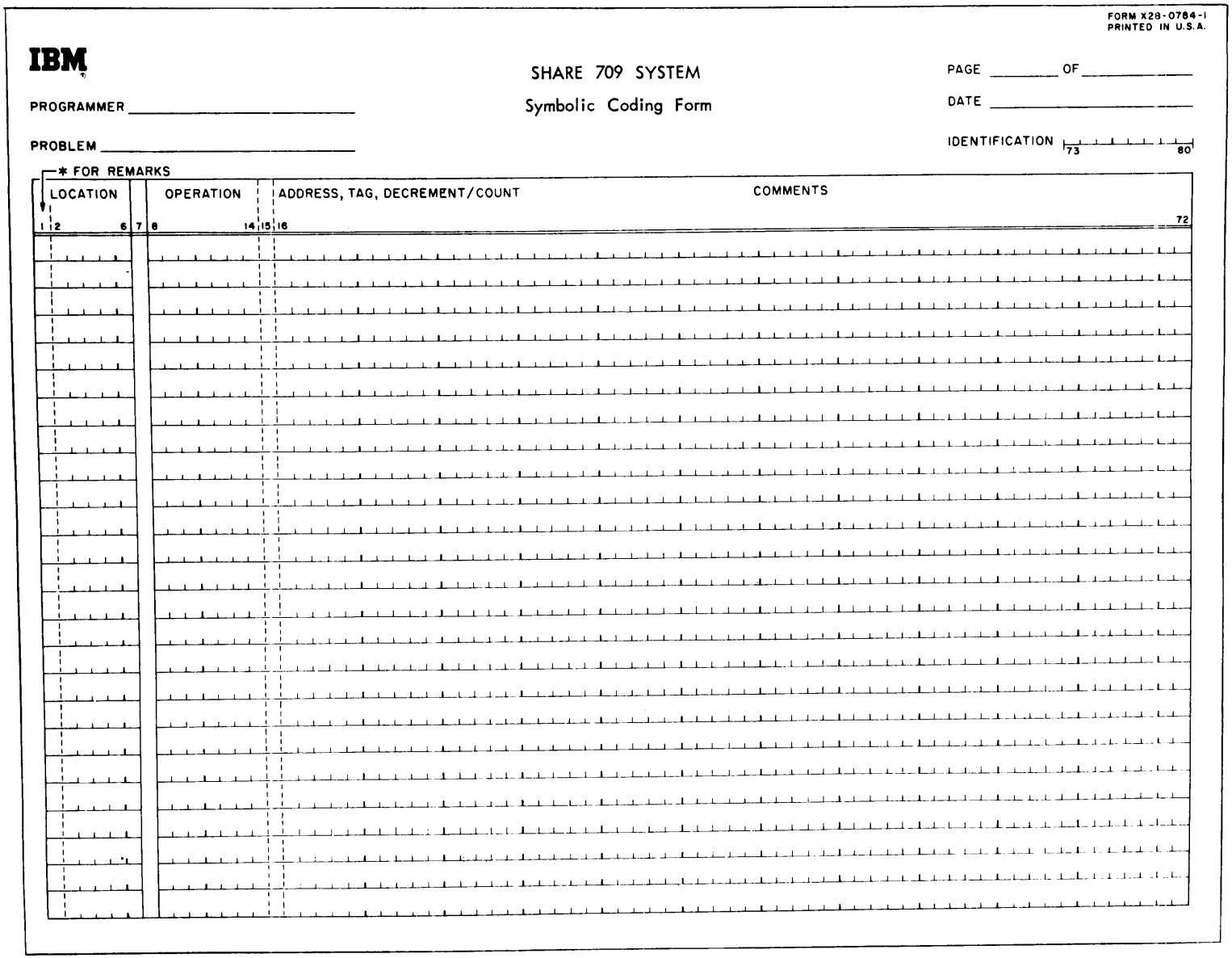


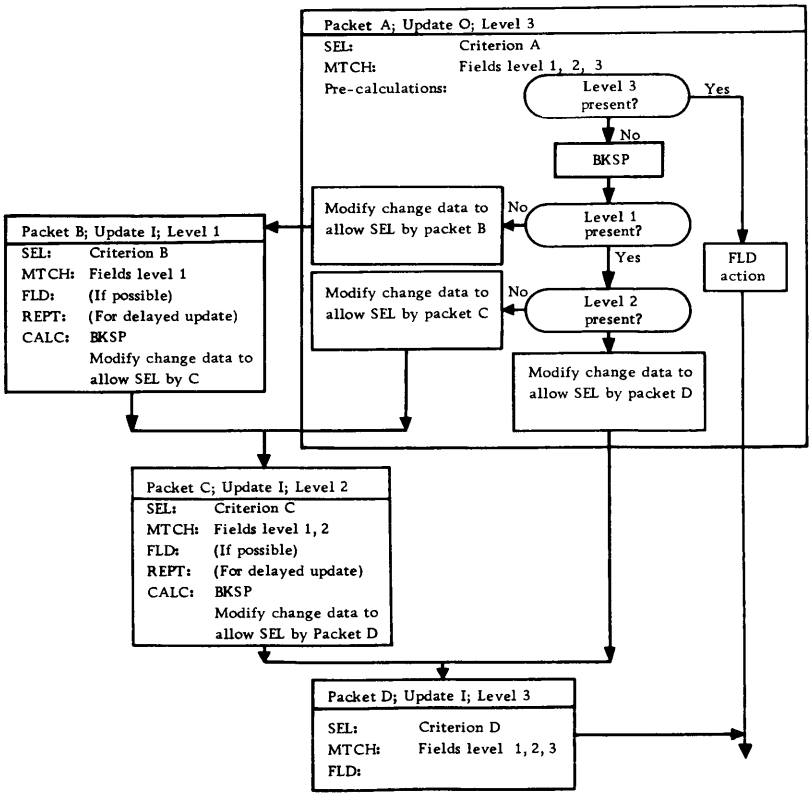
Figure 6. SHARE 709 Symbolic Coding form

A schematic diagram of the suggested method is shown below. In this example, packet A has a somewhat complex hand calculation, which is used, if the level 3 record is not present, for the purpose of determining whether its parents are present. If they are not, they are inserted prior to the insertion of the level 3 record.

If the level 3 record is present, it is assumed that its parents are also present, and the level 3 record is simply updated.

Some details of coding in the packet A pre-calculation are as follows:

```
        NZT     ISWCH       Is level 3 present? (Is ISWCH closed?)
        TRA     MLEV3       Yes
        STZ     ISWCH       No, close ISWCH
        BKSP
        GET     (RT, FLD# corresponding to sequence level 1 of master file)
        CAL     (X, 1 corresponding to sequence level 1 of change file)
         :      (Mask and shift as necessary)
        LAS     BCDBF
        TRA     INSL1       Level 1 missing
        TRA     *+2
        TRA     INSL1       Level 1 missing
         :
INSL1   CAL     C000N       Modify change data to allow selection
        ORS     Y,1         by packet B
        IGNORE              Exit
MLEV3   NOP
```



In the above example, the program does not actually transfer directly from one priority packet to another; exit is made from each priority packet to point 1.0 as shown in the object program flow charts in the appendix. Note that in packet A, when level 3 is not present, exit from the packet is made directly from the pre-calculation by use of the IGNORE macro-instruction.

Intermediate Parent Deletion

When there are several levels of parent records and the highest level parent record is deleted, all of the offspring records are deleted. However, for an intermediate parent (i.e., not the highest level parent) deletion, only that record is deleted — offspring records are not deleted. Deletion of an intermediate parent requires that the offspring records be deleted. A hand calculation may be used for this purpose.

Suppose a file is organized such that record type 10 is a parent of record type 20, which is a parent of record type 30. Following are the hand calculations required to cause intermediate parent deletion.

Vertical packet XX for record type 10:

```
C0001   PZE                 Establish a switch
        STZ       C0001     Reset switch for all records
```

Vertical packet for record type 20:

```
        STZ       -0XXJ     Reset switch for intermediate
                            parent
```

Update packet with D action for record type 20:

```
        STZ       -0XXJ     Sets switch when intermediate
                            parent is deleted
```

Vertical packet with D action for record type 30:

```
        ZET       -0XXJ     Test switch
        TRA       ON        Transfer for deletion if switch set
        IGNORE              Exit from packet if switch reset
ON      NOP                 Entry to deletion packet
```

Notes:

1. All records must enter the first two packets and the last packet; no selection is permitted.

2. The third packet must have a higher priority number than the second packet, since it must be entered first.

3. The calculation in the last packet must be a pre-calculation.

4. The third packet must have selection and/or matching.

It can be seen that a switch is reset to prevent deletion of record type 30 each time a record type 10 or 20 is encountered. When a change record which meets selection and matching criteria is processed, it causes deletion of record type 20 and sets the switch for deletion of record type 30. As each record type 30 is processed, the switch is tested and either the packet is ignored (i.e., the record is not deleted) or the packet is entered (i.e., the record is deleted).

Reading an Auxiliary File

Assume that it is desired to read a tape record each time the calculation is entered. Assume also that the auxiliary file has been defined as file 005 by an *FILE card.

```
C0001   PZE                     Run in switch
C0002   SVN     1,0,2           List — reserve file assumed
C0003   PZE     FIL05
        ZET     C0001           Test run-in
        TRA     RDNXT
        STL     C0001           Turn run-in switch to run
        AXT     ERR,4           } Required since symbolic
        SXD     RDNXT+2,4       } decrements are not allowed
        AXT     5,1             }
        CAL     BULOC,1         } To initialize calling sequence to IOCS
        SLW     DEF1            } describing available space for data buffers
        SLW     ATT1            }
        CAL     BUSZE,1         }
        SLW     DEF2            }
        TSX     DEFINE,4        }
DEF1    PZE     0               }
DEF2    PZE     0               }
        TSX     ATTACH,4        } File initialization described by IOCS
ATT1    PZE     0               }
        PZE     C0002,0,1       }
        TSX     OPEN,4          }
        PZE     FIL05           }
RDNXT   TSX     READ,4          }
        PZE     FIL05           } Read (decrement stored above)
        PZE     EOF             }
LOCT    IORTN                   }
```

The record can then be located from the address of LOCT, and its size can be obtained from the decrement.

The error routine at location ERR must also be coded.

Subroutines

Subroutines may be included in packet XX and may be referenced by any other packet. The following is the suggested method.

```
        TRA     OUT
SUB1    TRA     SUBA
SUB2    SXA     OUT2,4   ┐
        CAL     1,4      │
        .                │
        .                │  Subroutine #2
        .                │
OUT2    AXT     0,4      │
        TRA     2,4      ┘
SUBA    SXA     OUT1,4   ┐
        .                │
        .                │  Subroutine #1
        .                │
OUT1    AXT     0,4      │
        TRA     1,4      ┘
OUT     NOP
```

The calling sequence for SUB1 (from other packets) would be:

```
        TSX     0XXJ+1,4
        PZE     x           where x is the parameter word.
```

The calling sequence for SUB2 (from other packets) would be:

```
        TSX     0XXJ+2,4
```

The first and last instructions cause a transfer around the subroutines when the calculation is entered normally.

CHAPTER 6: DIAGNOSTIC AIDS

In addition to the change and error report file, there are two other aids for diagnosing File Processor source and object program errors: object program symbolic listings and source program diagnostic messages.

Object Program Symbolic Listings

During File Processor generation, standard program modules, or logical blocks, are assembled in the amount and variety needed to perform the necessary functions. By means of an L (Listing) in column 22 of the *JOB card, and the setting DOWN of appropriate Entry Keys on the 7090 Console, a listing of any or all of the assembled program modules may be obtained.

The Entry Keys, and the program modules whose listings they control, are given below in the Object Program Modules table.

