GA21-9184-3

Systems

IBM 3742 Dual Data Station Reference Manual

IBM

Fourth Edition (November 1977)

This is a major revision of, and obsoletes, GA21-9184-2. Changes are indicated by a vertical line at the left of the change; new or extensively revised illustrations are denoted by a bullet (●) at the left of the figure caption.

Changes are periodically made to the information in this manual and will be reported in subsequent revisions or technical newsletters.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A Reader's Comment Form is at the back of this publication. If the form is gone, address your comments to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901.

Preface

This publication is a reference manual that provides specific information about the IBM 3742 Dual Data Station. This publication is intended for installation supervisors, application personnel, system analysts, and system engineers who wish specific information on the programming for and operational considerations of the IBM 3742 Dual Data Station.

The IBM 3742 Dual Data Station is part of the IBM 3740 Data Entry System. The IBM 3742 Dual Data Station is a two operator key entry station used to record data on the data recording media called the IBM Diskette. Once data has been recorded on the diskette by the 3742, the data can be converted to tape by using the IBM 3747 Data Converter, or the diskette can be used with the IBM 3540 Diskette Input/Output Unit which provides direct input capabilities to a System/370 that has virtual storage capabilities.

This publication is divided into the following:

- *Chapter 1. IBM 3742 Dual Data Station* provides an overview of the 3742.
- *Chapter 2. Operation Modes and Considerations* provides information about buffers, modes of operation, and related considerations concerning the 3742.
- *Chapter 3. Program Information* provides information on program coding and program chaining.
- *Chapter 4. Production Statistics* provides information on the production statistic counters, and the displaying of the counters.
- *Chapter 5. Offline Field Totals* provides information on offline field totals capabilities.
- *Chapter 6. Second Disk* provides information on second disk search and disk copy operations.
- *Chapter 7. Disk Initialization* provides information on how to initialize or check a diskette.
- *Chapter 8. Self-Check Feature* provides information about self-checking when using modulus 10 or modulus 11.
- *Chapter 9. Proof Keyboard Feature* provides information about the numeric key arrangement for the proof keyboard.
- *Appendix A. Timings* provides information on timing for searches, offline field totals, copies, and disk initialization operations.
- Appendix B provides a glossary of terms used in this publication.
- An index is provided at the end of the publication.

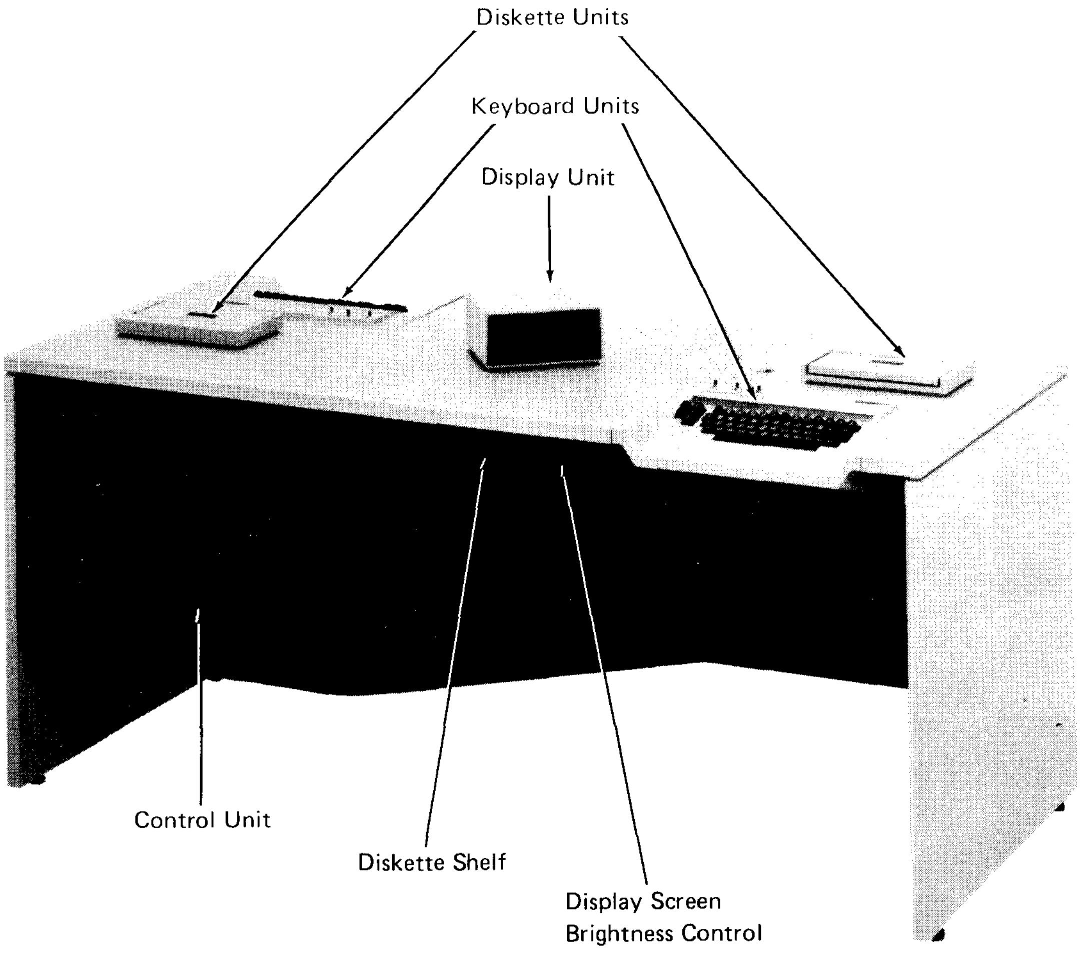
This publication assumes that the reader is familiar with data entry techniques and is also familiar with the contents of the following manuals:

- *IBM 3740 Data Entry System System Summary and Installation Manual - Physical Planning,* GA21-9152.
- *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136.
- *The IBM Diskette General Information Manual,* GA21-9182.
- *IBM 3741 Data Station Reference Manual,* GA21-9183.

Contents

IBM 3742 DUAL DATA STATION OVERVIEW

The IBM 3742 Dual Data Station (Figure 1) is a stand-alone device with two operator stations. Each operator station has its own keyboard and diskette unit. The display unit is shared between stations.



• **Figure 1. IBM 3742 Dual Data Station**

Keyboard

The keyboard (Figure 2) consists of data keys, special function keys, function select keys, two rows of labels, and switches. Data keys provide the means of keying 64 different displayable characters. All 256 EBCDIC codes can be keyed with the use of the Hex key. The function select keys are used to redefine the top row of keys as indicated by the two rows of labels above the top row of keys. The lefthand function select key, lettered in green, is associated with the upper row of labels (that is, redefines the top row of keys according to the top row of labels), that are also green. The righthand function select key lettered in white is associated with the lower row of labels. The special keys (DUP, FIELD BKSP, REC ADV, RESET, etc.) perform specific functions during data entry. The switches provide automatic skip/duplicate, automatic record advance, and program numeric shift functions. For a complete description of the keyboard's keys and switches, see the *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136.

Note: If the 3742 has the 128 feature, the scroll forward (SCRL FWD) key and the scroll backward (SCRL BKWD) key are provided on the keyboard. These keys' functions are also described in the *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136.

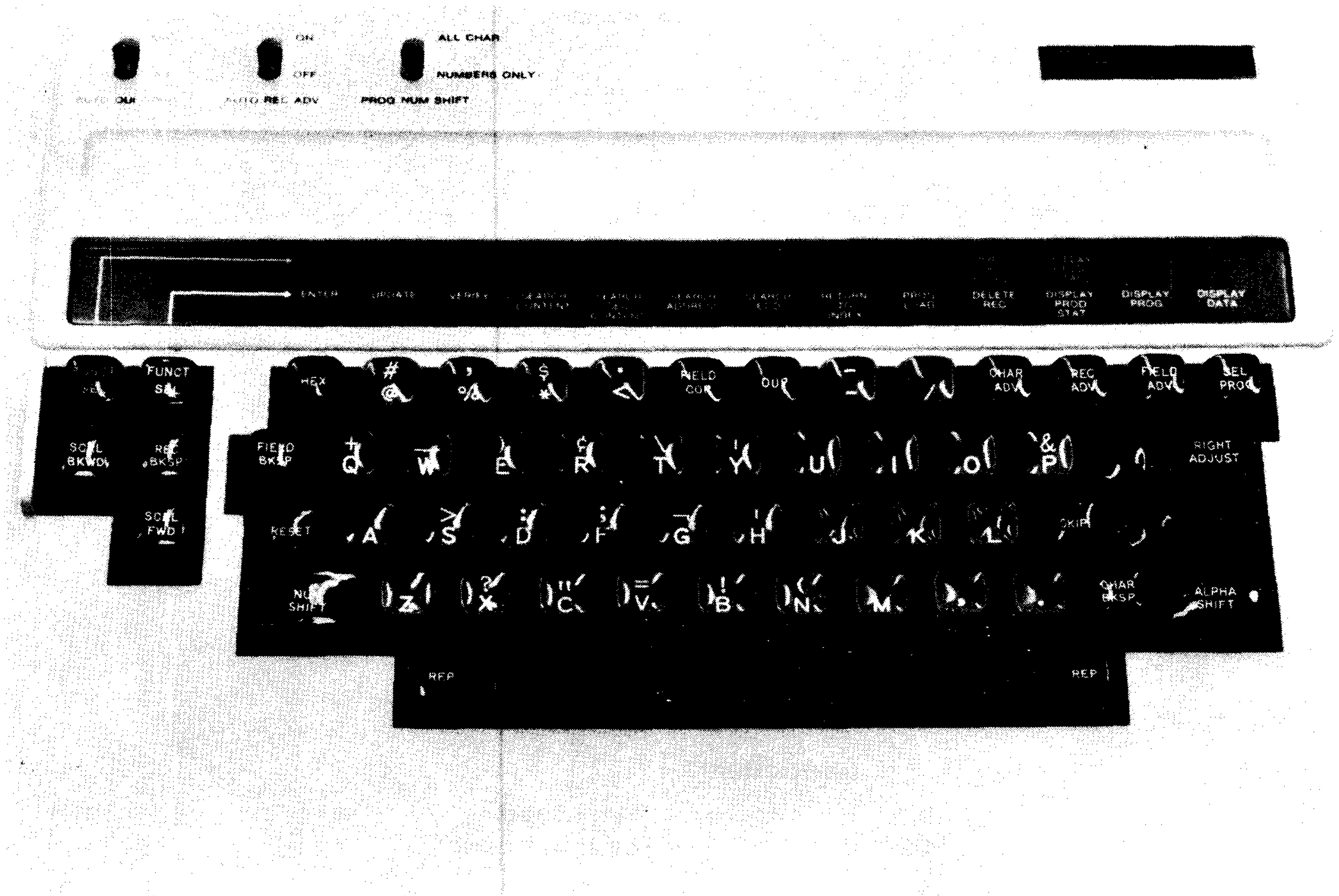


Figure 2. 3742 Keyboard with 128 Feature

Display

The display (Figure 3) provides station status information, program information, and data on the screen. Each operator station has three 40-character lines of display: line 1 is the station status and lines 2 and 3 are for data or program information. A cursor (underscore) is used to indicate the position of the next character to be entered. If the 3742 has a record length of 80 characters, the cursor position is indicated in positions 2 and 3 of the status line; position 1 is blank. If the 3742 has a record length of 128 characters, the cursor position is indicated in positions 1, 2, and 3 of the status line.

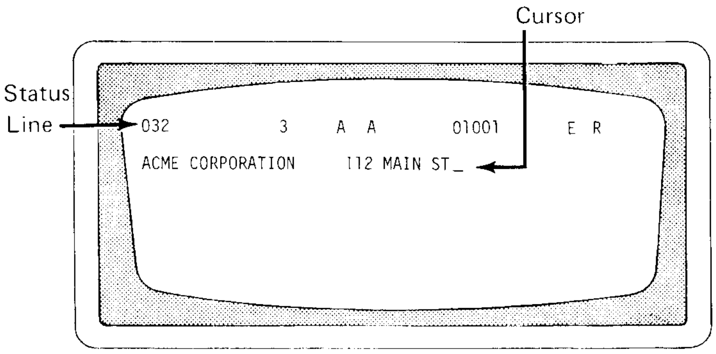


Figure 3. Display Unit

If the 3742 has the 128 feature, the scroll keys can be used to view 128 characters. When a record is displayed, only the first 80 characters are displayed. When the cursor moves to position 81, or the scroll forward key is used, character positions 41 to 120 are displayed.

Similarly, when the cursor moves to position 121, or the scroll forward key is used, character positions 81 to 128 are displayed on the two data lines (121 to 128 are displayed on the second data line). Moving the cursor beyond 128 (to 000) does not change the display. Moving the cursor backward from 121 to 120, or using the scroll backward key, causes character positions 41 to 120 to be displayed. Similarly, when the cursor moves backward from 81 to 80, or the scroll backward key is used, character positions 1 to 80 are displayed.

A complete description of the display is provided in the *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136.

Diskette Unit

The disk unit reads and writes on one side of the diskette. Figure 4 shows the IBM diskette. For a complete description of the IBM diskette, its layout, care, storage, and handling, see *The IBM Diskette General Information Manual,* GA21-9182.

Figure 4. IBM Diskette

STANDARD FUNCTIONS AND AVAILABLE FEATURES

Figure 5 shows the standard functions of the 3742 and the available features for the 3742.

Standard Functions	Available Features
● Entering and updating records	● Feature group A which provides: – Search on content – Self-checking number (modulus 10 and 11) – Offline field totals – Disk copy – Disk initialization
● Search on record address	
● Search on end of data	
● Program chaining	
● Six levels of program control for each operator station	● 128 feature which provides: – Variable record length from 1 to 128 – Four additional program buffers for each operator station
● Record length of 80 characters	
● Verify	● Record insert
● Production statistics	● Proof keyboard

Figure 5. Standard Functions and Available Features

Chapter 2. Operation Modes and Considerations

Entering, updating, verifying, or searching of data on a disk is performed by the data stations using different modes of operation. The sequence of operations performed by the control unit within the data station is directly dependent upon the mode in which the station is functioning. Each mode of operation is unique, and, therefore, the use of buffers, keys, and information displayed is different in each mode.

BUFFERS

Buffers are temporary storage areas used for transferring data between the keyboard, display, and disk. The buffers are the current record buffer, previous record buffer, hold buffer, and program buffers.

Current Record Buffer

The current record buffer receives data characters as they are keyed. Records read from disk are also placed in this buffer. Data displayed on the display screen is from the current record buffer.

Previous Record Buffer

The previous record buffer contains the record that was previously displayed in the current record buffer. Data is written on the disk from this buffer. The previous record buffer is the buffer from which characters are obtained by the control unit for duplicating into duplicate fields in the current record buffer.

Hold Buffer

The hold buffer contains the next record to be verified when in verify mode. This buffer contains the search mask during search on content or search on sequential content operations.

Program Buffers

The 3742 has six program buffers numbered 1 through 6. The 3742, with the 128 feature, has 10 program buffers numbered 1 through 9 and A. The program buffers usually contain programs that control record format and processing. For offline field totals operations, the program buffers will also contain some field totals control statements that are required in order to perform the field totals operations. See *Chapter 5. Offline Field Totals* for more information.

POWER ON INFORMATION

When the 3742 is turned on, the current record, previous record, and hold buffers are filled with blanks, and the program buffers are filled with N's. Program buffer 0 is selected. The status line displays the machine-not-ready disk status (N), the read index mode (X), program level 0, and current disk address 00000. All other positions on the status line are blank.

Closing the cover on the diskette that has been inserted changes the machine status to wait (W). The volume label is read but not displayed, and the accessibility field in the volume label is checked. The accessibility field is used to allow (field is blank) or inhibit (field contains any character) use of a diskette. If the accessibility field is not blank, the disk cannot be processed, and a disk accessibility (A) error is displayed. Then, the 3742 waits for the diskette to be removed.

A write test is made; if not successful, ?? is displayed in positions 9 and 10 of the status line. If ?? is displayed, other diskettes should be inserted to see if the ?? disappears. If the ?? continues to be displayed, call IBM for service since the machine cannot write records on the disk.

If all tests are successful, sector 08 is read and displayed. Sector 08 contains the first data set label (a data set label defines required information about a data set. See *Index Track Data Set Labels* for more information). Data entry or other data station operations can now be performed.