To provide an opportunity to set the Entry Keys, the following message is printed on-line and is accompanied by a machine halt:

```
SET KEYS FOR FP SYMBOLIC LISTING
OPERATOR ACTION PAUSE ...
```

After the keys are set, the Start button must be depressed to resume processing.

SORTT Table

Each type of program module has an associated Sort code, which is also given below in the Object Program Modules table. During generation, the Sort code of each program module that is used is entered into a table called the SORTT table.

Each entry in the SORTT table consists of four words, as follows:

Word 1. Sort code (8 octal digits) followed by 0001_8.

Word 2. Address contains the location of the first word of the module; decrement contains the number of words in the module.

Word 3. Sort code (same as word 1) followed by 7000_8.

Word 4. Address contains the location of the first word of storage (masks, constants, etc.) associated with this module; decrement contains the number of words of storage for this module.

Thus, the SORTT table is a memory map of the object program; it can be used to identify the function and location of each object program module in a memory dump. At object program time, the SORTT table is in upper memory.

Table of Object Program Modules

Codes used in octal digits 3-6 of the Sort code are:

RT	Two BCD characters, representing record type.
PN	Two BCD characters, representing major priority numbers.
U	Two BCD characters, representing a file number.
ypr	Three four-bit binary numbers; y is a file number; pr is a major priority number.
y	One four-bit binary number; y is a file number.

Entry Key	Sort Code	Frequency of This Module	Function of Generated Program
1	16 y00 00	1/Input unit	Read input tape drive y.
1	16 ypr 00	1/Priority packet	Test if SEL (and/or NOT) satisfied.
1	16 y77 00	1/Input unit	Counts non-selects or tests run-in switch.
2	16 ypr 07	1/Priority packet	Build file sequence number from MTCH cards. Check sequence on this unit and reject out-of-sequence records, if necessary. Set key to open entry switch for this priority packet.
3	25 0000 00	1/Run	Build sequence number for summary records, if required. Read a master file record.
4	25 0000 07	1/Run	Move a master file record from input to output buffer (with changes, if required).
5	27 0000 00	1/Run	Determine record type of master file record just read.
5	27 RT 00	1/Master file record type	Build sequence number for master file record just read (on basis of dictionary form sequence level). Open or close summary record switch.
5	27 7777 00	1/Run	Open summary identification switch on run-in. Sequence check master file input and stop on equal or low.
6	33 0000 00	1/Run	Open entry switches for all vertical packets.
7	34 7777 00	1/Run	Compare input change sequence number to master file sequence number and set high, equal, or low. On low — close entry switches for all vertical packets. On equal — test for end of job.

Entry Key	Sort Code	Frequency of This Module	Function of Generated Program
8	34 7777 07	1/Run	Transfer to end-of-job routines.
9	40 PN 21	1/Sumreset	Test whether this sumreset packet applies to this master file record type.
10	40 PN 24	1/Sumreset	Test whether SEL (and/or NOT) satisfied.
11	40 PN 44	1/Sumreset	Reset proper field values.
12	40 PN 65	1/Sumreset	Reject subroutine for this packet.
13	43 RT 00	1/Master file sumreset record type	Add or insert summed values to proper field in this summary record type.
14	52 PN 21	1/Priority packet	Test whether this priority packet applies to this master file record.
15	52 PN 24	1/Vertical packet	Test whether SEL (and/or NOT) satisfied for this vertical or sumreset packet.
16	52 PN 27	1/Report	Build and write change report for "before" change case.
17	52 PN 41	1/Pre-calculation	Pre-calculation, if present.
18	52 PN 43	1/I or update	Control ISWCH setting. Write Insert Equal error, if necessary.
19	52 PN 44	1/Priority	Do what is to be done for this package (e.g., FLD card on updates, VAL card on verticals, set D and E switches on horizontal D and E).
20	52 PN 45	1/Vertical or update D or E	Set DSWCH and ESWCH as required.
21	52 PN 47	1/Post-calculation	Post-calculation, if present.
22	52 PN 62	1/Report	Build and write change report for "after" change case.
23	52 PN 65	1/Priority packet	Subroutine to build and write any error or reject records from this packet.
24	61 0000 00 61 RT 00	1/Run 1/Summed record type	Write or delete on master file record. Subtract values of deleted fields from summed amounts.

Entry Key	Sort Code	Frequency of This Module	Function of Generated Program
25	63 RT 00	1/Master file record type	Insert a skeleton record type. Open or close summary record switches.
26	63 7777 77	1/Run	Common insert — backspace master file input, close D and E switches.
27			Not used.
28	65 U00 03	1/File Unit	Three word entry for each record type on file: Word 1 contains record type in BCD Word 2 has type location in address, record length in decrement Word 3 is used for output: Decrement is location of lower record type for "look back" Sign is set to minus to show that record of this type is in current output block
29	65 1000 07	1/File	Eight word entry for each priority packet: Word 1 contains priority number Word 2 contains pseudo unit number Word 3 contains count of rejects Word 4 contains count of "before" change reports Word 5 contains count of "after" change reports Word 6 contains count of selects Word 7 contains count of backspaces Word 8 is a spare word
29	65 0700 07	1/File	Twelve word count table for each record type in master output file: Word — Description 1 — Record type 2 — Read 3 — Backspace 4 — Insert 5 — Insert 6 — Insert 7 — Delete D 8 — Delete E 9 — Deleted from parent 10 — Written 11 — Spare 12 — Spare
30			Not used.

Entry Key	Sort Code	Frequency of This Module	Function of Generated Program
31			Not used.
32			Not used.
33		1/Run	HRDWR, SNTRY, ONTRY, VNTRY, and SORTT tables. These describe the allocation of memory.
34	16 y 0047	1/Edit Calculation	Edit calculation.
35			Not used.
-	00 0000 00	1/Run	Fixed object program.
-	00 0000 10	1/Run	Tables associated with input/output subroutines.
-	00 0000 20	1/Run	Object summary tables (3 word entries): Word 1 contains record type and field number of field being summed Word 2 contains record type and field number into which field is being summed Word 3 contains location of intermediate accumulating location
-	00 0000 30	1/Run	Summary words, sequencing buffers, and packet entry switches.

File Processor Source Program Diagnostic Messages

The File Processor scans all input packets for detectable errors. There are two possible types of error messages; they are the Code errors and the No Recovery errors.

Code Errors

The diagnostic message CODE ERROR NNN indicates a parameter card error. NNN represents a number in the Code Error list found below. On the line following the Code Error message, the contents of the parameter card just read will be printed. The error may be in this card, in the group of function cards immediately preceding, or, if the card just read is a packet header, in the preceding priority packet. Generation will not cease immediately due to this type of error but will continue until the following on-line message is printed:

PROCESSING HALTED DUE TO EDIT ERRORS INDICATED

Execution of the object program is not advisable.

No Recovery Errors

These errors are indicated by the message NO RECOVERY NNNN, where NNNN represents a number in the No Recovery Error list found below. An error of this type causes generation to cease immediately. There are three types of no recovery errors, as follows:

1. Parameter Card Error. This error type can be identified by the message PARAMETER ERROR and by an error number in the 3000-3999 range. On the line following the message, the contents of the parameter card just read will be printed. The error may be in this card, in the group of function cards immediately preceding, or, if the card just read is a packet header, in the preceding priority packet. These stops are caused by errors very similar to the Code errors.

2. Unexplained. This error type is indicated by the message UNEXPLAINED, TRY AGAIN and by an error number in the 4000-4999 range. Start the job again from the beginning.

3. Exceed System Limitations. This error type can be identified by the message GENERATOR LIMITATION and by an error number in the 5000-5999 range.

Code Error List

The following list pertains to Code errors; these are caused by errors in the input parameter packets. The error message is

Line 1: CODE ERROR NNN
Line 2: Parameter card last read

NNN is the error number in the list below. Generation of the object program will continue. However, execution of the object program is not advisable.

1 Update, vertical, or horizontal function should be, but is not, MTCH, SEQ, SEL, NOT, FLD, VAL, CALC, REPT, SELF, NOTF, or these cards are not grouped properly.

7 Field referred to is not in master file dictionary.

8 Field referred to is not in master file dictionary.

12 Field referred to is not in update with dictionary file dictionary.

16 Field referred to is not in master file dictionary.

17 Field referred to is not in master file dictionary.

25 Record type referred to is not in record type table (table is empty, indicating no record types have been created). Check that dictionary packet for this record type is present.

26 Record type referred to is not in record type table (there are entries in table, indicating at least one other record type has been created). Check that dictionary packet for this record type is present.

101 Record type being summed into (record type on dictionary SUMM card — columns 17-18) is not in Summary table. Check dictionary of summary record type.

102 Dictionary form parent code (columns 21-22 of header card) should be, but is not, blank or numeric.

103 Parent code on dictionary form (columns 21-22 of header card) is not in record type table. Record type referred to as parent has not been created or has not been designated as a parent.

104 Parent code on dictionary form (columns 21-22 of header card) should be lower numerically than master file record type (columns 16-17 of same card), but is equal or higher.

105 REPT function card follows DICT or SUMM function card. This is invalid.

106 Major priority (columns 1-2) of dictionary card should be greater than or equal to major priority of previous dictionary card, but is lower.

108 DICT card record type (columns 10-11) should, but does not, equal dictionary header record type (columns 16-17).

109 DICT card field number (columns 12-15) should be greater than, but is equal to or lower than, previous DICT card field number.

110 Function (columns 6-9) of card following dictionary header card should be, but is not, DICT.

111 Field number (columns 12-15) of dictionary SUMM card should be equal to field number of previous DICT card, but is lower.

112 This card and previous card are dictionary SUMM cards and record type and field number (columns 17-22) into which this card is being summed should be, but are not, greater than record type and field number into which the previous card is being summed.

113 DICT card record type (columns 10-11) and field number (columns 12-15) should be numeric, but one or both of them is not.

114 Dictionary packet having delete action (D in column 20 of header card) should only have a header card, but has other cards before next header card.

115 DICT card field size (columns 17-19) should be, but is not, numeric.

116 Field designated as a sequence field on a previous File Processor run is having its mode (columns 20-22 of DICT card) changed. This is invalid.

117 Field action (column 16 of DICT card) error in dictionary mode change.

118 DICT card field size (columns 17-19) should be, but is not, divisible by six when mode (columns 20-22) is 100 or 110 (BCD).

119 DICT card field size (columns 17-19) should, but does not, agree with those allowed when mode (columns 20-22) is 001 (binary).

120 DICT card minimum and maximum range (columns 27-38) should be numeric or blank, but one or both of them is not.

122 DICT card having field number 0001 (columns 12-15) should, but does not, have a numeric sequence level (columns 23-24).

123 DICT or SUMM card should, but does not, have 0 or 1 as a replace only code (column 47).

124 DICT card field size (columns 17-19) should be, but is not, divisible by six when mode (columns 20-22) is 100 or 110 (BCD).

125 DICT card field size (columns 17-19) should, but does not, agree with those allowed when mode (columns 20-22) is 001 (binary).

126 DICT card field size (columns 17-19) should be, but is not, numeric.

127 DICT card field size (columns 17-19) should be, but is not, divisible by six when mode (columns 20-22) is 100 or 110 (BCD).

128 DICT card field size (columns 17-19) should, but does not, agree with those allowed when mode (columns 20-22) is 001 (binary).

130 DICT card sequence level (columns 23-24) should be, but is not, numeric or blank.

131 DICT or SUMM card should, but does not, have 0 or 1 as a replace only code (column 47).

132 First character of DICT card minimum range (columns 27-32) should be, but is not, numeric or blank.

133 SUMM card record type and field number (columns 17-22) being summed into should be numeric, but one or both of them is not.

134 SUMM card record type (columns 17-18) being summed into should be, but is not, greater than record type (columns 10-11) being summed.

135 Dictionary form header card record type (columns 16-17) should be, but is not, numeric.

137 Dictionary form parent code (columns 21-22 of header card) should be, but is not, blank or numeric.

138 DICT card sequence level (columns 23-24) should be, but is not, numeric or blank.

139 DICT card field size is zero.

301 MTCH card change field size should be, but is not, equal to master file field size.

302 MTCH card change field size should be, but is not, smaller than or equal to buffer size.

303 Number represented by n in SEQn was previously used in this packet. Only one field may be used at a given level with SEQn.

304 There should be, but there is not, a MTCH card for every sequence level used in this record type. A MTCH card is needed not only for the levels specified in the dictionary of this record type, but also for any sequence levels taken from its parents.

305 MTCH card record type should be, but is not, equal to or numerically smaller than header card record type of this packet.

306 Field used on MTCH card has been defined as being in the binary mode. This is invalid.

307 MTCH card change field size should be, but is not, equal to master file field size.

308 Field used on MTCH card has been defined as being in the binary mode. This is invalid.

309 MTCH card change field size should be, but is not, equal to master file field size.

310 There is more than one SEL (or NOT) on same horizontal change field increment (columns 10-12) and change field size (columns 13-15). This is invalid.

311 The number of characters used in the horizontal change selection value or range is determined by the change field size (columns 13-15). The character immediately following the last valid character should be, but is not, a blank (for a value) or a minus sign (for a range).

312 The number of characters used in the horizontal change selection value or range is determined by the change field size (columns 13-15). The character immediately following the last valid character should be, but is not, a blank (for a value) or a minus sign (for a range).

313 A literal field number (columns 17-20) has been used twice on a change report form that is reporting horizontal changes.

316 The literal field number (columns 17-20) on a change report form that is reporting horizontal changes should be, but is not, AUTH, TYPE, FLD#, ACTF, ACTR, or DESC.

317 There should be, but there is not, a MTCH card for every sequence level used in this horizontal insert (I) record type. A MTCH card is needed not only for the levels specified in the dictionary of this record type, but also for any sequence levels taken from its parents.

318 There should be, but there is not, a MTCH card for every sequence level used in this update insert (I) record type. A MTCH card is needed not only for the levels specified in the dictionary of this record type, but also for any sequence levels taken from its parents.

319 There should be, but there is not, a MTCH card for every sequence level used in this update (O) record type. A MTCH card is needed not only for the levels specified in the dictionary of this record type, but also for any sequence levels taken from its parents.

320 Update with dictionary card has change field length (columns 26-28) but no change field increment (columns 23-25).

401 Update (with or without dictionary) record action (column 23 of header card) should be, but is not, D, E, M, O or I.

402 Update (without dictionary) form MTCH, FLD, or SEQn card record type (columns 10-11) should be equal to or numerically less than header card record type (columns 12-13), but is greater.

403 Update without dictionary field action (column 22) should be, but is not, R for field which has been designated as BCD (columns 20-22 of dictionary form).

404 Update without dictionary change field size (columns 19-21) should, but does not, equal master file field size.

405 Update without dictionary change field size (columns 19-21) should equal, but is larger than, master file field size.

406 Update without dictionary field action (column 22) should be, but is not, R, +, or - for field which has been designated as binary (columns 20-22 of dictionary form).

407 Update without dictionary field action (column 22) should be R, but is + or -, for field which has been designated as replace only (1 in column 47 of dictionary form).

408 There is more than one SEL (or NOT) on same update without dictionary change field increment (columns 10-12) and change field size (columns 13-15). This is invalid.

409 The number of characters used in the update without dictionary change selection value or range is determined by the change field size (columns 13-15). The character immediately following the last valid character should be, but is not, a blank (for a value) or a minus sign (for a range).

410 The number of characters used in the update without dictionary change selection value or range is determined by the change field size (columns 13-15). The character immediately following the last valid character should be, but is not, a blank (for a value) or a minus sign (for a range).

411 Dictionary card field size (columns 17-19) should, but does not, agree with those allowed when mode (columns 20-22) is 001 (binary).

412 Update without dictionary FLD function (columns 6-9) has been used with field designated as a sequence field (columns 23-24 of dictionary form). This is invalid.

413 AUTH has been used in literal field number (columns 17-20) of change report form that is reporting update changes. This is invalid.

414 FLD# has been used in literal field number (columns 17-20) of change report form that is reporting update changes. This is invalid.

415 DESC has been used in literal field number (columns 17-20) of change report form that is reporting update changes. This is invalid.

416 ACTF has been used in literal field number (columns 17-20) of change report form that is reporting update changes. This is invalid.

417 FROM has been used in literal field number (columns 17-20) of change report form that is reporting update changes. This is invalid.

418 TO has been used in literal field number (columns 17-20) of change report form that is reporting update changes. This is invalid.

419 A change report field number (columns 13-16) has been used twice within the same major priority number (columns 1-2). This is invalid.

420 Change report record type and field number (columns 11-16) should be numeric when reporting update changes, but one or both of them is not.

500 Update with dictionary form MTCH, FLD, or SEQn card record type (columns 10-11) should be equal to or numerically less than header card record type (columns 12-13), but is greater.

501 Update with dictionary field action (column 22) should be, but is not, R for field which has been designated as BCD (columns 20-22 of dictionary form).

502 Update with dictionary change field size should be, but is not, equal to master file field size. Change record type and field number (columns 16-21) have been used, but change field increment and length (columns 23-28) have not been used. (Entire change field is being used.)

503 Update with dictionary change field size should be, but is not, equal to master file field size. Change field increment and length (columns 23-28) have been used, indicating only part of change field is to be used.

504 Update with dictionary change field length (columns 26-28) has been used, but change field increment (columns 23-25) has not. This is invalid.

505 Update with dictionary file contains a binary field that has an invalid size. If the update with dictionary file was created using 9PAC, this error condition was probably caused by a machine error, a bad tape, or a bad tape unit.

506 Update with dictionary change field size should be, but is not, equal to master file field size. Change record type and field number (columns 16-21) have been used, but change field increment and length (columns 23-28) have not been used. (Entire change field is being used.)

507 Update with dictionary change field length (columns 26-28) should be equal to or smaller than update with dictionary file field size as established in its dictionary, but is greater.

508 Update with dictionary change field length (columns 26-28) is greater than master file field size. This is invalid.

509 Update with dictionary change field length (columns 26-28) has been used, but change field increment (columns 23-25) has not been used. This is invalid.

510 Update with dictionary change field length (columns 26-28) is greater than master file field size. This is invalid.

511 Update with dictionary field action (column 22) should be, but is not, R, +, or - for field which has been designated as binary (columns 20-22 of dictionary form).

512 Update with dictionary field action (column 22) should be R, but is + or -, for field which has been designated as replace only (1 in column 47 of dictionary form).

513 Mode of field on update with dictionary file is packed binary (11 in columns 21-22). This is invalid. If update with dictionary file was created using 9PAC, this error condition was probably caused by a machine error, a bad tape, or a bad tape unit.

514 Mode of field on update with dictionary file is unsigned binary (010 or 000 in columns 20-22). This is invalid. If update with dictionary file was created using 9PAC, this error condition was probably caused by a machine error, a bad tape, or a bad tape unit.

515 DICT card field size (columns 17-19) should, but does not, agree with those allowed when mode (columns 20-22) is 001 (binary).

516 Update with dictionary change field size should be, but is not, equal to master file field size. Change record type and field number (columns 16-21) have been used, but change field increment and length (columns 23-28) have not been used. (Entire change field is being used.)

517 There is more than one SEL (or NOT) on same update with dictionary change field (columns 12-15). This is invalid.

518 Update with dictionary change field size should, but does not, agree with those allowed when mode is binary. If update with dictionary file was created using 9PAC, this error condition was probably caused by a machine error, a bad tape, or a bad tape unit.

519 The number of characters used in the update with dictionary change selection value or range is determined by the change length (columns 19-21) if used, or if not used, by the dictionary length of the field (in columns 12-15). The character immediately following the last valid character should be, but is not, a blank (for a value) or a minus sign (for a range).

520 The number of characters used in the update with dictionary change selection value or range is determined by the change length (columns 19-21) if used, or if not used, by the dictionary length of the field (in columns 12-15). The character immediately following the last valid character should be, but is not, a blank (for a value) or a minus sign (for a range).

530 The number of characters used in the update with dictionary change selection value or range is determined by the change length (columns 19-21), if used, or if not used, by the dictionary length of the field (in columns 12-15). The character immediately following the last valid character should be, but is not, a blank (for a value) or a minus sign (for a range).

531 Field size on DICT card (columns 17-19) should, but does not, agree with those allowed when mode (columns 20-22) is 001 (binary).

532 FLD function for update with dictionary (columns 6-9) has been used with field designated as a sequence field (columns 23-24 of dictionary form). This is invalid.

601 Record action for vertical packet (column 17) should be, but is not, D, E, or blank.

602 Calculation constant cards (C in column 2) should all be before calculation instruction cards (no C in column 2), but at least one is after an instruction card.

604 There is a character in column 15 of a calculation card. This is invalid.

605 Calculation operation code (columns 8-14) being used is not acceptable to 9PAC.

606 Calculation address (starting in column 16) is more than five BCD characters. This is invalid.

607 Calculation symbolic address has a minus sign. This is invalid.

608 Calculation constant card (C in column 2) has symbolic address. This is invalid.

609 Address of OCT operation in calculation is more than 12 characters. This is invalid; only 6 characters are allowed in the File Processor.

610 Address of OCT operation in calculation is more than 6 characters. This is invalid; only 6 characters are allowed in the File Processor.

611 BCI operation in calculation is requesting 9 or more words. This is invalid; only 8 words are allowed in the File Processor.

612 There should be, but there is not, a comma in column 17 of a calculation card that has operation code BCI.

613 Calculation address has a minus sign as its first character, but is not a reference to storage or to instructions in a calculation having a different major priority number (columns 1-2). This is invalid.

614 Calculation address has a minus sign as its first character, but has less than four characters. This is invalid.

615 The character following J or P in a calculation address which refers to storage or instructions in another calculation should be, but is not, +.

616 The first character of a calculation address is *, but there are no additional characters. This is invalid.

617 Calculation address of GET macro-instruction should contain (in columns 16-17) a record type equal to or numerically less than the record type in the header card of this major priority packet (columns 1-2), but contains a numerically higher record type.

618 Calculation constant card (C in column 2) has GET macro-instruction (columns 8-14). This is invalid.

619 Calculation constant card (C in column 2) has PUT macro-instruction (columns 8-14). This is invalid.

620 Calculation constant card (C in column 2) has IGNORE macro-instruction (columns 8-14). This is invalid.

621 Calculation constant card (C in column 2) has REJECT macro-instruction (columns 8-14). This is invalid.

622 REJECT macro-instruction (columns 8-14) is being used in a vertical packet. This is invalid.

623 Calculation constant card (C in column 2) has ERRFLD macro-instruction (columns 8-14). This is invalid.

624 ERRFLD macro-instruction (columns 8-14) is being used in a packet that is <u>not</u> horizontal. This is invalid. ERRFLD can be used only in horizontal packets.

625 Undefined symbol; calculation symbolic address (starting in column 16) is not a File Processor symbol, nor is it found in the assigned symbols (columns 2-6) in this calculation.

626 Field in calculation address (columns 18-21) of SUM macro-instruction (columns 8-14) has been defined as being BCD (100 or 110 in columns 20-22 of dictionary form). This is invalid.

627 Calculation constant card (C in column 2) has SUM macro-instruction (columns 8-14). This is invalid.

628 Record type (columns 10-11) of vertical SEL (or NOT) card is numerically higher than record type in header card (columns 14-15). This is invalid.

629 There is more than one vertical SEL (or NOT) on the same master file field (columns 12-15). This is invalid.

630 Record type for VAL function (columns 10-11) should, but does not, equal record type of vertical header card (columns 14-15).