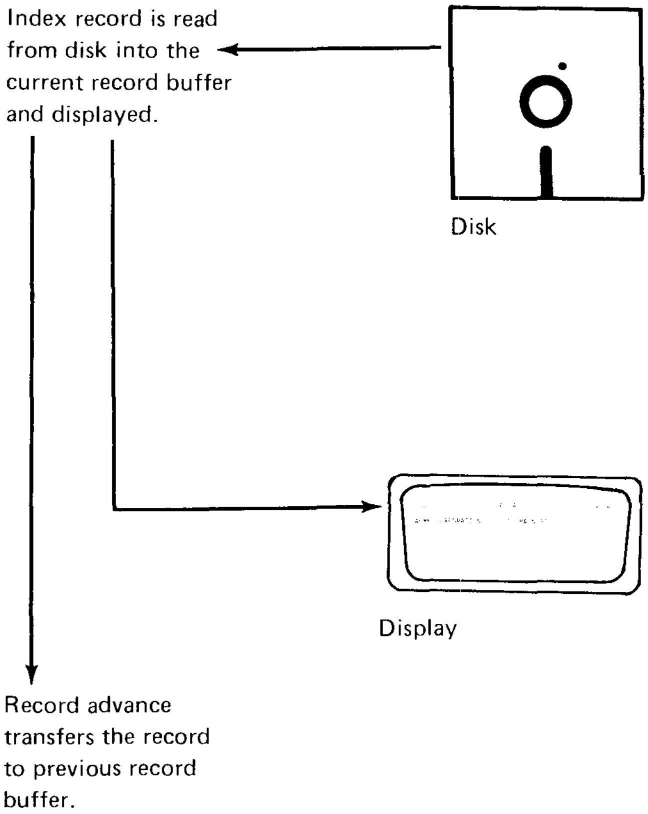
MODES OF OPERATION

The mode of operation is indicated by a code displayed in position 38 of the status line of the display unit. Some of the modes of operation for the 3742 and their corresponding display codes are:

Mode	Display Code
Read index	X
Modify index	M
Enter	E
Update	U
Verify	V
Field correct	C
Search	S
Field totals	F

Read Index Mode (X)

The 3742 is in read index mode whenever it is on the index track. The read index mode allows records on the index track representing the volume label, the data set labels, and other system data to be displayed on the display screen. These records can also be modified or deleted by a special procedure called modify index. See *Modify Index Mode (M)* in this section for further information. Figure 6 shows the data flow for read index mode.



Index record is read from disk into the current record buffer and displayed.

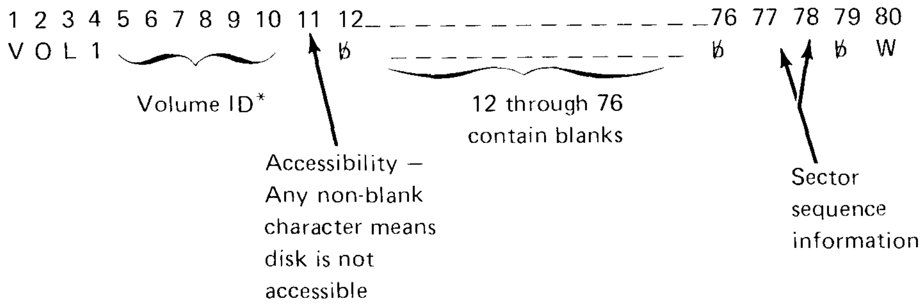
Record advance transfers the record to previous record buffer.

Figure 6. Data Flow for Read Index Mode

Index Track Layout

The first track on a disk is called the index track and is numbered 00. The index track's 26 sectors are for system or data set label information. The 26 sectors on each of the remaining tracks, numbered 01 through 73, are used for data. Each sector can contain one record. Track 74 is reserved and should not be used for data entry.

Each record on the index track has a record length of 80 characters. The index track layout is shown in Figure 7. The first seven sectors of the index track may contain system information or information about the diskette, such as the location of tracks that have developed errors in use. These seven sectors are not usually used or modified by the operator. The remaining sectors (08-26) may contain data set labels, which are used to define the data sets resident on the disk.

Sector	Contents
Sector 1	80 blanks
Sector 2	80 blanks
Sector 3	Position 1: machine test character for a write test
Sector 4	80 blanks
Sector 5	Positions 1-5: ERMAP; positions 6-80: blanks If the disk has one or two bad tracks, positions 7 and 8 indicate the first bad track and position 9 contains a 0; positions 11 and 12 indicate the second bad track and position 13 contains a 0.
Sector 6	80 blanks
Sector 7	Volume label: positions  1 2 3 4 5 6 7 8 9 10 11 12 76 77 78 79 80 V O L 1 b̸ b̸ b̸ W Volume ID* 12 through 76 contain blanks Accessibility – Any non-blank character means disk is not accessible Sector sequence information
Sectors 8-26	Data set labels

*The volume ID field contains IBMIRD when the diskette is receivea trom IBM. When you initialize the diskette, the volume ID field contains either the volume ID entered by the operator or is blank.

Figure 7. Index Track Layout

Index Track Data Set Labels

Sectors 8 through 26 of the index track are used primarily for data set labels. A data set label defines required information about a data set on disk such as beginning of extent (BOE), end of data (EOD), end of extent (EOE), and the record length of records for the data set.

Data Set Label Field Description

Figure 8 shows the field layout of a data set label. The following is a description of each of the fields in the data set label.

Header 1 Field (Positions 1-4): The header 1 field, a required field, is a data set label identifier for system application. It must be HDR1 in order to be a valid data set label identifier. When a disk is initialized, only sector 8 of the index track has a HDR1. Sectors 9-26 contain DDR1 in positions 1 through 4 and are deleted. The deleted label identifier can be modified to a valid data set label identifier (HDR1) by using the modify index mode. Modifying the label in this way allows you to define more data sets on one disk.

Data Set Name Field (Positions 6-13): The data set name field, an optional field, is provided to allow you to give the data set a descriptive name. When a disk is initialized, the data set name field of the data set label in sector 08 contains DATA as a name. The data set name field of the data set labels in sectors 09 through 26 contains DATA 09 through DATA 26, respectively. The data set name can be changed to a descriptive data set name by using the modify index mode.

Record Length Field (Positions 23-27): The record length field, a required field, must contain the record length for the records to be placed in the data set. For the 3742, the record length must always be bbb80. Disk initialization places the record length into this field for you. For the 3742 with the 128 feature, the record length field can be 00001 through 00128. Disk initialization can be used to set the record length to bbb80 or 128. The modify index mode can be used to change the 128 to any value from 1 through 128.

Beginning of Extent Field (Positions 29-33): The beginning of extent (BOE) field, a required field, identifies the address of the first sector of the data set (where the first record of the data set will be). The BOE field has this format: TT0SS where TT is the track number and SS is the sector number. The track number must be in positions 29 and 30, a 0 must always be in position 31, and the sector number must be in positions 32 and 33. The BOE address must be 01001 or greater.

End of Extent Field (Positions 35-39): The end of extent (EOE) field, a required field, identifies the address of the last sector of the data set (where the last record of the data set will be). The EOE field has the same format as the BOE field. The track number (TT) must be in positions 35 and 36, a 0 must always be in position 37, and the sector number (SS) must always be in positions 38 and 39. The EOE address cannot be larger than 74026 or an incorrect data set label (B) error will occur. However, track 74 is reserved and should not be used for data entry.

Bypass Data Set Field (Position 41): The bypass data set field is a field for use by the 3747 or the 3741 with the communications feature, data recorder attachment feature, and I/O adapter. This field must contain a B or be blank. If the field contains a B, the 3747 bypasses the data set (the data in the data set is not converted to tape). If the field is blank, the data in the data set is converted to tape. This field allows you to store programs and data on the same disk. A B would be put into the bypass data set field of the data set label describing the data set containing the programs. The B would prevent the programs from being converted to tape.

For the IBM 3540 Diskette Input/Output Unit, a B in position 41 causes the data set to be bypassed during read operations.

Accessibility Field (Position 42): The accessibility field is used to control access to a data set. If this field is blank, the data set is processed. If any character is in this field, the 3742 does not allow the data set to be processed and a disk accessibility (A) error is displayed. For the 3742, this field cannot be modified.

Position by Position Representation of Data on the Index Track in Any One Sectors 8-26.		
Field Name	**Position**	**Purpose**
Header 1	1-4*	Label identifier for system application; must be HDR1.
	5	Reserved
Data set name	6-13	Descriptive name for data set
	14-22	Reserved
Record length	23-27*	Logical record length
	28	Reserved
Beginning of extent (BOE)	29-33*	Identifies the address of the first sector of the data set.
	34	Reserved
End of extent (EOE)	35-39*	Identifies the address of the last sector reserved for this data set.
	40	Reserved
Bypass data set indicator	41	The IBM 3747 data converter and the IBM 3741 communications feature require that this field contain a B or a blank. If a B is present, the data set is ignored. If a blank is present, the data set is processed. This coding allows the user to store programs and data on the same disk.
Accessibility	42	This field must contain a blank to allow processing.
Write protect	43	If this field contains a P, the data set can be read only. This field must contain a blank to allow both reading and writing.
Interchange type indicator	44	Must be blank. A blank indicates the data set can be used for data interchange.
Multivolume indicator	45**	A blank in this field indicates a data set is contained on one diskette; a C indicates a data set is continued on another diskette; an L indicates the last diskette on which a continued data set resides.
Volume sequence number	46-47	Volume sequence number specifies the sequence of volumes in a multivolume data set. The sequence must be consecutive, beginning with 01 (to a maximum of 99), if used.
Creation date	48-53	May be used to record the date the data set was created. The format of the creation date is YYMMDD, where YY is the year, MM is the month, and DD is the day.
	54-66	Reserved
Expiration date	67-72	May be used to contain the date that the data set expires. The format of the expiration date is the same as for the creation date.
Verify indicator	73	This field must contain a V or a blank. V indicates the data set has been verified.
	74	Reserved
End of data (EOD)	75-79*	Identifies the address of the next available sector.
	80	Reserved

*Indicates an entry required by the 3740 system.
**See description of *Multivolume Indicator Field* in this section.

Figure 8. Data Set Label Layout

Write Protect Field (Position 43): The write protect field controls writing of records on the disk. If this field is blank, records in a data set are read from the disk and records are written on the disk. If this field contains a P, the data set is write protected (data cannot be written on the disk).

Interchange Type Indicator (Position 44): The interchange type indicator field must be blank. A blank indicates the data set can be used for data interchange.

Multivolume Indicator Field (Position 45): The multivolume indicator field is used to indicate whether a data set is on one or more diskettes. A multivolume data set is a data set that is continued from one diskette to another. A blank in this field indicates that the data set is contained on only one diskette. C is placed in this field if the data set is continued on another diskette. When a data set is continued on several diskettes, this field in the data set label of the last diskette must contain an L. The L means that this diskette is the last for a multivolume data set.

For 3747, the L means that the data set ends with the data associated with this label. A ƀ or C means that the data set is not complete with this label.

Volume Sequence Number Field (Positions 46-47): The volume sequence number field, an optional field, is used to specify the sequence of volumes in a multivolume data set (data set is on more than one diskette). The sequence must be consecutive, beginning with 01 (to a maximum of 99), if used. If a diskette will be used on the 3540, this field is required.

Creation Date Field (Positions 48-53): The creation date field, an optional field, is used to record the date the data set was created. The format of the creation date can be YYMMDD, where YY is the year, MM is the month, and DD is the day. An example is: 730704. This is July 4, 1973. If a diskette will be used on the 3540, this field is required.

Expiration Date Field (Positions 67-72): The expiration date field, an optional field, is used to record the date the data set expires. The format of the expiration date is the same as for the creation date. If a diskette will be used on the 3540, this field is required.

Verify Indicator Field (Position 73): The verify indicator field is used by the 3742 to indicate that a data set has been verified by placing a V into this field when the last record of a data set has been verified. This field is blank until the data set has been verified. If a data set was verified and then a change made to the data set (either in enter or update mode), the 3742 replaces the V with a blank.

End of Data Field (Positions 75-79): The end of data (EOD) field, a required field, contains the address of the sector following the last data record in the data set (the EOD address). The address of the last data record in a data set is EOD minus one. When creating a data set label, the EOD address must be equal to the BOE address. The EOD address is automatically updated by one each time a record advance occurs in enter mode. In this manner, the 3742 keeps track of the location on disk where the next record can be written. When the operator completes entering records and performs the return to index function, the EOD value in the data set label is modified to the updated EOD value.

Data Set Label Error Checking

The 3742 checks to see if any of the following error conditions exist in the data set label:

1. The beginning of extent address is less than track 01, sector 01.
2. The end of extent address is greater than track 74, sector 26.
3. The end of extent address is less than the beginning of extent address.
4. The end of data address is greater than the end of extent address plus 1.
5. The end of data address is greater than track 75, sector 01.
6. The sector address is 00 or greater than 26.
7. The record length is not 80 for the 3742, or the record length is 0 or greater than 128 for the 3742 with the 128 feature.
8. The third character of the record address of beginning of extent, end of extent or end of data is other than 0.
9. The last record read and currently displayed is from track 00, sectors 1 through 7.
10. The fourth position of the header 1 field is a number other than 1.
11. The record has been deleted.

When a header 1 record in sectors 08 through 26 is read, the 3742 checks the first eight conditions in the previous list. If any of these conditions exist, an incorrect data set label (B) error is displayed.

When update, verify, enter, or search mode is attempted from the index track, all 11 conditions are checked. If any one of the 11 conditions is present, the incorrect data set label (B) error is displayed.

Multiple Data Sets Per Disk

For ease of processing, one data set per disk is the best. However, more than one data set can be on one disk. Each data set is identified by its respective data set label on the index track. Each data set label identifies a data set as to its beginning and ending addresses, whether the data set has been verified, the record length, the name of the data set (optional field) and other information about the data set. When a diskette is inserted into the disk drive, the data set label at sector 08 on the index track is always displayed first. If there are other defined data set labels on the index track, the operator can record advance until the appropriate data set label is found. To aid the operator in reviewing and finding the desired data set label, the data set name field (position 6-13) in the data set label should be used. In addition, the data set label creation date field and the data set expiration date field should also be used when a clear distinction between jobs is needed to ensure the selection of the correct data set. Then, in addition to reading the data set name, the operator could also check the creation and expiration dates to ensure processing of the right data set, or the right version of the data set.

Modify Index Mode (M)

The modify index mode is used to modify records on the index track. Fields in a data set label, new data set labels, the volume label, or other records on the index track can be modified by using the modify index mode. Any of the records can also be deleted with the use of the modify index mode function.

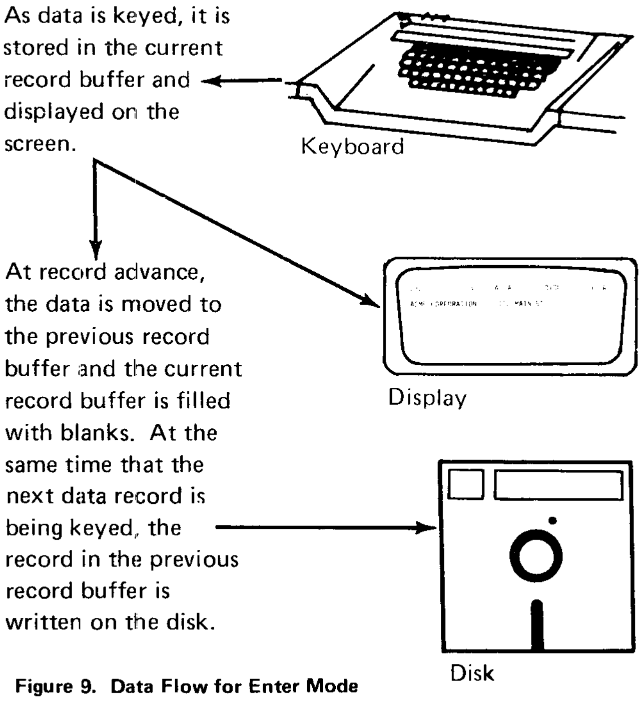
To perform a modify index function, record backspace or record advance to get to the appropriate index track record to be modified, key the record to be written, then select modify by pressing FUNCT SEL lower and M. The modify mode can be selected from the read index mode only; if it is selected from another mode, a mode selection (M) error occurs. No other mode can be selected from the modify mode. DELETE REC, RESET, and REC ADV are the only valid keys in the modify mode. All others give a function not available (F) error. The mode reverts to read index after the first keystroke following the selection of modify mode.

Enter Mode (E)

Enter mode is the most basic mode of operation. Enter mode can be selected from the index track only; a mode selection (M) error results if enter mode is selected from any other position on the disk. Enter mode is used to write new information on the disk. To select the enter mode, press FUNCT SEL lower and then press ENTER.

As the operator keys a record, the data is stored in the current record buffer until the record is completed. When a record advance occurs, the completed record is transferred to the previous record buffer. While the next record is being keyed into the current record buffer, the previous record is written on the disk.

The data flow is summarized in Figure 9. When enter mode is selected, the disk is positioned at BOE; the current record buffer and the previous record buffer are set to blanks. The station is then ready to accept data keyed into the current record buffer.



As data is keyed, it is stored in the current record buffer and displayed on the screen.

At record advance, the data is moved to the previous record buffer and the current record buffer is filled with blanks. At the same time that the next data record is being keyed, the record in the previous record buffer is written on the disk.

Figure 9. Data Flow for Enter Mode

Alternate Record Advance in Enter Mode

An alternate method of record advance in enter mode can be selected by pressing FUNCT SEL lower and then R. A dash will appear in position 39 in the status line. The record advance function in enter mode is changed so that all fields that follow the position of the cursor at the time of record advance are not executed in contrast to these fields being executed in normal enter (E R) mode. The alternate record advance function has this advantage: When REC ADV is pressed, the record is always written on the disk exactly as the operator sees it on the display (no automatic duplicate or skip fields are executed). The normal record advance may cause changes in automatic duplicate and skip fields that are beyond the cursor position at the time of record advance.

To return the machine to the normal enter mode, press FUNCT SEL lower and K. Inserting a disk or turning the power on will also establish the normal mode.

Right Adjust Fields

Programmed right adjust fields allow data entered into the current field to be moved to the right field boundary by pressing the right adjust key. Right adjust fields must be at least two characters long and can be as long as a whole record. In enter mode, the right adjust key is used to exit a right adjust field. When the right adjust key is pressed, the 3742 shifts the characters previously entered in the field (up to the cursor position) to the right field boundary. Fill characters are inserted to the left of the first keyed character as required. The fill character used depends on the field definition character used for the field. See *Right Adjust Fields* in chapter 3.

Pressing the right adjust key in the first position of a right adjust field enters fill characters into the entire field.

The dash key, when used in a right-adjust field that is programmed for numeric shift, gives a negative value to the number and then causes the field to be exited. The units (rightmost) position of the negative number contains an alpha equivalent if the digit is 1 through 9 (1 = J, 2 = K, 3 = L, 4 = M, 5 = N, 6 = O, 7 = P, 8 = Q, 9 = R) or the hex code D0 if the digit is 0. The numbers in the other positions of the right-adjust field have their numeric value. For example, if 21 is entered in a five-position zero fill right-adjust field and the dash key is pressed, the field will contain 0002J (negative 21).

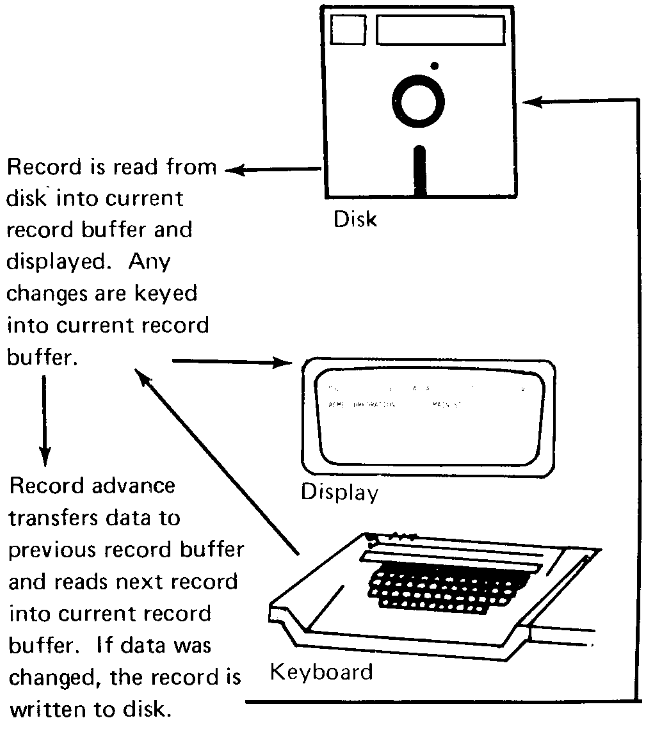
Note: Performing a character advance or field advance will cause the right adjust field to be exited in enter mode.

Update Mode (U)

Use the update mode to read records from the disk for review or modification. To select the update mode, press FUNCT SEL lower and then press UPDATE.

Data flow is summarized in Figure 10. Update mode can be selected from the read index, search, or verify mode. If update is selected from the index track, the disk is positioned at BOE; if update is selected from the verify mode, the record at the current disk address is reread. Selecting the update mode from the enter mode results in a mode selection (M) error. However, the mode automatically changes to the update mode when record backspacing occurs during the enter mode. The search mode also automatically changes to update mode at the end of a search operation. An empty data set (Y) error occurs if update mode is selected when the end of data address equals the beginning of extent address; in other words, when the data set is empty.

If record advance reaches EOD during update mode, the mode of operation automatically changes to enter mode. However, an automatic return to index track is performed if the data set is protected by a write protect byte in the label (a write protect, G, error is displayed) or if the data set is full (an end of extent, E, error is displayed).



Record is read from disk into current record buffer and displayed. Any changes are keyed into current record buffer.

Record advance transfers data to previous record buffer and reads next record into current record buffer. If data was changed, the record is written to disk.

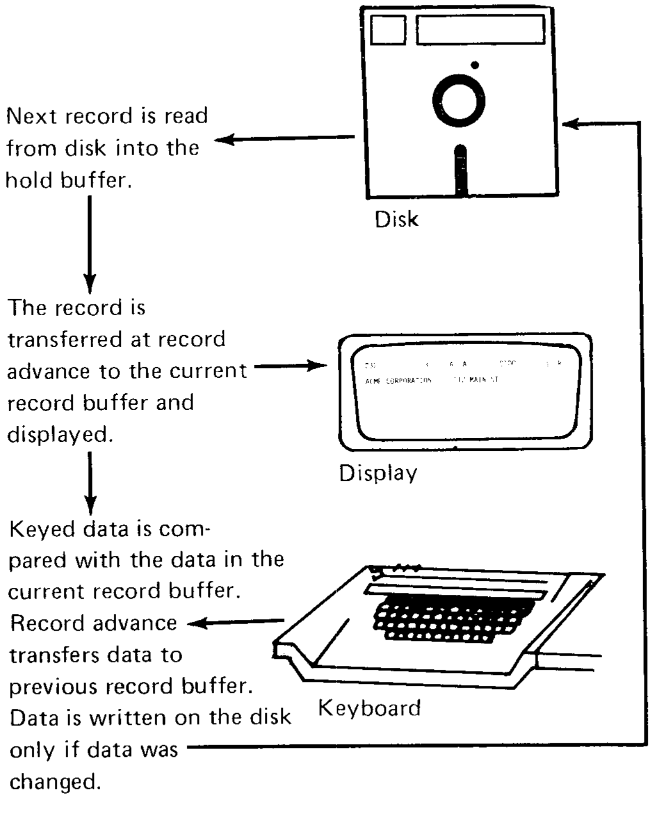
Figure 10. Data Flow for Update Mode

Verify Mode (V)

Use the verify mode to check the accuracy of data records already on the disk and to make corrections to the records when errors are found. The process of verifying data is similar to that of entering data. As the data for a record is rekeyed from the source document, each character is automatically compared with the corresponding character in the record in the current record buffer. If the characters do not compare, a verify mismatch (V) error occurs.

To select the verify mode, press FUNCT SEL lower and then press VERIFY. Select the verify mode from read index or update mode. For data flow, see Figure 11. When verify mode is selected, the previous record buffer is automatically filled with blanks, and the first non-deleted record is read into the current record buffer.

Verify mode can also be selected from verify mode. Selecting verify mode from verify mode is provided in the event that the operator discovers, after making some corrections to the record being displayed, that the wrong source record is being used to verify the record being displayed. By selecting verify mode from verify mode, all changes made to the record are ignored and the entire record is redisplayed.



Next record is read from disk into the hold buffer.

The record is transferred at record advance to the current record buffer and displayed.

Keyed data is compared with the data in the current record buffer. Record advance transfers data to previous record buffer. Data is written on the disk only if data was changed.

Figure 11. Data Flow for Verify Mode

Display

During verification, the data record is displayed up to, but not including, the cursor position. However, the *entire* record is displayed (1) after the record is moved into the current record buffer, (2) after a record backspace operation, or (3) when an error occurs. The keystroke following the RESET key returns the normal verify display mode.

Error Correction

If a character keyed while verifying is not the same as the corresponding character in the current record buffer, the following occur:

1. The keyboard locks.
2. A verify mismatch (V) error code is displayed on the status line.
3. The cursor is under the position in error.
4. The screen flashes and the entire record is displayed.

Use RESET to reset the error condition and then key in the correct character.

The first character key following the error is accepted if the character keyed agrees with either the character in the current record buffer or the character that caused the error. If the first character key pressed following the error is not accepted, a verify mismatch (V) error is again displayed. The correction process is continued until all corrections have been made to the record.

Skip Functions

If SKIP is pressed or a programmed skip field is encountered during verification, the entire field or the remainder of the field is checked for blanks. If a non-blank character is encountered, the cursor stops in that position, the skip action terminates, and a verify mismatch (V) error is displayed. After pressing RESET, the non-blank character can be replaced with a blank by pressing SKIP. The station then continues to check the remainder of the field for blanks.

Duplication Functions

Pressing DUP or entering a programmed duplication field during data verification causes the rest of the field to be verified automatically. The characters of the record in the current record buffer are compared with the corresponding characters in the previous record buffer. If one of the characters in the record disagrees with the corresponding character in the previous record buffer, a verify mismatch error is posted. After RESET is pressed, the character from the previous record buffer can be inserted into the current record buffer by pressing DUP. The device then continues comparing the field.

When using the auxiliary duplication feature, the characters in the current record are compared with the corresponding characters in program buffer 4.

Right-Adjust Fields

If a right-adjust field is encountered while verifying data, the cursor stays in the first position of the field until the first data key is pressed. The fill characters are compared depending on the field definition characters, and then the first data character keyed is compared with the first non-fill character in the field. If neither agree, an error occurs. When fill characters are in error, the cursor is at the start of the field. When data does not match, the cursor is at the first non-fill character.

To exit from the right-adjust field after verifying the last character in the field, press either RIGHT ADJ or the dash (–) key. RIGHT ADJ and the dash key also verify the field's sign, unless the field is alphabetic. (The dash key is used to verify a numeric field having a minus sign.) If the right-adjust field is alphabetic, the last character in the field is verified in the same way as a character in any other position in the field.

If a verification error occurs while verifying the sign of a right-adjust field, a right adust sign (X) error is displayed. The sign can be changed by pressing of RIGHT ADJ or the dash key twice.

Press RIGHT ADJ or the dash key, if applicable, to exit from every programmed right-adjust field. Press the key even when one right-adjust field immediately follows another and when right-adjust fields contain all fill characters. Failure to press RIGHT ADJ or the dash key at the end of a right-adjust field causes right adjust (R) error on the next data keystroke.

When a right-adjust field is the last field in an 80-character record on the3742 or a 128-character record on the 3742 with the 128 feature and the field has been verified but RIGHT ADJ has not yet been pressed, the cursor position indicator is 129 for the 3742 with the 128 feature, or 81 for the 3742, and the cursor is not displayed.

In verify mode, RIGHT ADJ or the dash key should be used only when the cursor is in the first position of the right-adjust field (to verify a zero or a blank field) or after the units position was keyed (the cursor will have moved to the first position of the next field). If these keys are pressed when the cursor is in any other position, a right adjust (R) error occurs.

Signs

If the units position of a numeric right-adjust field contains a negative number (D zone), the sign and the value of the field are verified separately. The first keystroke for the units postion verifies the digit. If the character is corrected, only the digit portion is changed. The sign (D zone) is not affected. To exit from the field, use the dash key. The dash key also verifies the sign.

If the units position of a right-adjust field has a D zone (dash) but the field is alpha, the entire character is compared. If corrected, the entire character is changed. Exit from the field using RIGHT ADJ. To verify a dash character in a numeric right-adjust field, press ALPHA SHIFT with the dash key.

Hexadecimal Data

Verify hexadecimal data in the same way it is entered:

1. Press HEX

2. Press two character keys corresponding to the hexadecimal digits wanted. The character is verified after the last keystroke for that character.

Record Advance

When a record advance is performed on a corrected record, the record is rewritten on the disk as corrected. If there were no corrections to the record, it is not rewritten during record advance. If no verification or correction to a record has taken place and the cursor is in the first manual position of the record, pressing REC ADV causes a record advance function to occur without verification of the record. If a correction had been made to the record, and the cursor is positioned at the first manual position of the record, pressing REC ADV will not cause a record advance function and the operator is forced to verify the record.

Pressing record advance when the cursor is in other than the first manual position of a record causes all remaining automatic skip and manual fields to be verified as blanks; remaining automatic duplicate fields are verified and bypass fields are bypassed.

Verification Code

After the last record in a data set has been verified, a verification code (V) is automatically inserted in the verify field of the data set label, the mode of operation changes to a read index mode, the data set label is displayed, an end of extent (E) error is displayed, and program level 0 is selected.

Field Correct Mode (C)

Field correct mode may be used to correct an entire field with a minimum number of keystrokes. The field correct mode can be selected from the verify mode only. To select the field correct mode, press FIELD COR. The cursor moves to the first position of the current field, and the field can be rekeyed as if in the enter mode. When the end of the field is reached, the cursor then moves back to the first position of the field and the mode changes to verify.

Selecting field correct mode when not in verify mode or while processing a programmed automatic field in verify mode results in a mode selection (M) error. FIELD ADV and CHAR ADV are valid in the field correct mode. The field correct mode changes back to the verify mode when a field or character backspace moves the cursor out of the current field.

Search Mode (S)

Use search mode to:

- Search on record address from read index, update, or enter mode.
- Search to end of data from read index, enter, or update mode.
- Search on content from read index or update mode.
- Search on sequential content from read index, update, or enter mode.

For data flow information, see Figure 12.

Note: After pressing any search key, except search EOD, but before pressing REC ADV, the search operation can be canceled by pressing FUNCT SEL lower and RETURN TO INDEX or by pressing FUNCT SEL lower and UPDATE.

Search on Record Address

Search on record address allows you to search through a data set using record addresses. Search on record address is especially useful if specific records have to be updated and you know the addresses of the records.

To search for a record in a data set, key its track and sector address. The specified address must have the tormat TT0SS where TT is the track number and SS is the sector number. The record address is keyed into positions 1 through 5 of the display.

The track and sector address specified can be the beginning of extent address, the end of data address minus one, or the address of any record between these two addresses. Select the search mode to do a search by record address operation from the read index, update, or enter mode. Selecting the search by record address function from any other mode causes a mode selection (M) error.

The search stops on the record address specified and the record at the specified address is displayed. The current record buffer and the previous record buffer contain the displayed record at the end of the search and the mode changes to the update mode. The record can now be updated or reviewed. Once the record has been updated or reviewed, another search can be initiated from this address.

A search address (I) error is displayed if the search address is specified incorrectly, if the address is greater than or equal to the end of data address or less than the beginning of extent address for the data set, or if the address is on the index track. If an invalid address was entered, the cursor moves to the first position of the data line on the display screen.

Search on record address can be used to continue a verify job. This is done by initiating the search and then selecting verify mode when the record at the specified address is displayed.

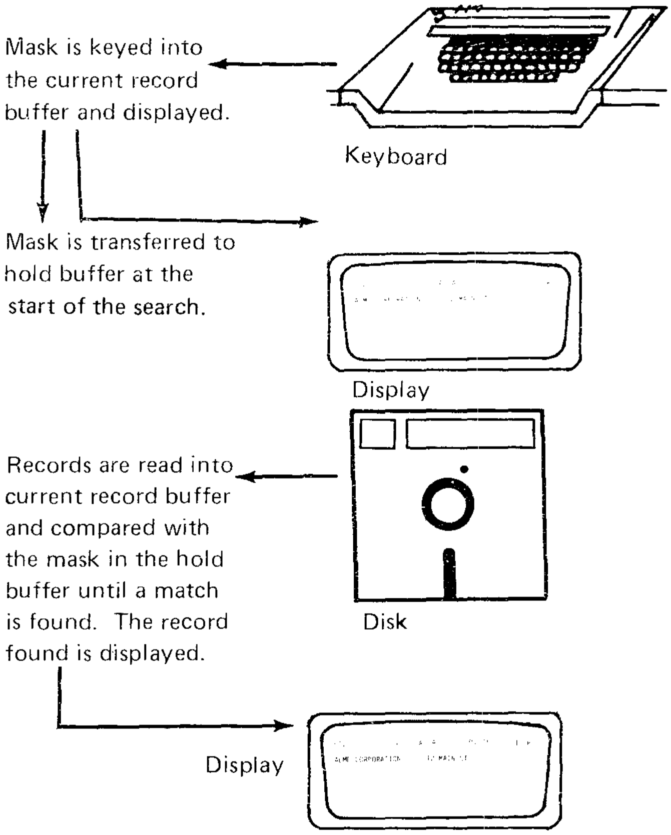

Mask is keyed into the current record buffer and displayed.
Keyboard
Mask is transferred to hold buffer at the start of the search.
Display
Records are read into current record buffer and compared with the mask in the hold buffer until a match is found. The record found is displayed.
Disk
Display

Figure 12. Data Flow for Search Mode

Search to End of Data

Search to end of data is used when you want to search to the last data record in a data set and begin adding new records to the data set.