631 Field action for vertical change (column 16) should be, but is not, R, S, or T when mode is BCD (100 or 110 in columns 20-22 of dictionary form).

632 Field size on DICT card (columns 17-19) should, but does not, agree with those allowed when mode (columns 20-22) is 001 (binary).

633 Field action for vertical change (column 16) should be, but is not, +, -, R, S, or T, when mode is binary (001 in columns 20-22 of dictionary form).

634 Vertical field action (column 16) should be R, S, or T, but is + or -, for field which has been designated as a replace only field (1 in column 47 of dictionary form).

635 Vertical change value (starting in column 17) for field being altered causes maximum range (columns 33-38 of dictionary form) of this field to be exceeded. This is invalid.

636 Vertical change value (starting in column 17) for field being altered causes contents of this field to be less than minimum range (columns 27-32 of dictionary form) of the field. This is invalid.

637 Field size on DICT card (columns 17-19) should, but does not, agree with those allowed when mode (columns 20-22) is 001 (binary).

638 Calculation address of DEC operation code (columns 8-14) should not be, but is, more than ten characters.

639 Field (columns 12-15) being changed by a VAL card has been designated as a sequence field (columns 23-24 of dictionary form). This is invalid; contents of sequence fields cannot be changed.

640 Calculation constant card (C in column 2) has BKSP macro-instruction (columns 8-14). This is invalid.

641 BKSP macro-instruction (columns 8-14) is being used in a vertical packet. This is invalid.

642 Calculation address of PUT macro-instruction (columns 8-14) should, but does not, contain (in columns 16-17) a record type equal to the record type in the header card of this major priority packet (columns 1-2).

643 Calculation address of SUM macro-instruction (columns 8-14) should, but does not, contain (in columns 16-17) a record type equal to the record type in the header card of this major priority packet (columns 1-2).

644 There are no MTCH cards in previous update or horizontal packet. This is invalid.

645 Vertical field action (column 16) should be, but is not, +, -, R, S, or T.

647 Field size on DICT card (columns 17-19) should, but does not, agree with those allowed when mode (columns 20-22) is 001 (binary).

648 Change report that is reporting horizontal changes should, but does not, have a zero in column 10.

649 Change report record type (columns 11-12) should, but does not, equal horizontal header card record type (columns 16-17) when reporting horizontal changes.

650 Sequence level (columns 23-24 of dictionary form) of a field has been designated as 00. This is invalid.

651 File Processor symbol is being used as a calculation location symbol (columns 2-6). This is invalid.

652 Doubly-defined symbol; a location symbol (columns 2-6) has been used more than once in a calculation. This is invalid.

654 Horizontal change data size should, but does not, agree with those allowed when field affected is binary.

655 Horizontal data work area is exceeded.

660 More than 10 different fields are being tested by a SELF (and/or NOTF) group preceding a FLD card.

664 Selection on field number 0001 is not allowed in vertical packets — action is automatic.

No Recovery Error List

PARAMETER CARD ERRORS

The following list pertains to No Recovery errors caused by errors in the input parameter packets. The error message is:

Line 1: NO RECOVERY NNNN
Line 2: PARAMETER ERROR
Line 3: Parameter card that was last read

NNNN is the error number in the list below. The last digit is represented as X in the error messages below, because this digit will vary depending upon how the error is detected. Generation of the object program will halt.

301X Major priority number of header card should be, but is not, greater than major priority number of previous header card.

302X Invalid function on header card.

303X Major priority should be, but is not, greater than or equal to previous major priority.

304X Record type referred to is not in record type table.

306X There is more than one horizontal packet and, therefore, a select is required in the first horizontal packet.

308X Two or more horizontal packets with no SEL cards in second packet.

309X Record action should be, but is not, I, D, E, or M.

310X Dictionary label should be, but is not, E, F, G, or P. There is an error in the dictionary packet (such as no header or DICT cards) causing an error in the assignment of the executive codes.

311X Mode of master file field should be packed or unpacked BCD or unpacked binary. The packed binary mode is invalid at this time.

312X Improper change action code; I action is being used to try to insert a dictionary for a field that already exists.

313X A SUMM card is trying to sum a field that has been deleted.

314X The field being summarized is in BCD. It must be in binary.

315X SUMM card record type does not equal previous DICT card record type.

316X Dictionary record action of summary record should be, but is not, D or I.

317X Improper change action code in the summary record. I action is being used to try to insert a dictionary for a field that already exists.

318X No dictionary on master file.

320X Input unit for update with dictionary has no dictionary.

321X Input unit for update with dictionary does not have record type requested.

325X Parent record type specified in dictionary header card does not exist in file.

326X Parent record type specified in dictionary header card is equal to or greater than record type of this packet. The parent record must have a lower numeric record type.

327X Record type should be, but is not, in field number 0001.

328X Sequence level is not in the sequence level table.

329X Missing sequence level in master file. All sequence levels specified must be consecutive.

330X Sequence level table is out of order. All sequence levels specified must be consecutive.

331X This record type has a parent, but no sequence identification.

332X Parent code referred to is not in record type table.

333X Parent is missing a sequence level. All sequence levels, up to and including record type, not found in the offspring must be in the parent.

334X This record type has no parent and no sequence identification.

335X This record type has a missing sequence level, but no parent.

336X Record type is not in sequence level table. Check DICT card for sequence level entry for this record type.

337X Sequence levels should be, but are not, consecutive.

338X Dictionary record action should be, but is not, D, I, M, or O.

339X Master file record types should be, but are not, in ascending numerical order.

340X Field number on DICT card should be, but is not, greater than field number on previous DICT card. Field numbers must be in ascending numerical sequence.

341X Dictionary change record type, should be, but is not, greater than or equal to previous master file record type.

342X Mode of master file field should be, but is not, packed BCD (110), unpacked BCD (100), or unpacked binary (001).

343X Dictionary field action should not be M for size change.

344X SUMM card field number should, but does not, equal previous DICT card field number.

345X There is no dictionary for the summary field.

347X Non-numeric sequence level used. This is invalid.

348X Dictionary field action for summary record should be, but is not, I.

349X Record type being summarized into has not been established. Check dictionary of summary record.

350X Incorrect dictionary size (BCD field is not multiple of 6 or binary field does not agree with prescribed table sizes).

351X Invalid function. Should be, but is not, CALC, MTCH, SEL, VAL, FLD, or REPT.

352X Symbolic code is not in SORTT. Check inter-packet references in hand calculations.

353X There should be, but there is not, a dictionary for the group header record type.

354X Record type referred to is not in summary table (SUTBL).

355X BCD field should be, but is not, 12 characters or less for mode change.

356X Two fields with the same sequence level.

UNEXPLAINED ERRORS

The following list pertains to No Recovery errors caused by the File Processor system, machine malfunction, tape trouble, or invalid dictionary on update change tape. The error message is:

Line 1: NO RECOVERY NNNN
Line 2: UNEXPLAINED, TRY AGAIN THEN DUMP CORES

NNNN is the error number in the list below. The last digit is represented as X in the error messages, because this digit will vary depending upon how the error is detected. Generation of the object program will halt. Start again.

401X End of tape while writing change output dictionary.

402X Unexpected end of file on parameter input.

403X Invalid EOF on master output tape.

404X Invalid end of tape on master output tape.

405X Invalid EOF on system tape.

407X Invalid EOF on scratch tape.

408X End of tape while writing output file dictionary.

409X Exceeds previously checked record type table.

410X Exceeds previously checked other entry (ONTRY) table.

411X Exceeds previously checked vertical entry table.

412X Program expected, but did not find, EOF on system tape.

413X Buffer reference to sequence level table is invalid.

414X Impossible error if sequence level is used.

415X Impossible error in merging dictionary labels using dictionary changes.

416X Dictionary field mode less than zero.

418X There should be, but there is not, a P label on tape unit for update with dictionary.

419X Executive code should be, but is not, E, F, G, or P on tape unit for update with dictionary.

420X Summary entry is not in summary table (SUTBL).

421X Table address is invalid.

422X First trailing identification buffer exceeds buffer limitation.

423X No executive E record on master output file during processing.

424X Parent is not in record type table on master output file.

425X Record executive code should be, but is not, F on master output file.

426X Record type is not in field number 0001 on master output file.

427X Sequence level is greater than a BCD blank on master output file.

428X Bit number is not zero.

429X Record length is not equal to skeleton set up earlier.

430X Executive code at end of dictionary is greater than P on master output file.

432X No associated storage in sort table (SORTT).

433X Pseudo read area is not in sort table (SORTT).

434X Input/output tables are not in sort table (SORTT).

435X Address table (ATBL) instruction count exceeds program block.

436X PADDR exceeds program block.

437X SADDR entry exceeds storage.

438X Invalid GADDR code (not 2, 3, or 4).

439X GADDR code 2 not found in sort table (SORTT).

440X GADDR code 3 not found in sort table (SORTT).

441X GADDR code 4 not found in sort table (SORTT).

442X Scratch tape sort code not in sort table (SORTT).

443X Sort code not in print mask table.

444X Error in select table during preprocessing.

445X Horizontal select zero table entry.

446X Invalid select starting point.

448X Select should, but does not, have a GADDR code 3.

449X Sort code not found in SUB SORT table.

450X No GADDR block found.

451X Executive code P encountered without a previous executive code E during preprocessing.

452X BCD to binary conversion error in field size.

453X BCD to binary conversion error in word number.

454X BCD to binary conversion error in bit number.

455X BCD to binary conversion error in minimum range value.

456X BCD to binary conversion error in maximum range value.

457X There should be, but there is not, a zero at the end of the summary table (SUTBL).

459X There should be, but there is not, a zero at the end of the sequence level table.

460X There should be, but there is not, a zero at the end of the summary entry table (SNTRY).

461X Record type in sequence level table is not in record type table.

464X Executive code should be, but is not, E, F, G, or P on master output file.

465X There should be, but there is not, a zero at the end of the vertical entry table (VNTRY).

466X Invalid missing address (MADDR) code.

467X Exceed packed BCD table previously checked.

468X Non-vertical priority is not in other entry (ONTRY) table.

469X Priority not in other entry (ONTRY) table.

470X Update priority not in other entry (ONTRY) table.

471X Horizontal priority not in other entry (ONTRY) table.

472X Update field previously established in summary record dictionary is not in summary table.

473X Update with dictionary field previously established in summary record dictionary is not in summary table.

474X Vertical field previously established in summary record dictionary is not in summary table.

475X Bit number error.

476X Summary table (SUTBL) has been altered after it was previously checked.

477X It is not address, tag, or decrement time.

480X Step down error on internal move.

482X EOF not expected on input master.

483X End of tape not expected on output master.

484X End of tape not expected on change output file.

485X End of file not expected on master output file.

ERRORS CAUSED BY EXCEEDING SYSTEM LIMITATIONS

The following list pertains to No Recovery errors caused by exceeding File Processor system limitations. The error message is:

Line 1: NO RECOVERY NNNN
Line 2: GENERATOR LIMITATION

NNNN is the error number in the list below. The last digit is represented as X in the error messages below, because this digit will vary depending upon how the error is detected. Generation of the object program will halt.

501X Exceed non-sequence match table (30 words, 5 words per entry). MTCH function is allowed on only 6 fields that are not designated as sequence determining fields.

502X More than ten different fields are being tested by a SEL or NOT in a horizontal packet. Only ten selection fields are allowed.

503X More than ten different fields are being tested by a SEL or NOT in a vertical packet. Only ten selection fields are allowed.

504X More than 75 vertical and sumreset packets (combined) have been used.

505X Exceed already defined table (800 words, 2 words per entry). Only 400 symbolic locations (including those beginning with C) are allowed per packet.

506X Exceed not yet defined table (800 words, 2 words per entry). Only 400 symbolic locations, not beginning with C, are allowed if no symbolic locations beginning with C have been used.