To search to the last record of a data set press FUNCT SEL lower and then press SEARCH EOD. Select the search to end of data function from the read index, enter, or update mode. Selecting from any other mode results in a mode selection (M) error.

When the search is completed, the current disk address is set equal to the last record in the data set and the record is displayed. The contents of the previous record buffer is set equal to the current record buffer and the mode of operation changes to update.

Search on Content

Use search on content to find records that are identified by the content of specified character positions. The specification of the identifying characters and their position is called a mask. The mask can be any one or more positions of a record or may include the entire record. The mask character positions do not need to be contiguous. The search will stop only on a record that contains an exact match between the record and the search mask. The blank character is used to identify positions that are not included in the mask; therefore, the blank cannot be used successfully as a character within a mask. For example, if a certain field in a record contains ABȼCD and the search mask ABȼCD is used, the search will stop on the first record that contains AB in the first two positions of the field and CD in the last two positions of the field. Any of the 256 EBCDIC characters might be in the third position of the field.

Note: On the 3742 with the 128 feature the search mask includes all 128 characters. The operator must be certain that the portion of the record not displayed is correct. To blank the current record buffer, press FUNCT SEL lower and DELETE REC.

Searching by content is accomplished in a forward direction, beginning with the next record. The records contained in the current sector and all previous sectors are not checked.

Select the search on content function from the read index or update mode. Selecting this function from any other mode causes a mode selection (M) error to occur.

Following a search operation, the previous record buffer contains the same data as in the current record buffer, and the 3742 is in update mode. If the entire data set has been searched and no record found, the search stops and a search (S) error is displayed.

To continue searching for records using the same mask, press FUNCT SEL lower, SEARCH CONTENT, and REC ADV in that order. Do not reenter the search mask.

If a deleted record is read during a search on content and the data in the deleted record does not match the mask, the search continues and a deleted record (6) error appears momentarily on the status line. However, if the data in the deleted record matches the mask, the search stops and the deleted record is displayed.

Ocassionally, when searching on fields that are variable length, the search may stop on a record in which a portion of the field in the record being displayed matches the mask. This is because the machine takes the mask, in the positions keyed, and compares the mask to records in the data set searching for the data that matches the mask. (Remember that the blank positions in the mask identify the positions of the record that will not be checked.) Therefore, a portion of a field could satisfy the mask even though it is not the record the operator desires. For example, if a mask of SMITH was keyed in positions 1 through 5 and the search was initiated, it is possible that the machine may find SMITHSON (with SMITH in positions 1 through 5) to be a match if the machine encountered SMITHSON before it encountered SMITH.

If the machine stops the search with the wrong record displayed, the operator should continue the search content operation from the address where the search stopped to see if the record can be found.

To reduce or eliminate the possibility of the machine stopping a search on a record that does not completely match the mask, it is suggested that the search be performed on fields that contain no blanks.

Search on Sequential Content

Search on sequential content is similar to search on content, but is faster. Search on sequential content requires a mask just as search content does. Use search on sequential content only when the records in a data set are arranged so that the search fields are in ascending order (the search field of each succeeding record must have a higher value than the preceding record). See Figure 13 for the collating sequence which can be used to determine character values and which characters have higher values than others. Having the search fields in ascending order permits faster searching because of the skipping of records.

Search on sequential content operates in either a forward or backward direction; therefore, a record can be located when the search is started at a higher address than that of the record being searched.

EBCDIC Collating Sequence

Order in the Sequence	Character	Corresponding Hexadecimal Number
1 (lowest)	blank	40
2		4A
3	.	4B
4	<	4C
5	(	4D
6	+	4E
7	|	4F
8	&	50
9	!	5A
10	$	5B
11	*	5C
12	)	5D
13	;	5E
14	¬	5F
15	- (minus)	60
16	/	61
17	,	6B
18	%	6C
19	_ (underscore)	6D
20	>	6E
21	?	6F
22	:	7A
23	#	7B
24	@	7C
25	' (apostrophe)	7D
26	=	7E
27	"	7F
28	A	C1
29	B	C2
30	C	C3
31	D	C4
32	E	C5
33	F	C6
34	G	C7
35	H	C8
36	I	C9
37	J	D1
38	K	D2
39	L	D3
40	M	D4
41	N	D5
42	O	D6
43	P	D7
44	Q	D8
45	R	D9
46	\	E0
47	S	E2
48	T	E3
49	U	E4
50	V	E5
51	W	E6
52	X	E7
53	Y	E8
54	Z	E9
55	0	F0
56	1	F1
57	2	F2
58	3	F3
59	4	F4
60	5	F5
61	6	F6
62	7	F7
63	8	F8
64 (highest)	9	F9

Figure 13. Collating Sequence

If a search fails to find the record sought, a search (S) error is displayed. Failure to find a record can occur even though the record is in the data set if one or more records do not meet the requirement of ascending order. Therefore, after a search failure, a search on content may be initiated using the same search mask. To do this operation, first return to the index track in order to search through the entire data set.

Select search on sequential content from the read index, enter or update mode. When a match occurs, the record is displayed and the 3742 is in update mode.

Occasionally, when searching on fields that are variable length, the search may stop on a record in which a portion of the field in the record being displayed matches the mask. This is because the machine takes the mask in the positions keyed, and compares the mask to records in the data set searching for the data that matches the mask. (Remember that the blank positions in the mask identify the positions of the record that will not be checked.) Therefore, a portion of a field could satisfy the mask even though it is not the record the operator desires. For example, if a mask of SMITH was keyed in positions 1 through 5 and the search was initiated, it is possible that the machine may find SMITHSON (with SMITH in positions 1 through 5) to be a match if the machine encountered SMITHSON before it encountered SMITH.

If the machine stops the search with the wrong record displayed, the operator can backspace a few records to see if the proper record can be found. If it cannot be found, the operator should perform a return to index function and initiate the search content function using the same mask to see if the record can be found.

To reduce or eliminate the possibility of the machine stopping a search on a record that does not completely match the mask, it is suggested that the search be performed on fields that contain no blanks.

Offline Field Totals Mode (F)

Field totals can be used for batch auditing, a process sometimes referred to as "hash totals," or for other applications that require totals. The offline field totals feature is used to obtain totals from records that have already been entered, updated, or verified. Offline field totals mode can be selected from read index or update mode.

HEXADECIMAL DATA

The HEX key provides a method of entering any of the 256 EBCDIC characters into a data record. Valid hex digits are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Hex characters are entered by pressing HEX and then keying two hex digits. Hexadecimal data is verified in the same manner. HEX can also be used like a multipunch key on a keypunch machine except that the characters combined are sometimes different from the keypunch, and they must be keyed in a certain order.

Multipunch Operations

HEX can be used to perform the equivalent of multipunch on card keypunch machines. The important multipunch operations are overpunching with 11, 12, and 0 punch. The following procedures are valid for overpunching numerics 0 through 9.

11 Punch: For the 11 punch, first press HEX, then press the D key or the dash (–) key, and then the numeric key (0 through 9). The 11-over punch in the units position of a numeric field indicates a negative value.

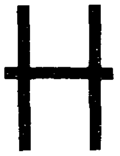
12 Punch: For the 12 punch, first press HEX, then press the C key and then press the numeric (0 through 9).

0 Punch: For the 0 punch, first press HEX, then press the E key and then the numeric key if the numeric is 2 through 9. If the numeric is a 1, the procedure is to press HEX, then press the 6 key and then the 1 key. If the numeric is a 0, do not use HEX, just press the 0 key. (Overscoring a zero with a zero on a punched card punches the same hole twice resulting in no change from a single zero punch.

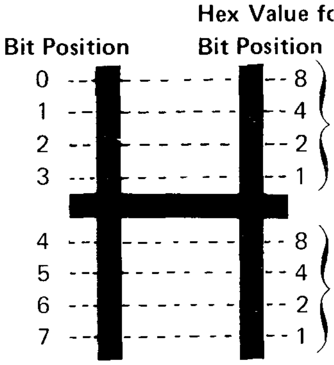
Hexadecimal Display

When hex data is keyed or displayed, not all of the data represents displayable characters such as an A B C . . . or a 0 1 2 3. . . . When the hex data keyed is not a displayable character, a hexadecimal display is provided that represents the eight bit code for the data.

The basic display (no bits on) looks like this:



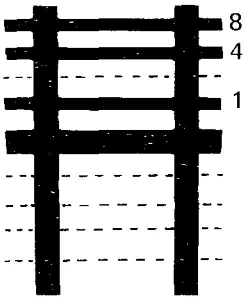
Other lines are added to the display for each bit that is on in the EBCDIC code.



Add these values together to get the first hex digit. Only add those values together for the lines displayed.

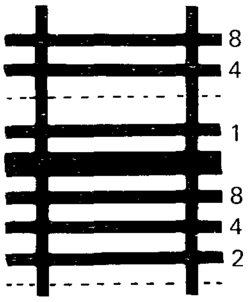
Add these values together to get the second hex digit. Only add those values together for the lines displayed.

Example: Assume the display looks like this:



The first hex digit is 8 + 4 + 1 = D and the second hex digit is a 0. The hexadecimal value is hex D0.

Assume another display looks like this:



The first hex digit is 8 + 4 + 1 = D and the second hex digit is 8 + 4 + 2 = E. The hexadecimal value is hex DE.

DELETED RECORDS

Every sector of a disk has two address marks, one of these address marks (data address mark) immediately precedes the first character of the record. This address mark identifies a record as either a non-deleted or a deleted record. The data address mark is not part of the data and is not displayed.

When a record is deleted, the data address mark is altered to identify the record as deleted. Depending on the procedure used to delete the record, the first character of the data record can be preserved or automatically replaced with the character D. See *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136, for information on how to delete a record and automatically replace the first character with a D. To delete a record and preserve the data in the first position, press FUNCT SEL lower and then press NUM SHIFT with DELETE REC.

Note: When the 3747, 3540, or any other system reads a record, it examines both the data address mark and the first position in the data record for a D. If the data address mark identifies a deleted record, but the first position in the data record is not a D, a halt will occur with an error condition. Therefore, if the diskettes are to be processed by other than a 3741 Model 1 or 2, or a 3742, or if a data set is used as input from a directly attached 3741, the procedure for deleting records and preserving the first data character should not be used. If the diskettes are processed by a 3741 Model 3 or 4, see *IBM 3741 Models 3 and 4 Programmable Work Station Programming Reference Manual,* GA21-9194, for information on deleted records requirements. Also for more information, see *Control Records* in the *IBM Diskette General Information Manual,* GA21-9182.

When the 3742 reads a record, it examines the data address mark to see if the record is a deleted record. If the data address mark identifies a deleted record, a 6 error is displayed in the status line. The 3742 does not check for a D in the first position of the data record, therefore, the operator must use the 6 error to identify deleted records.

To change a deleted record to a non-deleted record, the record must be rewritten. To do this, key a change to the record and press REC ADV. To rewrite a record in the Index mode, press FUNCT SEL lower, press M, and then press REC ADV.

Note: In the Index mode a record is rewritten even if no change is keyed to the record. That is, the record has a D in the first position but the record is not a deleted record.

Mode Considerations for Deleted Records

Enter Mode: Using the DELETE REC during enter mode blanks the current record buffer and positions the cursor at the first manual position. The previous record buffer is unchanged.

Update Mode: If, during update mode selection, a deleted record is read into the current record buffer, the previous record buffer will contain the same deleted record. If a deleted record is in the current record buffer when a record advance occurs during update mode, the record is not transferred to the previous record buffer. If a deleted record is encountered during record backspacing, the previous record buffer is not changed.

When deleting a record, the first character is changed to a D in the current record buffer, and the remainder of the content in the current record buffer remains unchanged.

Verify Mode: When selecting verify mode from the index track, the first non-deleted record in the data set is read into the current record buffer.

If verify mode is selected from update or verify mode and the currently displayed record is a deleted one, the next non-deleted record is read into the current record buffer. In either case, the previous record buffer is blanked.

When a record advance occurs and a deleted record is in the hold buffer (next sequential record is deleted), the next non-deleted record is read into the current record buffer bypassing all deleted records encountered. If a deleted record is encountered during record backspacing, the deleted record is read into the current record buffer with the previous record buffer unchanged.

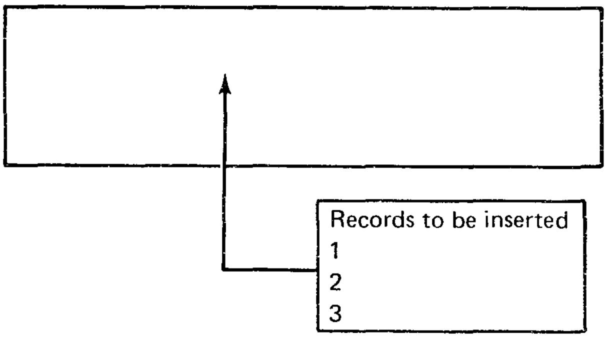
When deleting a record, the contents of the previous record buffer remains unchanged and a record advance function is performed.

Search Mode: Using DELETE REC during search mode blanks the current record buffer and positions the cursor at the first manual position. The previous record buffer and the record at the current disk address are unchanged. If the search operation terminates on a deleted record, both the current and previous record buffers contain the deleted record.

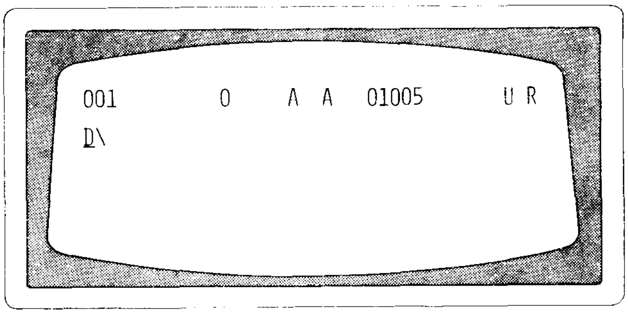
RECORD INSERT

With this feature, you can insert records within a data set as shown below:

BOE	Current Record Address	EOD	EOE
01001	01005	10026	15026



To insert records into a data set, the record insert function performs an operation which creates space in the data set for the number of records to be inserted. The 3742 puts deleted records in the space created. The deleted records have a D in position 001 and a \ (inverse slash) in position 002. The remainder of the record is blank.



```
001        0    A  A   01005       U R
D\
```

There are two ways to do record insert on the 3742:

- Continue to use both keyboards during the record insert operation.
- Lock out the other keyboard during the record insert operation. This way is faster; however, any programs loaded in storage areas 7, 8, 9, or A are lost. The 3742 uses these storage areas when moving the records.

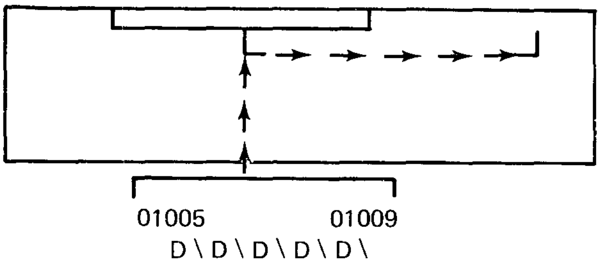
Note: If the Function Control Feature (3742 RPQ843266) is installed, do not perform the faster method of record insert from the primary station. Alteration of diskette data or program level content could result. Doing either way of record insert from the secondary station is not affected.

If you want to insert five records on track 01 beginning with sector 05, the 3742 would put five deleted records in sectors 05 through 09.

To create space in the data set, the 3742 moves the records from the current record address through the last record in the data set. The 3742 moves these records as many sectors as the number of records to be inserted (up to 99).

An example of creating space for five records:

BOE	Current Record Address	Last Record in Data Set	Last Record in Data Set After Move
01001	01005	10020	10025



Space filled with deleted records.

When using the record insert function, there must be enough space between EOD and EOE specified in the data set label. That is, if you want to insert 50 records but only have 40 sectors available (from EOD to EOE) in the data set, an error occurs.

If all of the space created is not used, the remaining deleted records will be ignored when processing data. Also, by having deleted records, you must reset the deleted record (6) error which the 3742 displays with each deleted record. Therefore, if enough space was not specified for the number of records to be inserted, you will realize it when you do not get the deleted record (6) error. Search on content can be used to search for the D\ when records will be inserted at some other time.

Record Insert Mode Characteristics

Enter Mode

No records are moved in this mode but it allows you to create a space within your data set for records you might want to add later. Deleted records are added starting at the current disk address for the number of records specified.

Update or Verify Mode

The machine searches to the last record in the data set. Then it reads records in groups starting at this address, and rewrites these records at their new addresses. After moving all the records, the 3742 adds deleted records at the specified disk address for the number of records specified.

Read Index Mode

The 3742 moves all records in the data set as if in Update mode. Then the machine adds deleted records at the BOE address.

Search Mode

The action taken by the machine depends on the mode it was in prior to selecting Search mode. See one of the above.

When all the records have been moved and the space filled with deleted records, the 3742 returns to the disk address of the first record of the space created.

The 3742 goes to update mode regardless of the mode you were previously in. The machine also displays a deleted record (6) error, goes to program 0, and blanks the hold buffer.

ADDING RECORDS TO A BATCH

Verify operations, batch control totals, and 3747 record counts can indicate that records are missing. Records that are missing may be added to the end of the batch, or may have to be added somewhere in the middle of the batch. Records may also have to be added to a batch if they become available after the batch to which they belong has been keyed.

The easiest way to add records to a batch is to add them to the end of the batch. To do this, the operator searches (by using search EOD operation) to the end of the data position in the data set. When the search is finished, the last record in the data set is displayed. The operator then presses REC ADV to get to the next record position where data can be entered. The missing records can then be added to the data set.

Some data sets may consist of records that are sequence sensitive. To handle addition of records to sequence sensitive data, the following procedure can be used on 3742s with the disk copy feature:

1. Determine the proper address at which records must be inserted, according to the sequence of data.

2. Insert the diskette from which records will be copied into the secondary station's disk drive.

3. Insert a scratch diskette (the records will be copied onto this diskette) into the primary station's disk drive.

4. Press FUNCT SEL lower and A on the secondary station.

5. Press FUNCT SEL lower and ENTER on the primary station.

   Note: All of the following steps are done on the primary station.

6. Press FUNCT SEL upper and DISK 2 RET TO INDEX.

7. Press FUNCT SEL lower, hold down the numeric shift key, press SEARCH ADDRESS, key in the address of the next record after the address at which you wish to begin to write.

8. Press FUNCT SEL upper, hold down the numeric shift key, and press COPY.

9. When the address entered is reached, all the records have been copied onto disk 1 except for the record displayed.

10. Key in the records to be added.

11. Press FUNCT SEL upper, hold down the numeric shift key, and press COPY. This step copies the remainder of the records still on disk 2 onto disk 1.

12. When the copy is complete, press FUNCT SEL lower and RETURN TO INDEX.

DATA RECOVERY

The most significant recovery step in data recovery is to re-establish the EOD address of the partially destroyed data set. The old EOD address is destroyed on a data set label under the following conditions:

- The operator inadvertently selected a data set label of a data set that contains currently valid data, selected enter mode, performed at least one record advance and then returned to the index track.

- The operator removed the diskette from the disk drive before return to index was pressed or before X R was displayed on the status line when RETURN TO INDEX was pressed.

Recovery of the data sets original EOD address, if known, is a simple data set label modification of the EOD field.

If the EOD address is not known, one of the following procedures should assist in determining the EOD address for the data set by finding the last record of the data set.

Procedure 1:

- Modify the EOD field in the data set label with an estimated EOD address.

- Execute a search EOD operation and examine the record displayed. If the displayed record is recognized as a valid record within the data set but is not the last record, return to the index track and repeat *procedure 1* using a higher estimated EOD address. If the displayed record is not recognized as a valid record within the data set, press record backspace until the last record in the data set is found.

Procedure 2:

- If the last record content is known, modify the EOD field in the data set label to the EOE address plus one and perform a search on content operation using the last record as the search mask. When the search stops, the last record entered should be displayed.

When the last record of the data set is found, perform the following:

1. Press REC ADV.

2. Write down the address.

3. Perform a return to index operation.

4. Modify the EOD field in the data set label with the recorded address.

5. Execute a search EOD operation to validate the EOD address.

6. If the operator had originally keyed over currently valid data, the records should be examined and corrected.

On occasion, an operator will remove a diskette from the disk drive before pressing RETURN TO INDEX. When this happens, there is the possibility that the EOD address will not be correct because the only time that the address is updated is when the return to index function is performed. The preceding procedures should also be used for this situation.

BAD SPOTS ON DISK

If a record cannot be read from or written onto disk, the copy function can be used to copy surrounding data and then manually key the bad record. Alternate methods depend on the mode the station is in when the error occurs: Enter mode – return to index and continue the batch on a new diskette. Verify or Update mode – backspace several records and then record advance to see if the problem will clear itself; if not, delete the record and add it to the data set (see *Adding Records to a Batch* in this section).

The following are recovery procedures to follow when disk errors 2, 3, 4, and 5 persist.

No Record Found (2) Error

A no record found error occurs when the address that identifies the location of a record on disk is defective. For verify mode, the record on which the no record found error occurred is the next record to be verified, not the current record being verified and displayed. The operator can reset the error, complete verification of the current record, and then press REC ADV. The record displayed after pressing REC ADV is the record just verified. For all other modes, the record read prior to the no record found error is displayed.

When the no record found error persists, perform the appropriate copy operation to copy the data. The *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136 provides copy procedures. Follow the procedures in the operator's guide for error recovery when the no record found error occurs while copying. Be sure to identify the diskette on which the no record found error occurred as being defective. The defective diskette should be initialized.

Seek (3) Error

A seek error occurs when the address that identifies a track is defective. When the seek error occurs, the 3742 either returns to the index track or remains within the data set depending on the mode of operation. When the seek error persists, perform the appropriate copy operation to copy data. The *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136 provides copy procedures. Follow the procedures in the operator's guide for error recovery when the seek error occurs while copying. Be sure to identify the diskette on which the seek error occurred as being defective. The defective diskette should be initialized.

Disk Read (4) Error

A disk read error occurs because a portion of the record, or the entire record read is in error. However, the record on which the read error occurred is displayed even though a portion or all of the record is in error. If the read error persists after being reset several times, the following can be performed to try and correct the error.

Note: For verify mode, the record on which the read error occurred is the next record to be verified, not the current record being verified and displayed. Therefore, the operator can reset the error, complete verification of the record, and then press REC ADV. The record on which the read error occurred is then displayed.

If you can identify what data is incorrect in the record, correct the data and press record advance. If no write error (5) occurs, the record has been written to disk correctly and it should be read properly.

If a write error does occur after pressing record advance, the sector is probably defective. Reset the write error and perform a delete record operation. If the delete record operation is successful (no write error), add the record to the end of the data set.

If the delete record operation is not successful, or if the record cannot be added to the end of the data set, the copy operation can be used to copy the surrounding data and then manually key in the record. Be sure to write the track and sector address on the external label of the diskette on which the error occurred. The diskette should be initialized.

Disk Write (5) Error

A write error occurs when a record is not written on disk correctly. When a write error persists, perform a delete record operation. If the delete record operation is successful (no write error), add the record to the end of the data set.

If the delete record operation is not successful, or if the record cannot be added to the end of the data set, the copy operation can be used to copy the surrounding data and then manually key in the record. Be sure to write the track and sector address on the external label of the diskette on which the error occurred. The diskette should be initialized.

CE MODE

CE mode is used by the customer engineer when testing or troubleshooting the equipment. This manual does not describe how this mode works or how it is used but merely points out how it is entered and how to get out of the mode.

Pressing FUNCT SEL lower, NUM SHIFT, and C puts the machine in the CE mode. The status line displays CER in positions 38, 39 and 40.

Pressing RESET several times returns the machine to the index mode with the label from sector 08 displayed. The CE mode may modify the EOD field and the verify indicator field of the data set label for the data set being used when the CE mode was selected. Therefore, those fields should be checked and corrected if in error.

Chapter 3. Program Information

Programs similar to those used on the IBM 029 Card Punch can be written for use on the 3742. A program is written to control the record format and assist the operator as records are keyed. Programs are first keyed into the current record buffer (as data is keyed in enter mode) from which they can be stored on the disk (for later program loading and selecting) or stored directly into a program buffer. If programs are stored on disk, they can be stored in a data set separate from the data or on a program disk (a disk that contains only programs). For information on loading a program, selecting a program, displaying a program, or modifying a program, see *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136.

The 3742 provides storage for up to six programs, numbered 1 through 6. The 3742 with the 128 feature provides storage for up to ten programs, numbered 1 through 9 and A. The operator loads programs and selects programs by these numbers. The selected program number is displayed to the operator in the status line at position 12.

When operating under manual control program, buffer 0 is displayed in the status line. Program buffer 0 is a single character all alpha field program. When the machine is turned on, program buffer 0 is automatically selected and all program buffers (1 through 6, or 1 through 9 and A for the 128 feature) are filled with N's (numeric field).

PROGRAM CODING

A program defines the format of the record for the operator. A program defines the:

- Position at which each field in a record starts.
- Type of each field in the record.
- Length of each field.
- Last position of the record.

A program consists of field definition characters, field continuation characters, and the end of program character. A field definition character defines the type of field and indicates where each field in the record begins. (The field definition character of the field being processed is displayed in the status line). If a field is more than one position long, the field definition character is followed by the appropriate field continuation characters. (The field definition character and the continuation characters must equal the length of the field.) For a list of field definition characters, continuation characters, and their meanings, see Figure 14.

The end of program character (E) normally follows the last field definition or continuation character. Programs that are 80 characters long for the 3742 or that are 128 characters long for the 3742 with the 128 feature do not require the E character. When the cursor reaches a position containing an E character in the program, a record advance occurs if AUTO REC ADV is on. If this switch is off, the following characters in the status line are changed: The field definition character is set to E, the field continuation character position is blanked, and the cursor position is set to 00 on the 3742 or 000 on the 3742 with the 128 feature. REC ADV should be pressed to write the record on the disk. Keying data when the cursor position is at 00 or 000 causes a truncation (T) error.

Program Field Descriptions

Field Continuation Characters

Field continuation characters follow a field definition character and are used to indicate the shift (numeric or alpha) and the length of the field (the field definition character plus the number of continuation characters must equal the field length). The field continuation characters are dash (—) for numeric and period (.) for alpha. If desired, field continuation characters can be intermixed within any particular field.

Field Definition Characters	Keyboard Shift	Meaning	
N	Numeric	Numeric field	
A	Alpha	Alphabetic field	
B	Not applicable	Unconditionally bypass the field	
J	Numeric	Right adjust numeric field with blank fill characters	
R	Numeric	Right adjust numeric field with zero fill characters	
I	Alpha	Right adjust alphameric field with blank fill characters	
D*	Numeric	Automatically duplicates	
U*	Alpha	Automatically duplicates	
S*	Numeric	Automatically skips	
K*	Alpha	Automatically skips	
V	Numeric	Bypass numeric field during verify mode only	
W	Alpha	Bypass alphabetic field during verify mode only	
X	Numeric	Bypass right adjust numeric field with zero fill characters during verify mode only	
Y	Numeric	Bypass right-adjust numeric field with blank fill characters during verify mode only	
Z	Alpha	Bypass right-adjust alpha field with blank fill characters during verify mode only	
H	Numeric	Self-check, modulus 10 field	
C	Numeric	Self-check, modulus 11 field	
F*	Numeric	Self-check, modulus 10 skip field	
G*	Numeric	Self-check, modulus 11 skip field	
L*	Numeric	Self-check, modulus 10 duplicating field	
M*	Numeric	Self-check, modulus 11 duplicating field	
Any fields definition character followed by 1, 2, or 3	Not applicable	Field totals entry	
Any B field followed by 4, 5, or 6	Not applicable	Field totals readout field	
Any B field followed by 7, 8, or 9	Not applicable	Field totals readout and reset field	
– (dash)	Numeric	Continuation character for numeric fields	Continuation characters can be mixed in a field
. (period)	Alpha	Continuation character for alphameric fields	
E	Not applicable	Indicates the end of the program	

*These are automatic fields only with the AUTO DUP/SKIP on.

Figure 14. Field Definition Characters

Manual Numeric Fields

The numeric field definition character, N, defines a manual numeric field. The N is followed by the dash (—) continuation character. The number of dashes required depends on the field length. The N plus the dashes must equal the field length. If the field is only one position, only the N is required. Any position in a numeric field can be overridden by using the alpha shift key.

If a manual numeric field does not have to be verified, use the V field definition character instead of the N field definition character. The V indicates to the 3742 that, when in verify mode, the corresponding manual numeric field in the data record does not have to be verified and the 3742 will bypass the field.

Manual Alphabetic Fields

The alphabetic field definition character, A, defines a manual alphabetic field. The A is followed by the period (.) continuation character. The number of periods required depends on the length of the field. The A plus the periods must equal the field length, If the field is only one position, only the A is required. Any position in an alphabetic field can be overridden by using the numeric shift key.

If a manual alphabetic field does not have to be verified, use the W field definition character instead of the A field definition character. The W indicates to the 3742 that, when in verify mode, the corresponding manual alphabetic field in the data record does not have to be verified and the 3742 will bypass the field.

Unconditional Bypass Fields

The unconditional bypass field definition character, B, causes the 3742 to bypass the field. The B can be followed by either dashes (—) or periods (.) continuation characters. The number of continuation characters required depends on the length of the bypass field. The B plus the continuation characters must be equal to the length of the field. Field totals readout and readout/reset fields are indicated by using the B field definition character. A B followed by a 4, 5, or 6 indicates a field totals readout field, and a B followed by a 7, 8, or 9 indicates a field totals readout and reset field. Again, either continuation character can be used to fill out the remainder of the field. For complete information on programming field totals, see *Chapter 5. Offline Field Totals.*

Right Adjust Fields

Right adjust fields allow data to be right adjusted when entered into a data record. For example, assume that there is a right adjust field that is 10 positions long and 12345 is keyed into the field. Before the right adjust key is pressed, the field would look like this: DATA... 12345ƀƀƀƀƀ. After the right adjust key is pressed, the field would look like this: DATA. . .ƀƀƀƀƀ12345. The right adjust field definition characters are J, R, I, X, Y, Z. Right adjust fields must be at least two characters long.

The J field definition character defines a numeric right adjust field that has blank fill characters. Fill characters are characters that are placed in front of the right adjust data. If a numeric right adjust field with blank fill characters does not have to be verified, use the Y field definition character instead of the J field definition character. The Y indicates to the 3742 that, when in verify mode, the corresponding right adjust field in the data record does not have to be verified and the 3742 will bypass the field.

The R field definition character defines a numeric right adjust field that has zeros for fill characters. If a numeric right adjust field does not have to be verified, use the X field definition character instead of the R field definition character. The X indicates to the 3742 that, when in verify mode, the corresponding right adjust field in the data record does not have to be verified and the 3742 will bypass the field.

The I field definition character defines an alphabetic right adjust field that has blanks for fill characters. If an alphabetic right adjust field does not have to be verified, use the Z field definition character instead of the I field definition character. The Z indicates to the 3742 that, when in verify mode, the corresponding right adjust field in the data record does not have to be verified and the 3742 will bypass the field.

Duplicate Fields

Fields can be programmed as duplicate fields. This means that the field in the current record can be duplicated from the previous record. The field is duplicated automatically when the AUTO DUP/SKIP switch is on . If the AUTO DUP/SKIP switch is off, the field appears as a manual alpha or numeric field depending on the field definition character.

Duplicate fields can be alphabetic or numeric. The duplicate field definition characters are D and U. The D field definition character defines a numeric duplicate field and the U field definition character defines an alphabetic duplicate field. Follow the field definition character used (D or U) by the appropriate continuation character (dash or period) to complete the duplicate field.

An additional duplicate function called auxiliary duplication is available. For the auxiliary duplicate function, the data to be duplicated into the current programmed duplicate field in the record is obtained from program buffer 4. The data in the program buffer must be in the same positions as the field in the current record. More than one field can be duplicated from program buffer 4 if desired. All that is required is that, for each field to be duplicated, the data in the program buffer be in the same positions as the duplicate fields in the record. Program buffer 4 is loaded with the duplicate data just as if a program were being loaded into the buffer. The auxiliary duplicate function is initiated by pressing NUM SHIFT and DUP. The auxiliary duplication function is useful when data is to be duplicated into certain records, such as header records.