507X More than 99 sequence levels have been used.

508X Exceed unpacked BCD table (3500 words, number of words per entry depends on format of file).

509X Exceed packed BCD table (3500 words, number of words per entry depends on format of file).

510X Exceed unpacked binary table (5000 words, number of words per entry depends on format of file).

511X Exceed packed binary table (one word, mode is invalid).

512X Exceed object program (OBPGM) table (2045 words).

513X Exceed storage table (798 words).

514X Exceed address table (ATBL1 — 998 words).

515X More than 75 update and horizontal packets (combined) or 15 horizontal packets have been used.

517X Change dictionary exceeds ENDIC table (4400 words, 11 words per entry).

518X Exceed record type table. Only 99 record types are allowed.

519X Exceed horizontal select table (62 words, 4 words per entry). Only 15 SELs or NOTs (combined) are allowed in one horizontal packet.

520X Exceed parent table. Only 20 parents are allowed.

521X Exceed summary table (450 words, 3 words per entry). Only 150 SUMM cards may be used in a File Processor run.

522X Exceed program identification table.

523X Exceed program size table.

524X Exceed associated storage identification table.

525X Exceed associated storage size table.

528X Exceed dictionary table (2000 words, 5 words per entry).

530X More than 10 record type deletions. Deletions may only be performed on 10 record types during one File Processor pass.

531X Exceed merge table (MGTBL).

533X Exceed vertical select field table. Only 50 SELs or NOTs are allowed in one vertical packet.

534X Exceed record table in I/O (200 words).

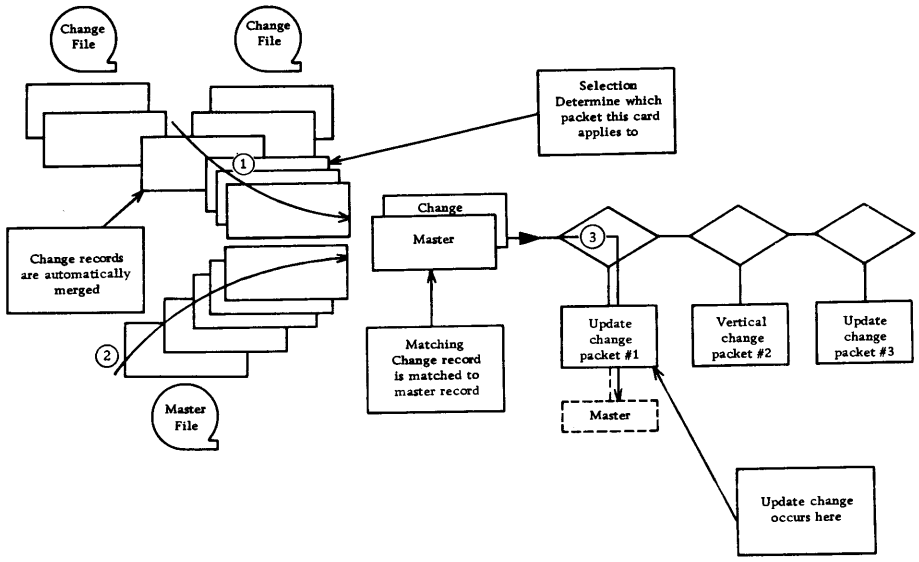
535X Insufficient storage for input/output.

590X Exceeds group header table.

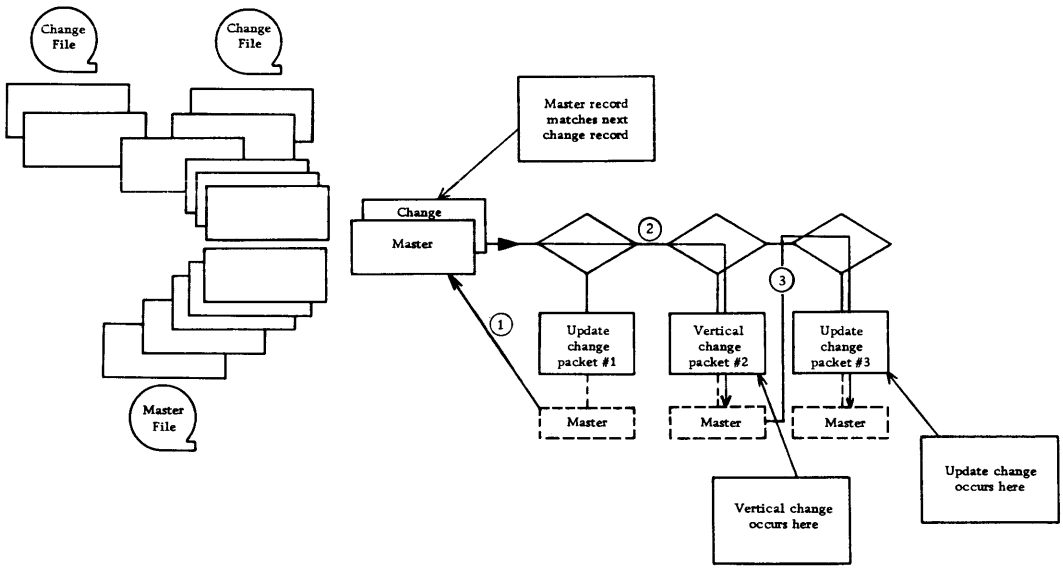
APPENDIX A: OBJECT PROGRAM FLOW CHARTS

Following is a schematic diagram of a sample File Processor object program which will help the reader to understand the object program flow charts. For the sake of simplicity, insertions and deletions are ignored as is the change and error report file.

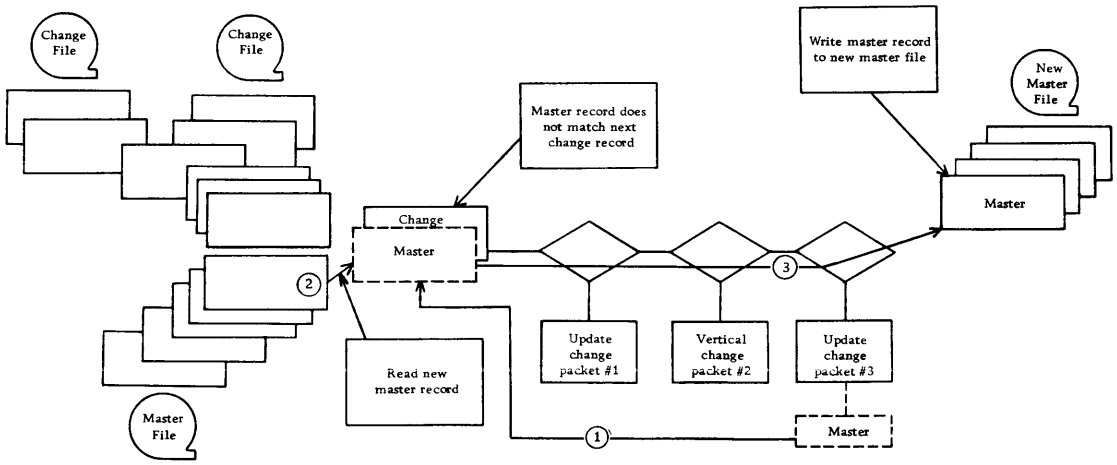
In the sample problem there are two change input files and three priority packets.



In the diagram above, the first change record matches the first master file record. The change record selects priority packet 1 where the first change to the master file record is made.



The master file record is again compared with the merged change file, this time with the second change record. The second change record also matches the first master file record but it selects priority packet 3. However, prior to entry into packet 3, packet 2 is examined. Packet 2 is a vertical change which applies to the same master file record, so this packet is entered and executed prior to entry into packet 3. Subsequently, priority packet 3 is entered and the master file record is appropriately modified.



The third change record does not match the master file record, which is consequently written to the new master file.

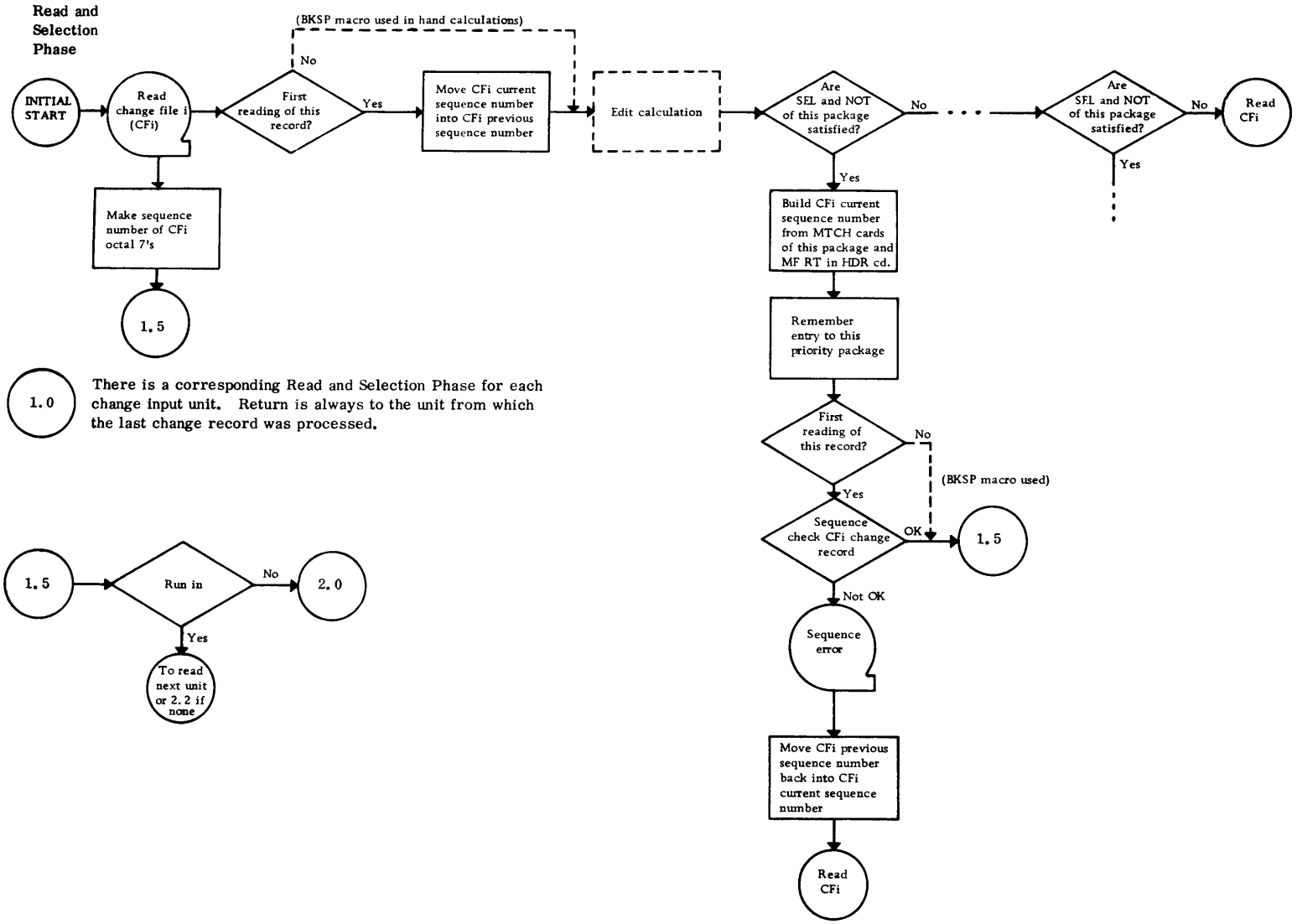
Both change records might have applied to either priority packet 1 or 3. However, a single master file record could not be changed first in packet 3 and then in packet 1.