Skip Fields

Fields can be programmed as skip fields. This means the 3742 will fill the field programmed as a skip field with blanks and then go to the next field. In verify mode, skip fields are checked for blanks. The skip field is skipped automatically when AUTO DUP/SKIP is on. If AUTO DUP/SKIP is off, the field appears as a manual alpha or numeric field depending on the field definition character. Skip field definition characters are S and K. The S field definition character defines a numeric skip field and the K field definition character defines an alphabetic skip field. Follow the skip field definition character used (S or K) by the appropriate continuation character (dash or period) to complete the field.

Self-Check Fields

Self-check fields provide a method for checking the entries in numeric fields that contain a precomputed self-check digit. The field definition characters for self-check fields are H, C, F, G, L, and M. The F, G, L, and M characters are automatic field definition characters (AUTO DUP/SKIP must be on).

The H and C field definition characters define manual numeric self-check fields for modulus 10 and modulus 11, respectively.

The F and G field definition characters define automatic numeric skip self-check fields for modulus 10 and modulus 11, respectively. In verify mode or field correct mode, these fields are handled as bypass fields with only the self-check computation being performed when the self-check feature is installed. If the self-check feature is not installed on the 3742, the fields are handled as auto skip fields (verified for blanks).

The L and M field definition characters define numeric automatic duplicate self-check fields for modulus 10 and 11, respectively.

For more information on self-check, see *Chapter 8. Self-Check.*

Creating and Storing Programs

There are two basic approaches to creating programs based on responsibility for creation. The first approach follows the procedure used in many of today's card punch installations, where each operator has the responsibility to make up and maintain program cards. With the 3740 system, an operator can simply key and maintain programs on the disk, rather than program cards. One disk can hold as many as 1898 programs.

The second basic approach is that of having a supervisor, lead operator, or system engineer create all of the programs for every operator in the installation. Here the responsible person keys all of the programs for the installation, then makes copies of the master program disk for all operators. Each operator then has a program disk, as in the first approach; the only difference is that the program disks were prepared centrally.

Centrally prepared program disks are maintained by changing existing programs or adding new ones to the master program disk. Copies are then made and issued to each operator. The effective date of the program disk can be noted on the external label, thus indicating whether the program disk is the latest version.

With either method of program disk preparation, the programs must be organized in some manner so that they can be located again. The simplest approach is to key the programs on the disk in any desired sequence, keep track of the addresses, and then locate the record by using search on record address.

Another method is to arrange each group of programs on a separate track, for example:

Address			
Track		Sector	Programs
01	0	01	Accounts receivable
01	0	02	Accounts receivable
02	0	01	Payroll
02	0	02	Payroll
02	0	03	Payroll
03	0	01	Accounts payable
03	0	02	Accounts payable

In this example, the accounts receivable job has two program buffers, starting in track 01, sector 01. Payroll has three program buffers, starting in track 02, sector 01, and so on. By arranging each group of programs on a separate track, allowance is made for adding new programs.

If, after keying these programs onto a disk, the operator wants to load the payroll programs, the operator performs a search on record address, keys in 01 (track) 0 01 (sector), and presses REC ADV. When the first of the three payroll programs is displayed, the operator presses FUNCT SEL lower, selects PROG LOAD, and keys number 1. This loads the first program into program buffer 1. By pressing REC ADV, FUNCT SEL lower and selecting PROG LOAD, then keying number 2, the next program is loaded into program buffer 2. The third payroll program is loaded into program buffer 3 in the same way.

Another approach would be to organize the program disk as follows:

Address			
Track		Sector	Record
01	0	01	Accounts receivable (header)
01	0	02	Accounts receivable program 1
01	0	03	Accounts receivable program 2
02	0	01	Payroll (header)
02	0	02	Payroll program 1
02	0	03	Payroll program 2
02	0	04	Payroll program 3
03	0	01	Accounts payable (header)
03	0	02	Accounts payable program 1
03	0	03	Accounts payable program 2

In this example, each job has a header record recorded on the disk preceding the actual program. A header record contains descriptive information about the programs. A typical header record for payroll programs would be as follows:

PAYROLL PROGRAMS -EMPLOYEE MASTER,
DAILY TIME, DEDUCTIONS.

Messages can also be included in the header program. To find the payroll program using this approach, the operator performs a Search on Content, keys in the word Payroll, then presses REC ADV. This causes the data station to search for the Payroll header record. The operator then presses REC ADV, selects program load, and loads as many programs as needed for that job.

Note: To use the search on content approach on the 3742, the special feature group is required.

Thus far, the methods of creating and storing programs have assumed a program disk for each operator. The other basic system approach is to include programs on the same disks that are to contain data.

For simplicity, assume an installation has only 20 programs for all of the jobs to be performed on the 3742. Each disk in the installation can have a bypass data set label established for track 1 (26 sectors), in which programs are to be recorded. The second data set label can be established to reserve tracks 2 through 73 for data.

Thus, an operator can insert a diskette, search for the desired program, load it, and proceed to enter data on the same diskette. The procedure would be as follows:

1. Insert diskette.

2. Search for a program (record address or search on content).

3. Load program(s).

4. Press the return to index key. (This returns the read/write head to the index track, the label for the program data set.)

5. Press the record advance key. (This increments to the next sector on the index track, which in this case would be the data set label for the data portion of the disk.)

6. Select a mode. (The read/write head is positioned at the first sector of the data portion of the disk, and the data station is placed in the selected mode.)

In summary, rekeying programs as they are used is inefficient and an error prone practice. Programs should be stored on a diskette in one of the following ways:

1. Create a separate program diskette that contains all programs routinely used for key entry, verification, and field totals operations.

2. Record the programs on the same diskette that is used for recording and verifying data.

Whichever procedure is adopted (shared diskettes or separate program diskettes) the programs must be identifiable for the operator. Three common methods can be used:

1. Put all program formats into a single data set. Identify each program data set with a header record.

2. Put each program data set (payroll, invoicing, etc.) into a separate data set, include header information in that data set's label.

3. Put all program formats in a single data set and record the exact address of the first program in each job's set. To call a program for loading, search to the address of the first program.

There are many advantages to separate program diskettes. The most important is that chances for accidental destruction of program formats are reduced by isolating program disks, labeling them externally with color codes, and building two-disk procedures into each job that is converted to the 3742.

Displaying Programs

The currently selected program can be displayed by pressing FUNCT SEL lower and DISPLAY PROG. The program replaces the data record displayed on the screen. Attempting to display program 0 results in the current record being displayed. DISPLAY DATA is used to restore the current data record to the display after a display program operation has been performed. For more information about displaying programs, see the *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136.

Changing a Program in a Program Buffer

A program in a program buffer can be changed during data entry. This is usually done when a program that has been placed in a program buffer is in error. For information on how to change a program that is in a program buffer, see *How to change a Program that is in a Storage Area* in the *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136.

PROGRAM CHAINING

When records having different formats are being entered, different programs to control the format of the records are required. Program chaining provides the ability to automatically select programs when the format changes from one record to the next. Program chaining frees the operator from manually selecting a new program when the format of the record changes. Program chaining can be done in a forward (when record advancing) or backward (when record backspacing) manner.

Positions 79 and 80 (127 and 128 for the 128 feature) of a program are used for chaining. Position 80 (128 for the 128 feature) is used to indicate the program to select when a record advance operation occurs. Position 79 (127 for the 128 feature) is used to indicate the program to select when a record backspace operation occurs. These positions have to be filled in only when record format changes and a different program is needed.

Any valid program number can be placed into positions 79 or 80 (127 or 128 for the 128 feature) of the program. Acceptable characters for program chaining are numbers 0 through 6 for the 3742 and 0 through 9 and hex FA (corresponding to program buffer A) for the 3742 with the 128 feature.

Program chaining characters can be part of a programmed field. If the program chaining characters are part of a programmed field, they will be treated as numeric continuation characters when data is being entered, but they will also perform the program chaining function.

Program Chaining Example

The following is an example of program chaining.

Assume that a job is being entered with the program in program buffer 1 controlling the format of the data record. Program buffer 1 contains the following program:

```
A..................N------------------R---------E                  3
```

Position 79 contains no chaining character, therefore, on record backspace, the program in program buffer 1 will continue to be used to control the format of the data record.

The 3 is in position 80. This means that after the data record has been entered and record advance occurs, the program in program buffer 3 is to be selected to control the format of the next data record.

Assume now that record advance has occurred and program buffer 3 is selected. Program buffer 3 contains the following program for controlling the format of the data record to be entered:

```
A..............S....N-----------------R-----------D---E            12
```

The 1 is in position 79. This means that if a record backspace occurs, the program in program buffer 1 is to be selected. (Program buffer 1 contains the program that was used to control the format of the record being backspaced to.)

The 2 is in position 80. This means that after data has been entered and record advance occurs, the program in program buffer 2 is to be selected to control the format of the next data record.

Assume now that record advance has occurred and program buffer 2 is selected. Program buffer 2 contains the following program for controlling the format of the data record to be entered:

```
N---------A.........R---------D.........S---------V.........E          3
```

The 3 is in position 79. This means that if a record backspace occurs, the program in program buffer 3 is to be selected. (Program buffer 3 contains the program that was used to control the format of the record being backspaced to.)

Position 80 contains no chaining character, therefore, on record advance, the program in program buffer 2 will continue to be used to control the format of the data records.

Chapter 4. Production Statistics

Production statistics provide statistics on data production for use in measurement of workload, analysis of errors, and job accounting. The production statistics function provides three counters to accumulate totals on:

- Number of keystrokes.
- Number of records processed.
- Number of characters corrected during verify mode, including all characters keyed in field correct mode.

COUNTERS

Keystroke Counter

The keystroke counter counts every keystroke, except the following, in all modes of operation:

1. RESET
2. Keystrokes with a pending error condition except dash (–) or ampersand (&) when used to reset a self-check error.
3. R key when used to restore the display after displaying production statistics or field totals.
4. Any keystroke during disk copy, offline field totals, or search.
5. Any function select keying sequence.

The keystroke counter does not normally count keystrokes that cause errors. However, the following are exceptions:

1. Miskeying hex data.
2. When keying characters other than 0 through 9 in a program numeric field with the program numeric switch set to NUMBERS ONLY. (Each non-numeric keystroke is counted.)

The keystroke counter is a six-position counter that can accumulate up to a total of 999,999. When 999,999 is reached, the counter is reset to 000,000. No overflow condition is indicated.

Note: The keystroke counter contents is in error after any disk 2 search or copy operations (primary station only).

Record Counter

The record counter counts the number of records processed in the enter, update, and verify modes. Record advances in update mode are not counted unless data has been entered into the current record buffer (a change has been made to the record). Record advances in verify mode are not counted if performed from the first manual position in the record. The record counter is a six-position counter that counts up to a maximum of 065,535. When the counter reaches 065,535, it is reset to 000,000. No overflow condition is indicated.

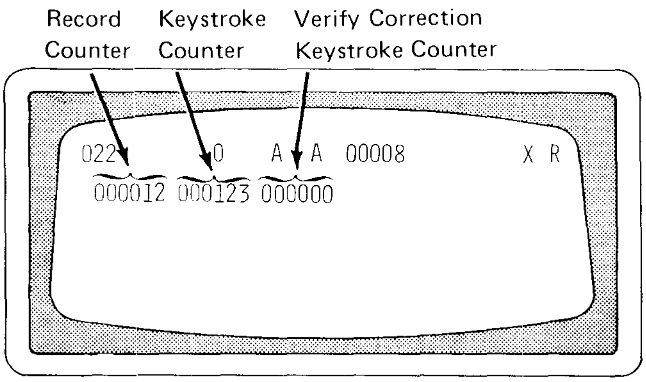
Note: If the operator backspaces and rekeys previous records, these records are counted again. Also, when a record is deleted the counter is not decremented. Therefore, the record count after keying a batch can exceed the number of records in the batch.

Verify Correction Keystroke Counter

The verify correction keystroke counter counts the number of characters corrected during verify mode or entered during field correct mode except when a sign is changed on a numeric right adjust field or when a dash or ampersand is used to override a self-check error. The verify correction keystroke counter is a six-position counter that counts up to a maximum of 065,535. When the counter reaches 065,535, it is reset to 000,000. No overflow condition is indicated.

DISPLAYING PRODUCTION STATISTICS

The production statistic counters are displayed in the first 21 positions with a blank between each counter. See Figure 15. The record count is displayed in positions 2-7, the keystroke count in positions 9-14, and the verify correction count in positions 16-21.



```
022     0   A A  00008        X R
000012 000123 000000
```

Figure 15. Production Statistic Counters Display

The production statistics counters can be displayed in read index mode or displayed and then written on disk in enter mode.

To display production statistics counters in read index mode, press FUNCT SEL lower and DISPLAY PROD STAT. After the counters are displayed, the data display can be restored by:

- Pressing RESET, which causes the data set label to be displayed. The production statistics counters are not reset to zero.

- Pressing R, which causes the data set label to be displayed and resets the counters to zero.

To display production statistics counters in enter mode, the following conditions must exist:

- Program 0 must be selected.

- The record length must be 21 positions or greater.

- No data has been entered into the current record buffer.

- No leading automatic duplicate/skip fields have been executed.

After the production statistics counters are displayed in enter mode, they can be written on disk by pressing REC ADV. Also, when the counters are displayed in enter mode, the cursor is positioned at position 22. This positioning allows additional data to be keyed beyond position 22, up to the record length prior to pressing record advance. The record is written on disk and the disk address is incremented by one when the REC ADV is pressed. The production statistic counters are not reset to zero in enter mode. See the *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136, for a complete description on how to display production statistics counters in read index or enter mode.

TRACK TO RECORD CONVERSION TABLE

Figure 16 is a track to record conversion table. It is provided as a quick method for determining the number of records keyed. The conversion table is an accurate alternative to production statistics for obtaining a record count. The conversion table could also be used when a grand total is being maintained in the production statistics record counter and totals for individual batches are desired. The number of records keyed is shown for each track on the disk.

Note: The conversion table assumes there are no deleted records on a track. If there are deleted records, subtract the number of deleted records from the total.

The EOD address in the data set label can also be used to determine the number of records in a data set. However, remember that the EOD address is one more than the actual number of records in the data set.

Track	Number of Records
01	Number of sectors used on this track
02	26*
03	52*
04	78*
05	104*
06	130*
07	156*
08	182*
09	208*
10	234*
11	260*
12	286*
13	312*
14	338*
15	364*
16	390*
17	416*
18	442*
19	468*
20	494*
21	520*
22	546*
23	572*
24	598*
25	624*
26	650*
27	676*
28	702*
29	728*
30	754*
31	780*
32	806*
33	832*
34	858*
35	884*
36	910*
37	936*
38	962*
39	988*
40	1014*
41	1040*
42	1066*
43	1092*
44	1118*
45	1144*
46	1170*
47	1196*
48	1222*
49	1248*
50	1274*
51	1300*
52	1326*
53	1352*
54	1378*
55	1404*
56	1430*
57	1456*
58	1482*
59	1508*
60	1534*
61	1560*
62	1586*
63	1612*
64	1638*
65	1664*
66	1690*
67	1716*
68	1742*
69	1768*
70	1794*
71	1820*
72	1846*
73	1872*

*Add the number of sectors used on this track to this number to get the total number of records.

Figure 16. Track to Record Conversion

Field totals can be used for batch auditing, to obtain what is sometimes referred to as hash totals, or for other applications where totals are required, such as summing the prices of individual items on an invoice to obtain the total order cost.

The offline field totals feature is used to obtain totals from records that have already been entered, updated, or verified. The offline field totals feature uses three 19-digit accumulators to add, under program control, data in specified fields into the assigned accumulator. A field total may include any number, 0 through 9, and any one of the 256 EBCDIC characters. All codes with low-order four bits equal to 0 through 9 are added with a value of 0 through 9. All other codes have the value of 0. All fields having units position codes with the high-order four bits equal to a B or D are negative fields.

The offline field totals feature allows you to:

- Accumulate totals.
- Read field totals from an accumulator into a record without resetting the accumulator.
- Read field totals from an accumulator into a record and reset the accumulator to zero.

OFFLINE FIELD TOTALS OPERATION

The offline field totals feature can be selected from the read index or update mode. When this feature is selected, the mode changes to field totals and an F is displayed in the status line. Selecting field totals mode on either station of the 3742 prevents the use of the other station during the field totals operation. See the *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136, for information on field totals operating procedures.

Offline field totals require programs and control statements. Programs define what field totals operations are to be performed and control statements identify which records are to be processed and which program is to be used to process each record. A description of how to prepare programs and control statements for field totals operations is provided later in this chapter.

When the field totals operation is initiated, the accumulators are reset to zero. Records are processed under control of a program specified by a control statement that was satisfied. Field definition characters, except those followed by a number, are ignored. Field totals fields are identified by following a field definition character with a number (1 through 9) which specifies which of the accumulators are involved in the field totals operation.

The low-order 14 digits of a field totals field are added into the specified accumulators. Deleted records are bypassed in a field totals operation. Disk errors and invalid control statement errors stop the field totals operation.

If AUTO REC ADV is turned off during a field totals operation, the field totals operation stops, and the record following the last record processed is displayed. Press REC ADV to process the record displayed and advance to the next record. Pressing RESET during a field totals operation changes the mode to update mode and terminates the field totals operation.

At the end of a field total operation, the accumulators retain the accumulated field totals unless a read out and reset was the last operation performed. If the end-of-data address is reached, the data set label is displayed, and the mode changes to read index mode.

Displaying Offline Field Total Accumulators

The contents of the field total accumulators and their sign can be displayed any time during the field totals operation. For information on how to display the accumulators during a field totals operation, see the *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136. Figure 17 shows the field totals display.

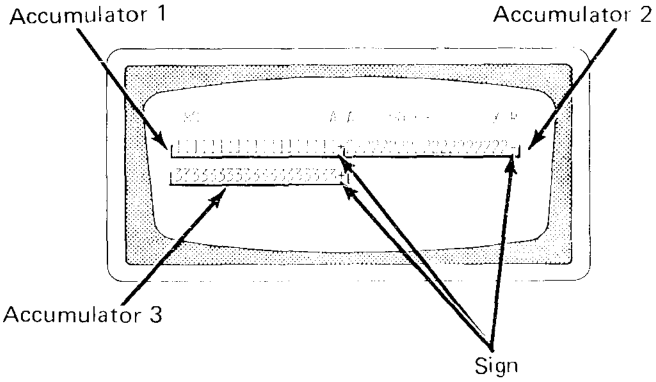


Figure 17. Field Totals Display

PROGRAMMING FIELD TOTAL FIELDS

The following describes how to program the field totals fields when you are writing programs to be used by the operator.

Accumulate Totals Fields

To indicate which fields in a data record are to be accumulated into a field totals accumulator, use any field definition character in a program followed by a 1, 2, or 3. A field definition character followed by a 1 means the contents of that field in the data records will be accumulated into accumulator 1. A field definition character followed by a 2 or 3 means the contents of that field in the data records will be accumulated into accumulator 2 or 3, respectively. Figure 18 is an example of a program containing field totals fields that will be accumulated.

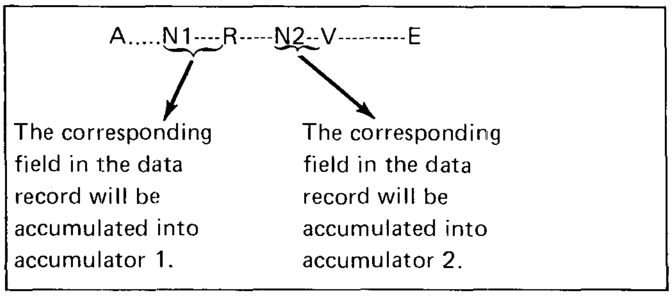

A.....N1----R-----N2--V----------E
The corresponding field in the data record will be accumulated into accumulator 1.
The corresponding field in the data record will be accumulated into accumulator 2.

Figure 18. Example of a Program Containing Field Totals Fields.

A field may be accumulated into more than one field totals accumulator by using any field definition character followed by more than one number. For example, if a program contains a N12--- field, the corresponding field in the data record is accumulated in accumulators 1 and 2.

Read Out Field Totals Fields

The field totals accumulations can be read out and placed into a data record without resetting the accumulators to zero. To indicate a read out field in a program, use the bypass field definition character, B, followed by a 4, 5, or 6. The 4 means read out accumulator 1 into the field in the record, 5 means read out accumulator 2 into the field in the record, and 6 means read out accumulator 3 into the field in the record. Figure 19 shows an example of a read out field in a program.

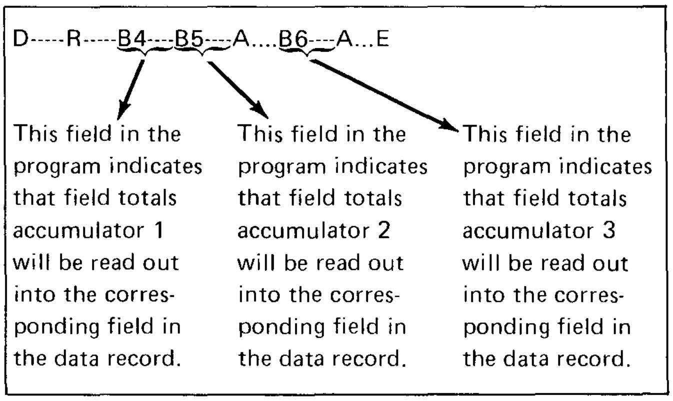

D-----R-----B4----B5----A....B6----A...E
This field in the program indicates that field totals accumulator 1 will be read out into the corresponding field in the data record.
This field in the program indicates that field totals accumulator 2 will be read out into the corresponding field in the data record.
This field in the program indicates that field totals accumulator 3 will be read out into the corresponding field in the data record.

Figure 19. Example of a Read Out Field in a Program

Not all of the accumulators have to be read out when performing a field totals read out operation. You can select the accumulators to be read out. If desired, each accumulator can be read out at a different point during a field totals operation.

Read Out/Reset Field Totals Fields

Read out/reset field totals fields allow the accumulators to be read into a specified field in a data record and then the accumulators are reset to zero. To indicate a read out/reset field in a program, use the bypass field definition character, B, followed by a 7, 8, or 9. The 7 means read out and reset accumulator 1, the 8 means read out and reset accumulator 2, and the 9 means read out and reset accumulator 3. Figure 20 shows an example of a read out/reset field in a program.

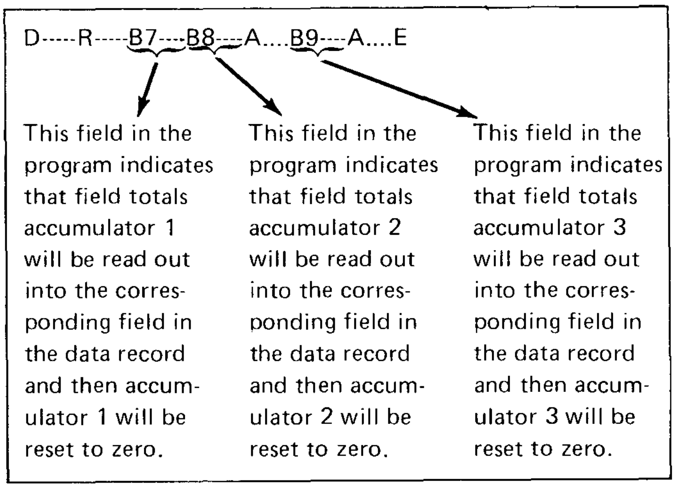

D-----R-----B7----B8----A....B9----A....E
This field in the program indicates that field totals accumulator 1 will be read out into the corresponding field in the data record and then accumulator 1 will be reset to zero.
This field in the program indicates that field totals accumulator 2 will be read out into the corresponding field in the data record and then accumulator 2 will be reset to zero.
This field in the program indicates that field totals accumulator 3 will be read out into the corresponding field in the data record and then accumulator 3 will be reset to zero.

Figure 20. Example of Read Out/Reset Fields in a Program

Not all of the accumulators have to be read out and reset when performing a field totals read out/reset operation. You can select accumulators to be read out and reset. If desired, each accumulator can be read out and reset at a different point during a field totals operation.

CONTROL STATEMENTS

Offline field totals require a control statement as part of the field totals operation. A control statement, when satisfied, causes a particular program to be selected for processing the record. A control statement:

- Must start with a < character (this is the only means by which the field totals program can identify a control statement).
- May be continued in a consecutive or higher numbered buffer.
- Identifies which positions in the current record are to be compared against the compare characters.
- Identifies what character must be at a position being compared, what character should not be at a position being compared, or some combination of both conditions in order for the control statement to be satisfied.
- Must not be loaded into any program buffer that will be selected when a control statement is satisfied.
- Identifies the program to use when the control statement is satisfied. The selected program identifies:
  - — the fields in the current record that are to be accumulated and into which accumulated the contents of each field must be accumulated.
  - — the fields in the current record that an accumulated value will be read into (if indicated by the program).
  - — the fields in the current record that an accumulated value will be read into and then the accumulator reset to zero (if indicated by the program).

Figure 21 identifies the control statement format characters.

Characters	Meanings
f	The **f** stands for any character 1 through 9 or A on the 3742 with the 128 feature or any character 1 through 6 on the 3742. The character indicates which program is to be selected if this control statement is satisfied.
ccc	**ccc** stands for a 3-digit position number indicating the position in the data record that is to be compared with the compare character(s): (001 - 080) 3742 (001 - 128) 3742 with 128 feature
W or N	W indicates that the position in the record must contain the character used in the control statement in order for the search to be satisfied. N indicates that the record position must not have the compare character.
m	**m** stands for the character with which the record position is to be compared. A comma (,) must separate compare characters.
,	The comma (,) is used to separate compare characters (**m**).
&	The & is used in a control statement to perform logical AND functions. The AND function allows you to indicate that two or more positions in the record must match the compare characters before the record is processed.
;	More than one control statement can be in a program buffer. The semi-colon (;) must be used to separate control statements.
:	Field total control statements can be continued to a higher numbered buffer. To indicate that a control statement is continued in a higher numbered buffer, use the colon (:) after the last position of the control statement in the current program buffer.
.	The period (.) indicates the end of the control statements. Place the period after the last control statement.
<	This character identifies the program as a field totals control statement and must be in the first position of every program buffer used for control statements.

Figure 21. Control Statement Format Characters

Control Statement Format Character Description

The following paragraphs will discuss each of the control statement characters and also build an example of field totals control statements.

Note: In the following descriptions, the lower case characters are a representation of entries you must make. The upper case characters are the exact character to be used.

< (Less Than Sign)

The less than sign (<) identifies a statement in a program buffer as being a field totals control statement. The less than sign must be in the first position of *every* program buffer that contains a field totals control statement.

```
<
```

f *(Program Buffer to be Selected)*

The **f** stands for the program buffer number that is to be selected if the control statement is satisfied. For the 3742, **f** can be 1 through 6, or 1 through 9 and A for a 3742 with the 128 feature.

```
<f
```

ccc *(Three-digit Position Number)*

The **ccc** stands for a three-digit position number that indicates the position in the data record that is compared with the compare character(s). Valid entries for the **ccc** field are: 001 through 080 for the 3742 and 001 through 128 for the 3742 with the 128 feature.

```
< fccc
```

W or N Field

This field is used to indicate whether a character should be present or absent in the position specified by **ccc**. The W indicates that the position in the record specified by **ccc** must contain the character **m** specified in the control statement in order for the control statement to be satisfied. The N indicates that the position in the record specified by **ccc** must not contain the character **m** specified in the control statement, in order to satisfy the control statement. The W or N must follow the position number **ccc**.

```
< fcccW
```

```
< fcccN
```

m *Field*

The **m** field stands for the character with which the record position specified by **ccc** is to be compared. The **m** can be any character A through Z, 0 through 9, or a special character. The character, **m**, is called the compare character. The compare character must follow the W or N field.

```
< fcccWm
```

```
< fcccNm
```

A position in a record can be compared against more than one character. However, each compare character (**m**) in the control statement must be separated by a comma (,).

```
< fcccWm, m, m
```

The position being checked must have one of the characters specified by **m** in order to satisfy the control statement.

```
< fcccNm, m, m
```

The position being checked must not have any of the characters specified by **m**, in order to satisfy the control statement.

(period) – End of Control Statement

The period (.) indicates the end of the control statement. If there is more than one control statement, the period must be at the end of the last statement.

```
< fcccWm.
```

```
< fcccNm.
```

; (semicolon) – Multiple Control Statement Separator

The semicolon is used when there is more than one control statement in a program buffer. The semicolon must follow each control statement except the last control statement which must have a period (.). When there are multiple control statements, each control statement is compared against the current data record until one of the control statements is satisfied. If none are satisfied, the next record is read.

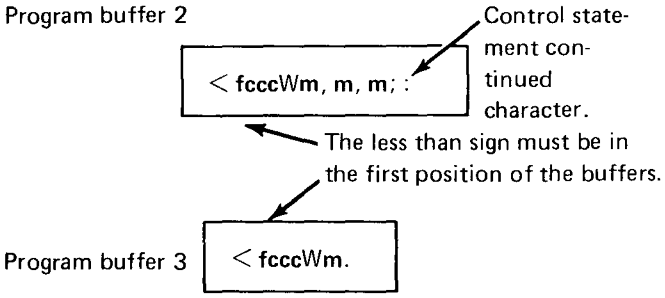
```
< fcccWm; fcccNm.
```

```
< fcccWm, m, m; fcccWm; fcccNm, m.
```

Note: When coding multiple control statements, place the control statement that will be satisfied most often first. This will save time in running field totals operations.

: (colon) – Control Statement Continued in a Higher Numbered Buffer

Control statements can be continued into another higher numbered buffer. When a control statement is continued in a higher numbered buffer, place a colon (:) after the last character in the current program buffer. The colon (:) can be placed after any character desired.


Program buffer 2

< fcccWm, m, m; :

Control statement continued character.

The less than sign must be in the first position of the buffers.

Program buffer 3

< fcccWm.

When a control statement will be continued in a higher numbered buffer, it does not have to be in the next consecutive higher numbered buffer. The statement has only to be continued in a higher numbered buffer.

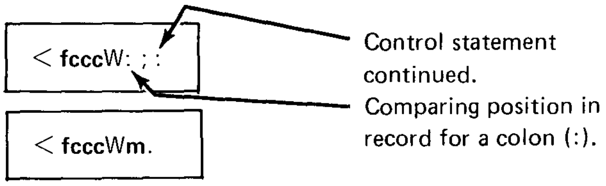
Program buffer 3

```
< fcccWm, m, m; :
```

Program buffer 6

```
< fcccWm.
```

Note: If the compare character **m** is a colon (:), it is treated as a compare character. Therefore, if **m** is a colon (:) and the control statement will be continued in a higher numbered buffer, another colon is needed.


< fcccW: ; :

< fcccWm.

Control statement continued.

Comparing position in record for a colon (:).

& (ampersand) Logical AND

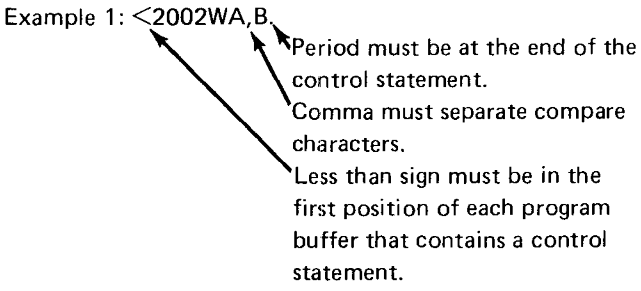
The & (ampersand) can be used in a control statement to perform a logical AND function. This means that more than one position in a record can be compared against the control statement.

```
< fcccWm&cccWm.
```

Example: <1010WA&020W4. This control statement is satisfied if position 10 of the record has an A *AND* position 20 has a 4. Program buffer 1 will be selected to process the record if the control statement is satisfied.

Examples

The following are examples using all of the control statement format characters that are available.


Example 1: <2002WA,B.

Period must be at the end of the control statement.

Comma must separate compare characters.

Less than sign must be in the first position of each program buffer that contains a control statement.

This control statement will be satisfied if the current data record has an A or B in position 2. Program buffer 2 will be selected if the statement is satisfied.

```
Example 2: <2002WA,B;1010NV.
                    Control statement separator.
```

This is a multiple control statement. The field totals program will first check to see if the data record has an A or B in position 2. If there is an A or B in position 2, the control statement is satisfied and program buffer 2 will be selected to process the record. If the first control statement 2002WA,B is not satisfied, the field totals program goes to the next control statement (1010NV) to see if it can be satisfied. This control statement is satisfied if position 10 of the data record does not have a V. Program buffer 1 is selected for processing the data record if this statement is satisfied. If neither statement is satisfied, the next data record is read.