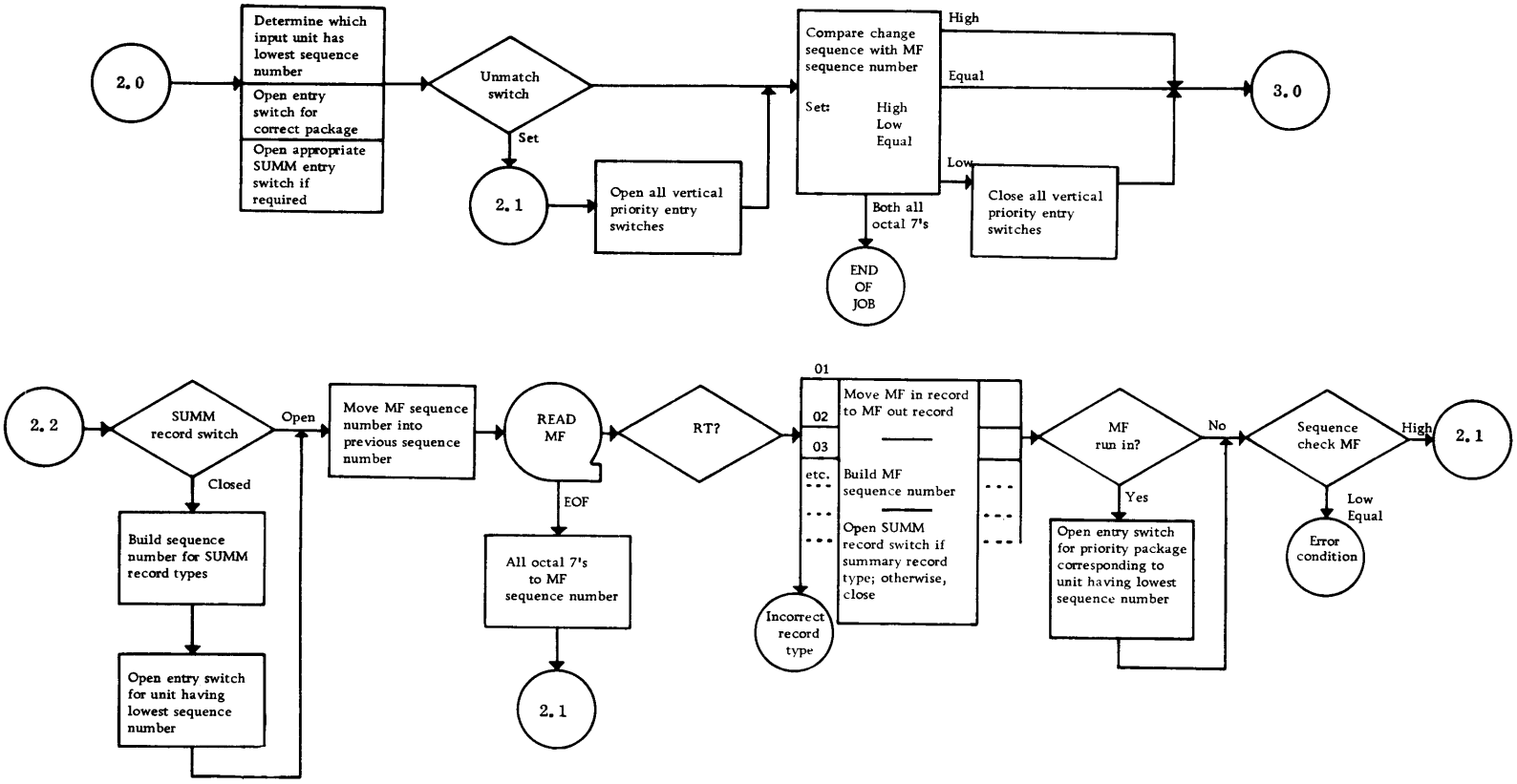
If only one change record matched the master file record, it would be changed only in that priority packet and in vertical packet 2. Similarly, if the selection criteria for packet 2 were not met, only packets 1 and 3 would cause change.

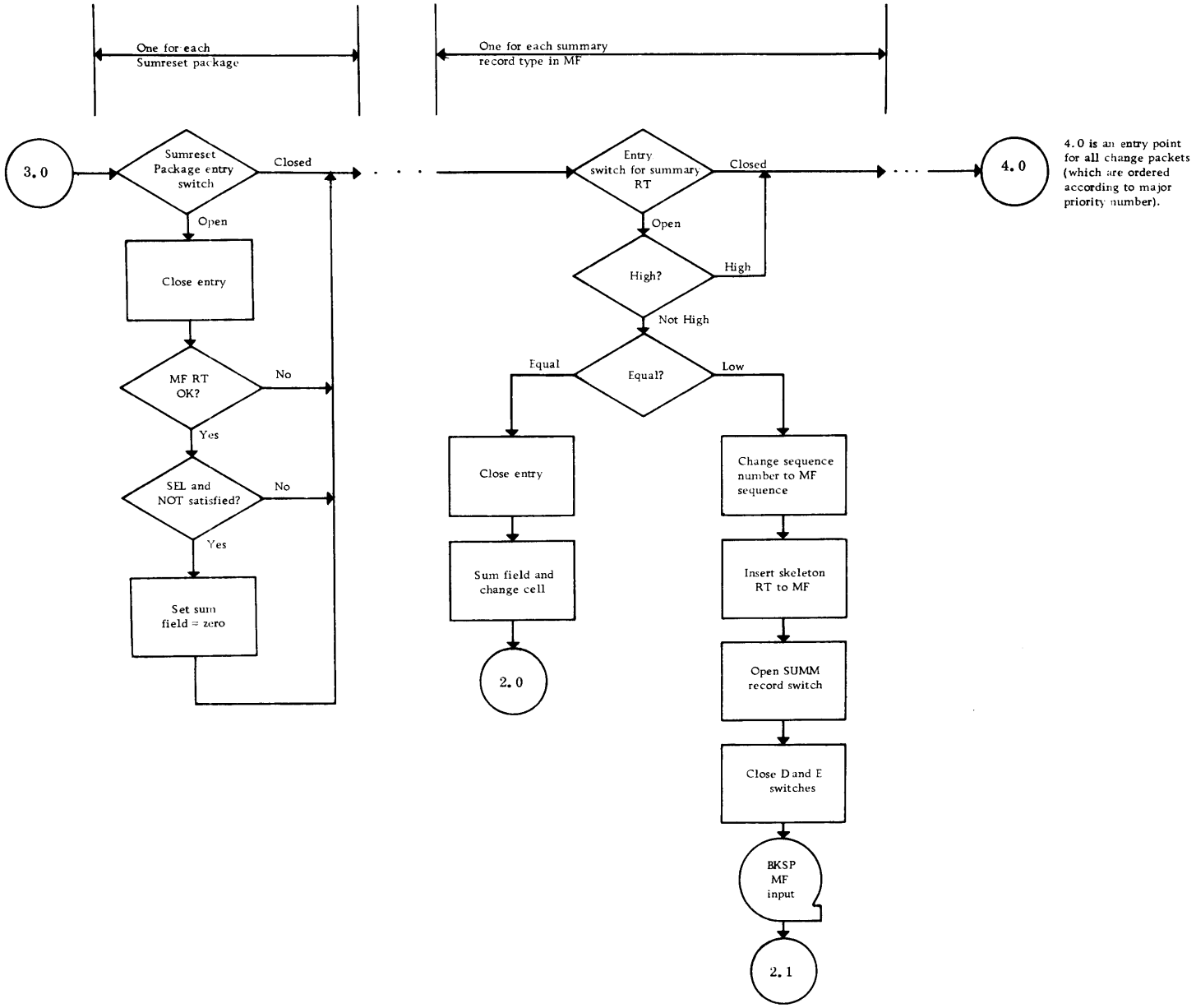
Change records not matching a master file record either cause inserts or are bypassed.

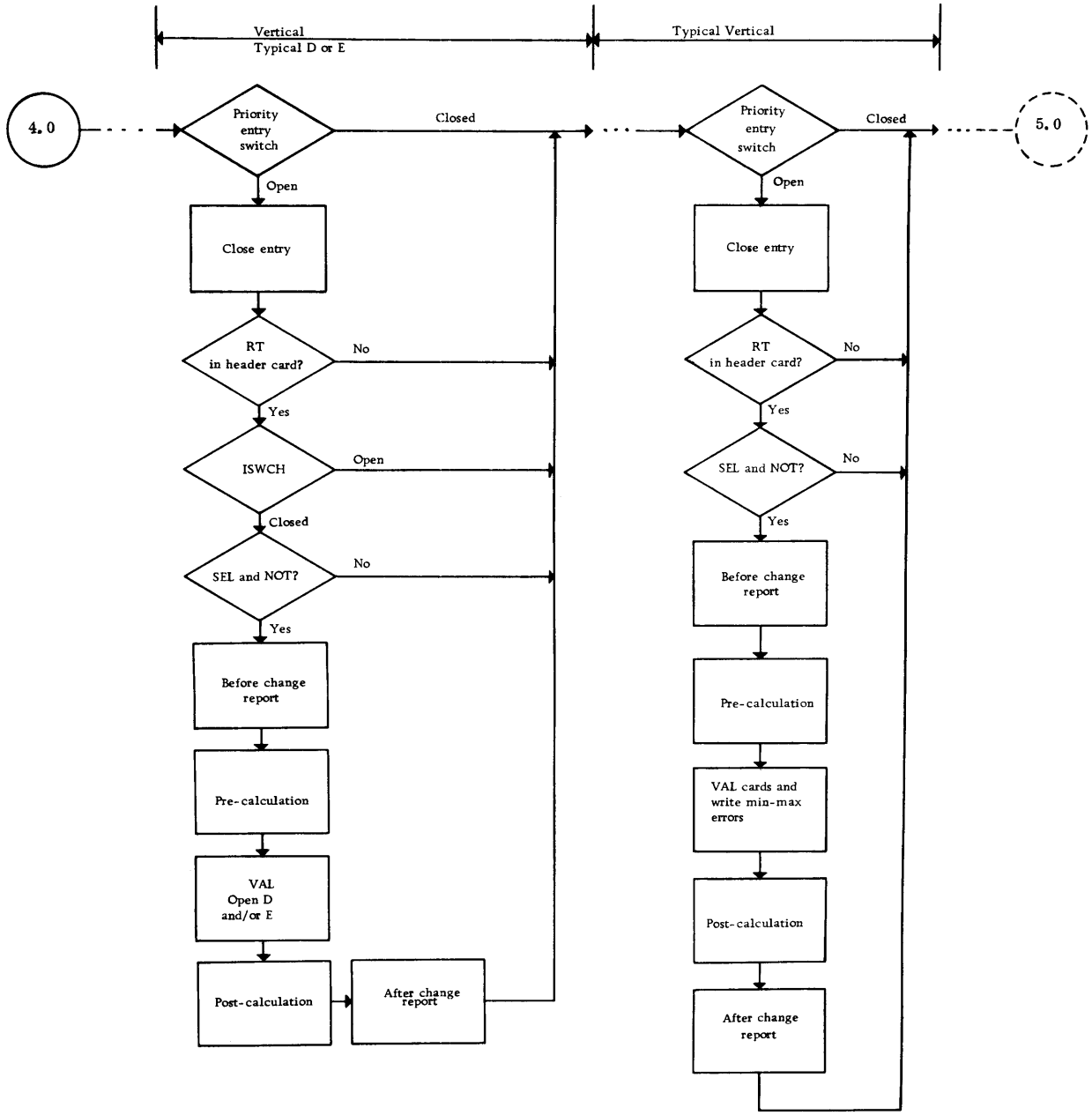
Master records not matching a change record are written directly to the new master file after passing through vertical change packets.

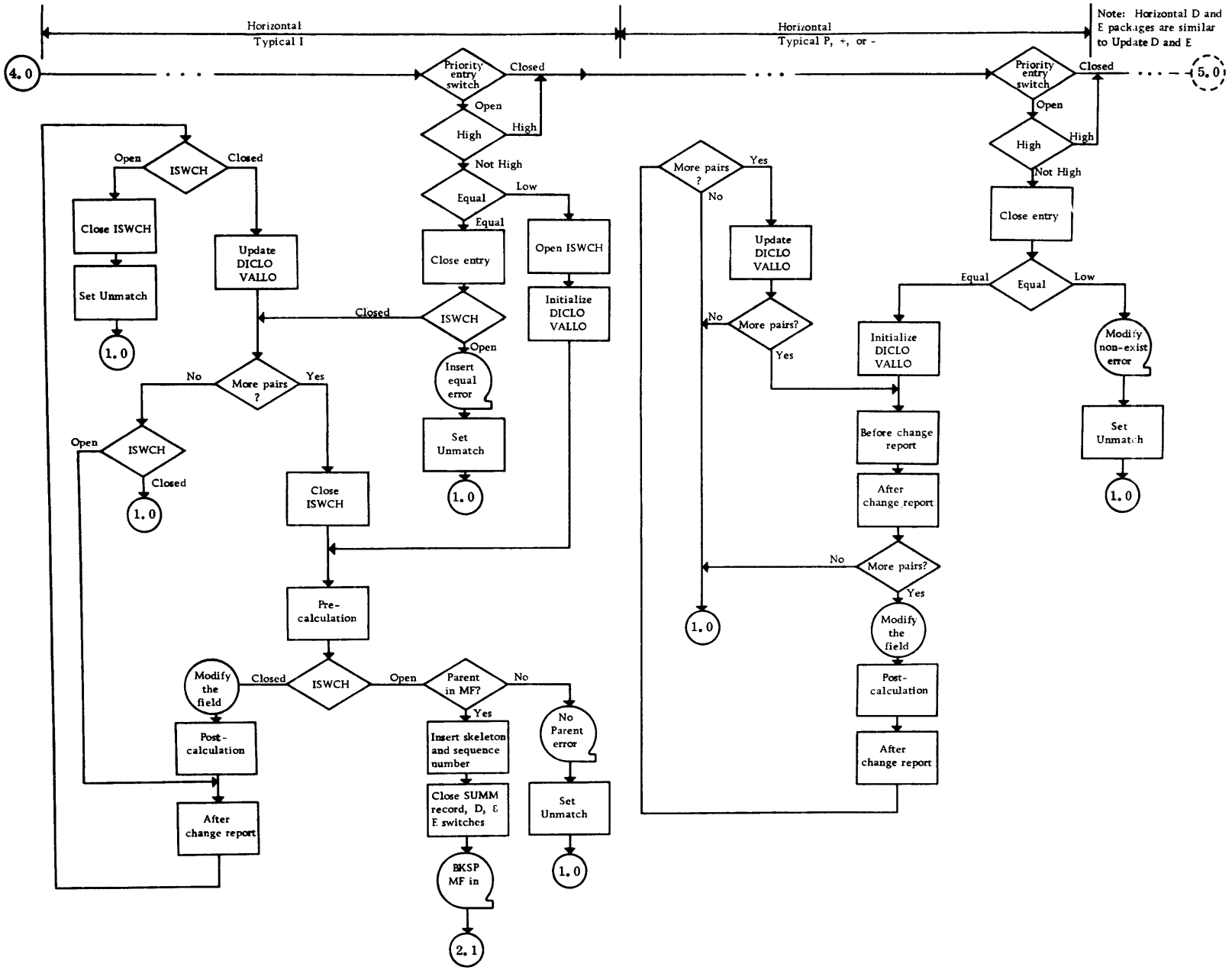
The following flow charts are representative of File Processor object program flow charts.

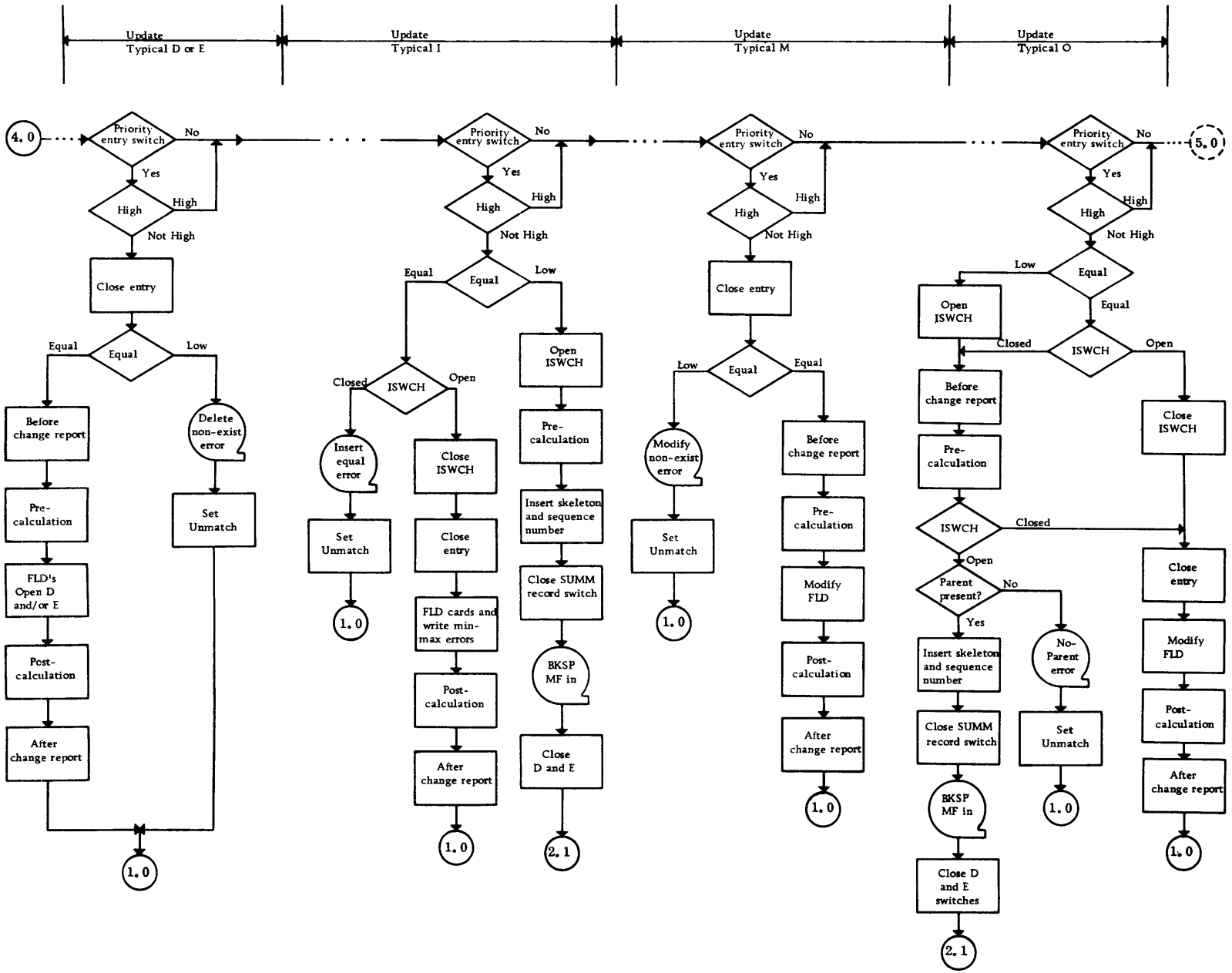

Read and Selection Phase
(BKSP macro used in hand calculations)
No
INITIAL START
Read change file i (CFi)
First reading of this record?
Yes
Move CFi current sequence number into CFi previous sequence number
Edit calculation
Are SEL and NOT of this package satisfied?
No
Are SEL and NOT of this package satisfied?
No
Read CFi
Yes
Make sequence number of CFi octal 7's
1.5
Build CFi current sequence number from MTCH cards of this package and MF RT in HDR cd.
Remember entry to this priority package
First reading of this record?
No
(BKSP macro used)
Yes
Sequence check CFi change record
OK
1.5
Not OK
Sequence error
Move CFi previous sequence number back into CFi current sequence number
Read CFi
1.0
There is a corresponding Read and Selection Phase for each change input unit. Return is always to the unit from which the last change record was processed.
1.5
Run in
No
2.0
Yes
To read next unit or 2.2 if none

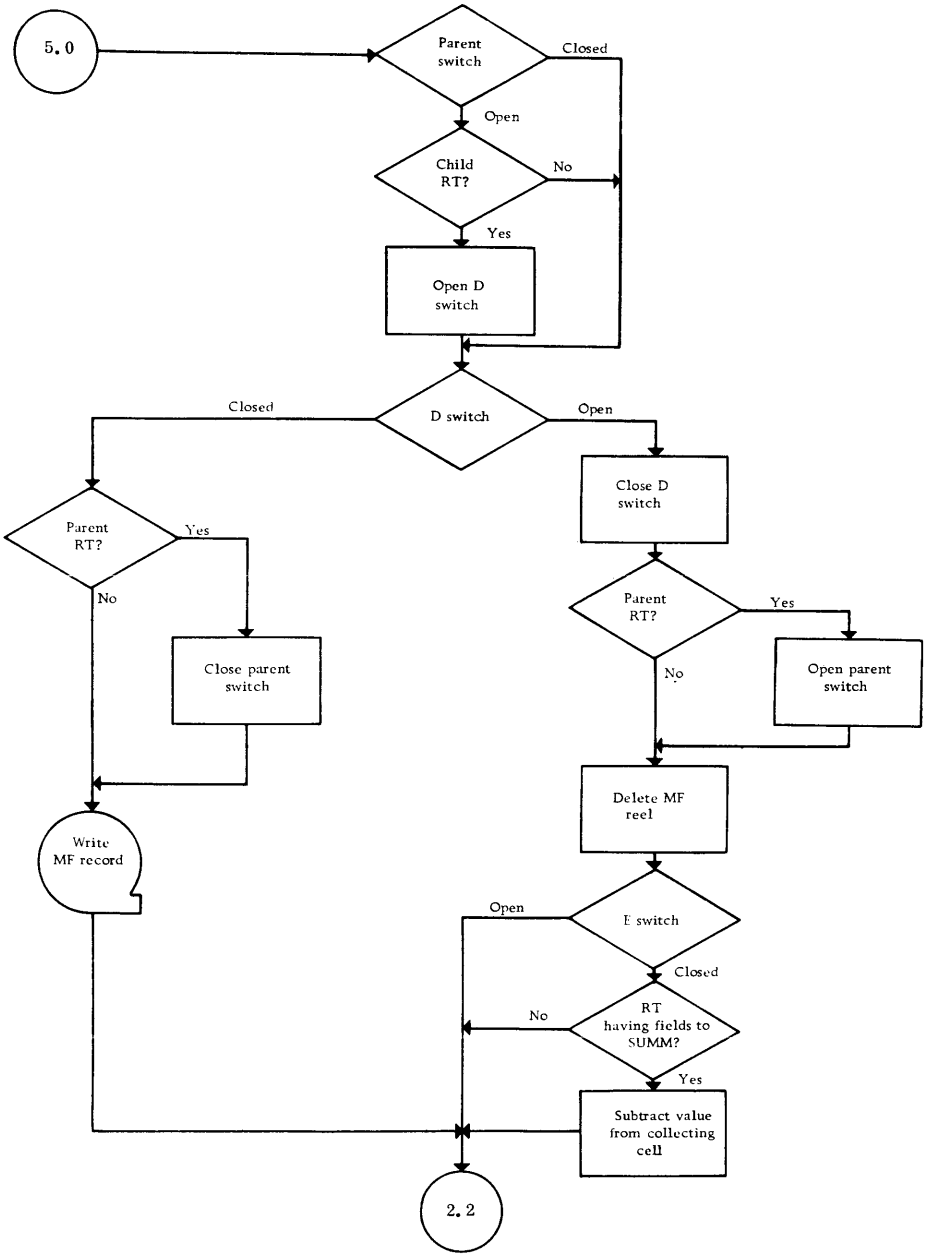

2. 0
Determine which input unit has lowest sequence number
Open entry switch for correct package
Open appropriate SUMM entry switch if required
Unmatch switch
Set
2. 1
Open all vertical priority entry switches
Compare change sequence with MF sequence number
Set: High Low Equal
High
Equal
Low
Both all octal 7's
END OF JOB
Close all vertical priority entry switches
3. 0
2. 2
SUMM record switch
Closed
Open
Build sequence number for SUMM record types
Open entry switch for unit having lowest sequence number
Move MF sequence number into previous sequence number
READ MF
EOF
All octal 7's to MF sequence number
2. 1
RT?
01
02
03
etc.
Move MF in record to MF out record
Build MF sequence number
Open SUMM record switch if summary record type; otherwise, close
Incorrect record type
MF run in?
No
Yes
Open entry switch for priority package corresponding to unit having lowest sequence number
Sequence check MF
High
Low
Equal
Error condition
2. 1


One for each
Sumreset package
One for each summary
record type in MF
3.0
Sumreset
Package entry
switch
Closed
Open
Close entry
MF RT
OK?
No
Yes
SEL and
NOT satisfied?
No
Yes
Set sum
field = zero
Entry
switch for summary
RT
Closed
Open
High?
High
Not High
Equal
Equal?
Low
Close entry
Sum field and
change cell
2.0
Change sequence
number to MF
sequence
Insert skeleton
RT to MF
Open SUMM
record switch
Close D and E
switches
BKSP
MF
input
2.1
4.0
4.0 is an entry point
for all change packets
(which are ordered
according to major
priority number).


Vertical
Typical D or E
Typical Vertical
4.0
Priority entry switch
Closed
Open
Close entry
RT in header card?
No
Yes
ISWCH
Open
Closed
SEL and NOT?
No
Yes
Before change report
Pre-calculation
VAL Open D and/or E
Post-calculation
After change report
Priority entry switch
Closed
Open
Close entry
RT in header card?
No
Yes
SEL and NOT?
No
Yes
Before change report
Pre-calculation
VAL cards and write min-max errors
Post-calculation
After change report
5.0


Horizontal
Typical I
Horizontal
Typical P, +, or -
Note: Horizontal D and E packages are similar to Update D and E
4.0
5.0
Priority entry switch
Closed
Open
High
Not High
Equal
Low
ISWCH
Close ISWCH
Set Unmatch
1.0
Update DICLO VALLO
More pairs ?
Yes
No
Close ISWCH
Pre-calculation
Modify the field
Post-calculation
After change report
Parent in MF?
Insert skeleton and sequence number
Close SUMM record, D, & E switches
BKSP MF in
2.1
No Parent error
Close entry
Open ISWCH
Initialize DICLO VALLO
Insert equal error
Before change report
Modify non-exist error


Update
Typical D or E
Update
Typical I
Update
Typical M
Update
Typical O
4. 0
Priority entry switch
No
Yes
High
High
Not High
Close entry
Equal
Low
Before change report
Pre-calculation
FLD's Open D and/or E
Post-calculation
After change report
Delete non-exist error
Set Unmatch
1. 0
Closed
ISWCH
Open
Insert equal error
Close ISWCH
Close entry
FLD cards and write min-max errors
Post-calculation
After change report
Open ISWCH
Pre-calculation
Insert skeleton and sequence number
Close SUMM record switch
BKSP MF in
Close D and E
2. 1
Modify non-exist error
Before change report
Pre-calculation
Modify FLD
Post-calculation
After change report
Open ISWCH
Before change report
Pre-calculation
Parent present?
No
Yes
Insert skeleton and sequence number
Close SUMM record switch
BKSP MF in
Close D and E switches
No-Parent error
Close ISWCH
Close entry
Modify FLD
Post-calculation
After change report
5. 0


5.0
Parent switch
Closed
Open
Child RT?
No
Yes
Open D switch
D switch
Closed
Open
Parent RT?
Yes
No
Close parent switch
Write MF record
Close D switch
Parent RT?
Yes
No
Open parent switch
Delete MF reel
E switch
Open
Closed
RT having fields to SUMM?
No
Yes
Subtract value from collecting cell
2.2

APPENDIX B: FORMAT OF PREPROCESSED HORIZONTAL DATA

Following is the format of each record of the preprocessed horizontal data file.

Word	Description
1 to 1+n	Selection and matching data, i.e., a copy of the input data up to the first pair or triplet. When in core storage, word 1 is located at MRGLO.
1+n+1	Word count and folded check sum.
1+n+2	Master record type and field number in BCD. When in core storage, word 1+n+2 is located at DICLO.
1+n+3	Master record field mode, in BCD.
1+n+4	Address contains the master record field length, in binary.
1+n+5	Address contains the increment of the master record field, in words.
1+n+6	Address contains the increment of the master record field, in bits.
1+n+7	Minimum test value, if any.
1+n+8	Maximum test value, if any.
1+n+9	Not used.
1+n+10 to 1+n+12	Master file field name, in BCD.
1+n+13	Last 6 bits contain the action code, in BCD.
1+n+14 to 1+n+14+j	Horizontal change value (size determined by dictionary). When in core storage, word 1+n+14 is located at VALLO.

Words 1+n+2 through 1+n+14+j are repeated for each pair or triplet. The last word of each record contains all zeros.

APPENDIX C: MACRO-INSTRUCTION EXPANSIONS

This appendix gives the expansions of all of the File Processor macro-instructions. In some instances, these expansions vary depending upon the data being used.

The following symbols will be used:

x	Record type; this must be two digits
y	Field number; this must be four digits
bits	Partial word increment, in bits
word	Increment, in words
size	Length, in bits
irt	Indirect address of location which contains record type
min	Minimum test value
max	Maximum test value
error	A word whose address contains the first location of the error routine for this packet and whose decrement contains the number of words in that error routine
$errmsg_1$	Word 1 of error message
$errmsg_2$	Word 2 of error message
errfld	Location of the second field of the error record
errloc	Location of error routine for this packet
inst	Location of the first instruction to read the next pair or triplet
flno	File number
next	Location of next priority packet
switch	Location of entry switch for current packet

The GET Macro-Instruction

1. Either a, b, or c, depending upon whether the field being obtained is:

   a. Unpacked BCD

   ```
   TSX     GETUN,4
   ```

   b. Packed BCD

   ```
   TSX     GETPA,4
   ```

c. Binary

```
TSX     GETBI,4
```

2. Either a or b, depending upon whether the record being obtained is:

a. Lower than the current record type

```
PZE     irt
```

b. The current record type

```
PZE     SFLOC
```

3. Either a or b, depending upon whether the combination used above was:

a. 1a and 2a, 1b and 2a, 1a and 2b, or 1b and 2b

```
PZE     size
PZE     word,2
PZE     bits
```

b. 1c and 2a or 1c and 2b

```
PZE     word,2
```

The PUT Macro-Instruction

1. Either a, b, or c, depending upon whether the field being stored is:

a. Unpacked BCD

```
TSX     PUTUN,4
```

b. Packed BCD

```
TSX     PUTPA,4
```

c. Binary

```
TSX     PUTBI,4
```

2. Either a or b, depending upon whether the element used above was:

a. 1a or 1b

```
PZE     size
PZE     word,2
PZE     bits
```

b. 1c

```
BCI     1,xy
PZE     word,2
PZE     min
PZE     max
PZE     error
```

The SUM Macro-Instruction

```
TSX     SUMRY,4
BCI     1,xy
PZE     error
```

The REJECT Macro-Instruction

1.
```
TSX     errloc,4
BCI     1,errmsg1
BCI     1,errmsg2
LAC     MRGLO,1
LAC     SFLOC,2
```

2. Either a, b, or c, depending upon the packet action and the type of packet in which the macro-instruction is located:

   a. Update change with O action
   ```
   STZ     switch
   NZT     EQUAL
   TRA     MERGE
   STZ     ISWCH
   TRA     MRGUN
   ```

   b. All packets with I action
   ```
   STZ     switch
   STZ     ISWCH
   TRA     MRGUN
   ```

   c. All cases not covered by 2a or 2b
   ```
   TRA     MERGE
   ```

The IGNORE Macro-Instruction

1.
```
LAC     MRGLO,1
LAC     SFLOC,2
```

2. Either a, b, c, or d, depending upon the packet action and the type of packet in which the macro-instruction is located:

   a. Vertical packet
   ```
   TRA     next
   ```

   b. Update packet with O action
   ```
   STZ     switch
   NZT     EQUAL
   TRA     MERGE
   STZ     ISWCH
   TRA     MRGUN
   ```

   c. All packets with I action
   ```
   STZ     switch
   STZ     ISWCH
   TRA     MRGUN
   ```

   d. All cases not covered by 2a, 2b, or 2c
   ```
   TRA     MERGE
   ```

The ERRFLD Macro-Instruction

1.
```
CAL*    DICLO
ANA     =O000077777777
SLW     errfld
TSX     errloc,4
BCI     1,errmsg1
BCI     1,errmsg2
LAC     MRGLO,1
LAC     SFLOC,2
```

2. Either a or b, depending upon whether the macro-instruction is in a:

   a. Pre-calculation
   ```
   TRA     STEPP
   ```

   b. Post-calculation
   ```
   TRA     inst
   ```

The BKSP Macro-Instruction

```
SXA     *+3,1
AXT     flno,1
STL     BKSPT,1
AXT     **,1
```

APPENDIX D: HAND CALCULATION OPERATION CODES

The following operation codes may be used in either File Processor or Reports Generator hand calculations. Those operation codes preceded by an asterisk (*) are IOCS commands; those preceded by a double asterisk (**) are SCAT language prefix codes; those preceded by a triple asterisk (***) are SCAT language pseudo-operations; all others are 7090 machine instructions.

ACL	FDP	LRS	**PZE	SXA
ADD	FMP	LTM	RFT	SXD
ADM	**FOR	LXA	RIA	*TCH
ALS	FRN	LXD	RIL	TIF
ANA	FSB	**MON	RIR	TIO
ANS	FSM	MPR	RIS	TIX
ARS	**FVE	MPY	RND	TLQ
AXC	HPR	MSE	RNT	TMI
AXT	HTR	**MTH	RQL	TNO
***BES	IIA	**MTW	SBM	TNX
***BCI	IIL	**MZE	SIL	TNZ
***BSS	IIR	NOP	SIR	TOV
CAL	IIS	NZT	**SIX	TPL
CAQ	*IOCD	OAI	SLF	TQP
CAS	*IOCP	***OCT	SLN	TRA
CHS	*IOCT	OFT	SLQ	TSX
CLA	*IORP	ONT	SLT	TTR
CLM	*IORT	ORA	SLW	TXH
CLS	*IOSP	ORS	SSM	TXI
COM	*IOST	OSI	SSP	TXL
CRQ	IOT	PAC	STA	TZE
CVR	LAC	PAI	STD	UAM
DCT	LAS	PAX	STI	UFA
***DEC	LBT	PBT	STL	UFM
DVH	LDC	PDC	STO	UFS
DVP	LDI	PDX	STP	USM
ENK	LDQ	PIA	STQ	VDH
ERA	LFT	**PON	STR	VDP
ETM	LGL	PSE	STT	VLM
FAD	LGR	**PTH	STZ	XCA
FAM	LLS	**PTW	SUB	XCL
FDH	LNT	PXA	**SVN	XEC
		PXD	SWT	ZET

INDEX

IBM
International Business Machines Corporation
Data Processing Division, 112 East Post Road, White Plains, N. Y.

Printed in U. S. A. J28-6167