Example 3:

```
                              Logical AND function.
Program buffer 3:   <2002WA,B&031WH;4042;
                                        Control
                                        statement
                                        continued
                                        in higher
                                        numbered
                                        buffer (5).
                           Control statement separator.
Program buffer 5:   <W6,7,8.
```

This is a multiple control statement that is continued in a higher numbered buffer. The first control statement (2002WA,B&031WH) contains a logical AND function. The control statement 2002WA,B&031WH is satisfied if position 2 of the data record has an A or B and position 31 has an H. Program buffer 2 is selected to process the data record if the statement is satisfied. If this control statement is not satisfied, the next control statement (4042W6,7,8) will be used. Note that this control statement is continued in a higher numbered buffer. This control statement is satisfied if position 42 of the data record has a 6, 7, or 8. Program buffer 4 is selected for processing the data record if the statement is satisfied. If neither statement is satisfied, the next data record is read.

Unconditional Program Buffer Selection

If only one program is required to process records in the field totals operation, a control statement can be written that will unconditionally select the desired program buffer. The general control statement is as follows:

```
<f.
```

where f is the desired program buffer number 1 through 6 for the 3742 or 1 through 9 and A for the 3742 with the 128 feature.

Offline Field Totals Example

An offline field totals operation is to be performed on a data set that contains several different kinds of records. The field totals operation to be performed depends upon the content of the records.

Figure 22 shows the field totals operation to be performed for each record type. If the record type indicated by the first column in the figure contains the data indicated by the second column, the operation in the third column needs to be performed.

Depending upon the type of record, one of four different types of field total operations are performed:

1. Add a field to both accumulators 1 and 2.

2. Add the field to accumulator 2 only.

3. Read the contents of accumulator 1 into the current record buffer and reset the accumulator.

4. Add the field to accumulator 3.

Four different programs are required to do these operations. This example used program buffers 1-4 for these programs. Program buffer 1 contains the program to add the field in positions 30-35 of the data record into accumulators 1 and 2. To program a field as a field totals field, use any field definition character followed by 1, 2, or 3 to indicate the accumulator. All field definition characters in the program except those used by field totals fields are ignored.

Type of Data Record	Contents	Desired Operation
1	A in position 10 and B or C in position 25	Add the field in positions 30-35 of the record to accumulators 1 and 2.
2	Numbers in position 50	Add the field in positions 30-35 of the record to accumulator 2.
3	Blanks in positions 50 through 55	Read the contents of accumulator 1 into the record in positions 1-10 and reset accumulator 1.
4	All records that do not satisfy the conditions of the other 3 types of records	Add the field in positions 30-35 of the record to accumulator 3.

Figure 22. Field Total Operations

In this example all fields except the field totals field are programmed as bypass fields. The program in program buffer 1 is:

$$\texttt{B----------------------------}\underbrace{\texttt{N12}}\texttt{---E}$$

Field totals field
in positions 30-35.

Program buffer 2 contains the program to add the field in positions 30-35 of the data record into accumulator 2. The program in program buffer 2 is:

$$\texttt{B----------------------------}\underbrace{\texttt{N2}}\texttt{----E}$$

Field totals field
in positions 30-35.

Program buffer 3 contains the program for the record that accumulator 1 is to be read into. The accumulator is read into positions 1 through 10 of the record. The program also causes accumulator 1 to be reset to zero. The program in program buffer 3 is:

$$\underbrace{\texttt{B7}}\texttt{--------E}$$

Field that accumulator 1 is read into.

Program buffer 4 contains the program to add the field in positions 30-35 of the record in accumulator 3 if the conditions to use the programs in program buffers 1 through 3 are not satisfied. The program in program buffer 4 is:

$$\texttt{B----------------------------}\underbrace{\texttt{N3}}\texttt{----E}$$

Field totals field
in positions 30-35.

To cause the program that is to process each record type to be selected automatically, write control statements. Each control statement indicates the conditions that must be satisfied before a program is selected. The control statements for this example begin in program buffer 5. Each control statement is separated by a semicolon. The colon is used at the end of program buffers 5 and 6 to indicate that the control statements are continued in a higher program buffer. The control statements follow:

Program Buffer 5

$$\underbrace{\texttt{<1010WA\&025WB,C}}\qquad\underbrace{\texttt{;2050W0,1,2,3:}}$$

Selects program buffer 1 if position 10 contains an A and position 25 contains a B or a C.

Selects program buffer 2 if position 50 is numeric. This control statement is continued in program buffer 6.

Program Buffer 6

<,4,5,6,7,8,9;3050Wƀ&051Wƀ&052Wƀ&053Wƀ&:

Continued from program buffer 5.

Selects program buffer 3 if positions 50-55 are blank. (The ƀ represents a blank space.) This control statement is continued in program buffer 7.

Program Buffer 7

<054Wƀ&055Wƀ;4.

Continued from program buffer 6.

Selects program buffer 4 if another program buffer has not been selected.

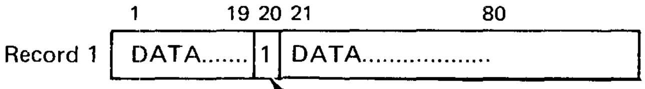
Offline Field Totals Setup Example

The following is an example of an easy way to write control statements when different programs are used for entering data.

When writing programs required for a job that has field totals fields, set aside one position in the program for keying the program number used when keying data for the record. This one position field will assist you in writing the offline field totals control statements required to process records in the data set.

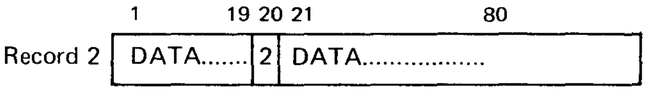
For example, assume programs with field totals fields have been written and are in program buffers 1, 2, and 3. Also, assume that position 20 in the program represents the position in the data record into which the program number being used to control the format of the record will be keyed. Also assume that enter mode has been selected and data can now be keyed.

The first data record requires program number 1.



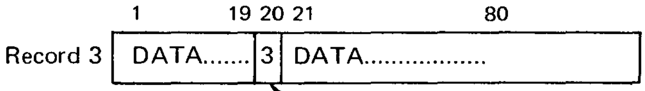
A 1 is keyed into this position (20) to indicate that program number 1 was used to control the format of this data record.

The second data record requires program number 2.



A 2 is keyed into this position (20) to indicate that program number 2 was used to control the format of this data record.

The third record requires program number 3.



A 3 is keyed into this position (20) to indicate that program number 3 was used to control the format of this data record.

This process continues until all data records have been keyed. Position 20 in the data record identifies the program that was used to control the format of each record.

After all data has been keyed, assume that an offline field totals operation will be performed. Since each record in the data set has a position 20 that identifies the program under which data was keyed for the record, this information can be used in writing control statements required to process the job in offline field totals mode.

The control statement would be:

< 1020W1; 2020W2; 3020W3.

This control statement will select program number 1 if position 20 in the data record has a one (position 20 will have a one if program number 1 was used to control the format of the record in enter mode).

This control statement will select program number 2 if position 20 in the data record has a two (position 20 will have a two if program number 2 was used to control the format of the record in enter mode).

This control statement will select program number 3 if position 20 in the data record has a three (position 20 will have a three if program number 3 was used to control the format of the record in enter mode).

If the preceding technique can be implemented, it will assist you in using offline field totals mode.

Chapter 6. Second Disk

On the 3742, dual disk capability is obtained on the primary station by disabling the secondary station and controlling the secondary stations disk from the primary station (the primary station has the power on switch).

The purpose of the second disk operations is to permit the following:

1. Duplicating a disk in order to have more than one copy or to save data from a disk that has developed a defect.

2. Merging records from a disk with records from the keyboard.

3. Pooling the data from several disks onto one disk.

DUAL DISK PREPARATION ON 3742

For dual disk preparation, always do the following:

1. Make sure that a disk is loaded in the secondary station and that the secondary station is in the read index mode.

2. On the secondary station, press FUNCT SEL lower and then press A. An A will appear in the status line in position 36 and the keyboard will be inactive. Disk 2 operations on the primary station now control this disk.

The 3742 is returned to normal two-station operation by pressing FUNCT SEL lower on the primary station followed by the A character key.

DUAL DISK OPERATION ON THE 3742

After inserting a disk in the secondary station's disk drive, the first of any of the following operations activates disk 2, positions disk 2 at track 00, sector 08, and displays the sector 08 label.

- Disk 2 record advance
- Disk 2 record backspace
- Disk 2 return to index

The address of disk 2 is displayed at the righthand end of the status line in positions 31 through 35. The address is posted or changed only when disk 2 operations are performed. Insertion and removal of disk 2 are not monitored by the machine and therefore inserting a disk does not post a disk 2 address or remove an existing disk 2 address; likewise, removing disk 2 does not cause the existing disk 2 address to be removed from the display. Keying FUNCT SEL lower and the A key will remove the existing disk 2 address.

Disk 2 is read but not written on. Disk 2 is controlled by the following keys:

- DISK 2 REC ADV
- DISK 2 REC BKSP
- DISK 2 RET TO INDEX
- COPY
- SEARCH CONTENT
- SEARCH SEQ CONTENT
- SEARCH ADDRESS

SEARCH OPERATIONS ON THE SECOND DISK

The following procedures are used to search on disk 2:

1. Press FUNCT SEL lower.

2. Hold down NUM SHIFT and press SEARCH CONTENT, SEARCH SEQ CONTENT, or SEARCH ADDRESS.

3. Press FUNCT SEL lower and DELETE REC to clear the display and key the search mask or address.

4. Press FUNCT SEL upper and DISK 2 REC ADV.

After the first content search, subsequent searches by content using the same search mask can be accomplished as follows:

1. Press FUNCT SEL lower, hold down NUM SHIFT, and press SEARCH CONTENT.

2. Press FUNCT SEL upper and DISK 2 REC ADV.

COPY

The copy function provides:

- Copying all or part of disk 2
- Copying all or part of a data set
- Copying up to a specified record on disk 2
- Copying a record

Copying is always done from disk 2 to disk 1.

Copy All (Image Copy) or Part of Disk 2

This copy function copies disk 2 onto disk 1, beginning with the sector in track 00 where disk 1 is positioned. The last record is copied from the sector located at the end of extent (EOE) of the label at which disk 2 is positioned at the start of the copy function.

Each record is copied from disk 2 into the corresponding track and sector address of disk 1; that is, the record from track 02, sector 14 of disk 2 will be copied into track 02, sector 14 of disk 1. All records will be copied regardless of content, including deleted records, records outside of defined extents, and normal data records between the beginning of extent (BOE) and the end of data (EOD).

During the Copy All or Part of Disk 2 operation:

- On 3742s without the 128 feature, program buffers 1 through 6 on the primary station and 1 through 3 on the secondary station are destroyed. If you are using these program buffers during data entry, they must be restored after the copy is complete.

- The write protect status is not checked and does not protect against copying onto disk.

For operating procedures on copying all or part of disk 2, see the *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136.

Copy All (Image Copy) or Part of a Data Set

The primary prupose of this function is to pool data from two or more disks onto one disk by adding data from disk 2 to the end of the data already recorded on disk 1. A sequence of such copies can involve loading several disks into the disk 2 drive and pooling the work of several operations onto one disk. Deleted records are not copied.

For operating procedures on copying all or part of a data set, see *IBM 3742 Dual Data Station Operator's Guide,* GA21-9136.

Copy Up to a Specified Record on Disk 2

This function can be used to merge records from disk 2 with records that are keyed from the keyboard. It can be used to insert missing records or add records to a data set. All records except deleted records are copied up to, but not including, the specified record, At the end of the copy function, the specified record is displayed.

For operating procedures on copying up to a specified record on disk 2, see *IBM 3742 Dual Data Station Operator's Guide,* GA21-8136.

Copying a Record

This function allows you to copy one record at a time by performing a search disk 2 operation, or by pressing DISK 2 REC BKSP or DISK 2 REC ADV, whichever is necessary to find the record you want to copy. The record to be copied is on disk 2. This copy function can be done in enter, update, or read index mode. If you are in enter mode after the record from disk 2 has been displayed, the record can be entered onto disk 1 by pressing REC ADV. If you are in update mode after a record from disk 2 has been displayed, the first character of the record must be rekeyed and then REC ADV pressed. If you are in read index mode, press FUNCT SEL lower, M, and REC ADV to enter the record onto disk 1.

Chapter 7. Disk Initialization

All disks are initialized before they are shipped to a customer. Reinitializing should be avoided unless necessary; it is required only if:

1. The disk was exposed to a strong magnetic field.

2. A defect occurred in one or two tracks. In this case, initialization can be used to take the bad track(s) out of service and add one or two tracks from the spare area.

3. A sector sequence other than the sequence existing on the disk is needed.

See *Initialization Procedures* in this chapter for information on how to specify the desired sector sequence.

The purpose of initialization is:

1. To write identification fields, one for each sector in all active tracks on a disk.

2. To write the bad track code in defective tracks. The maximum permissible number of bad tracks is two.

3. To write a record in each sector of each active track. All records are filled with the blank character except in track 00.

4. To write 80-character records in track 00.

After the index track is written, sectors 1 through 4 and 6 contain 80 blanks each. Sector 5 contains ERMAP followed by 75 blank characters. If one or two bad tracks were specified, the number of the first bad track will be positions 7 and 8 of sector 5 with a zero in position 9, the number of the second bad track will be in positions 11 and 12 of sector 5 with a zero in position 13. Sector 7 contains VOL1 in positions 1 through 4, the volume ID in positions 5 through 10, the sector sequence code is in positions 77 and 78, a W in position 80, and blanks in all remaining positions of the first 80.

Sector 8 contains the following data set labels:

- HDR1 in positions 1-4
- DATA in positions 6-9
- ƀ80 or 128 in positions 25-27 depending on the input specifications
- 01001 in positions 29-33
- 73026 in positions 35-39
- 01001 in positions 75-79
- Blanks in all other positions

Sectors 9 through 26 contain deleted records with the following content:

- DDR1 in positions 1-4
- DATA in positions 6-9
- Sector number in positions 10-11
- ƀ80 or 128 in positions 25-27
- 74001 in positions 29-33
- 73026 in positions 35-39
- 74001 in positions 75-79
- Blanks in all other positions

As a final step, disk initialization checks the disk to see that it is written correctly.

INITIALIZATION PROCEDURE

On the 3742, be sure the secondary station's disk drive does not contain a diskette.

To initialize the disk, do the following:

1. Insert a diskette in the primary station drive.
   Note: If the disk is bad or uninitialized, several disk errors may occur before X R mode is displayed.

2. Press FUNCT SEL lower, then DELETE REC.

3. Key the following data (each of these fields is optional and can be left blank):

   - Volume ID in position 1-6.

   - Sector sequence code in positions 7 and 8 (normally left blank). Valid entries, in addition to the positions being left blank, are shown in Figure 23. Most commonly used are 01 (same as being left blank) and 02. Entering 01 causes the sector arrangement to be 1, 2, 3, 4 . . . 25, 26. Entering 02 causes the sector arrangement to be 1, 3, 5, . . . 25, 2, 4, . . . 24, 26.

   - First bad track in positions 9 and 10.

   - Second bad track in positions 11 and 12.

   - Blank in positions 13 and 14.

   - Blank for record length of 80 or L for record length of 128 in position 15.

4. Press FUNCT SEL lower, then hold down NUM SHIFT and press the I key.

   Note: If the numeric shift key is not held down, only the disk checking function occurs. See *Disk Checking* in this chapter.

Blank	01	02	03	04	05	06	07	08	09	10	11	12	13
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	3	4	5	6	7	8	9	10	11	12	13	14
3	3	5	7	9	11	13	15	17	19	21	23	25	2
4	4	7	10	13	16	19	22	25	2	2	2	2	15
5	5	9	13	17	21	25	2	2	11	12	13	14	3
6	6	11	16	21	26	2	9	10	20	22	24	26	16
7	7	13	19	25	2	8	16	18	3	3	3	3	4
8	8	15	22	2	7	14	23	26	12	13	14	15	17
9	9	17	25	6	12	20	3	3	21	23	25	4	5
10	10	19	2	10	17	26	10	11	4	4	4	16	18
11	11	21	5	14	22	3	17	19	13	14	15	5	6
12	12	23	8	18	3	9	24	4	22	24	26	17	19
13	13	25	11	22	8	15	4	12	5	5	5	6	7
14	14	2	14	26	13	21	11	20	14	15	16	18	20
15	15	4	17	3	18	4	18	5	23	25	6	7	8
16	16	6	20	7	23	10	25	13	6	6	17	19	21
17	17	8	23	11	4	16	5	21	15	16	7	8	9
18	18	10	26	15	9	22	12	6	24	26	18	20	22
19	19	12	3	19	14	5	19	14	7	7	8	9	10
20	20	14	6	23	19	11	26	22	16	17	19	21	23
21	21	16	9	4	24	17	6	7	25	8	9	10	11
22	22	18	12	8	5	23	13	15	8	18	20	22	24
23	23	20	15	12	10	6	20	23	17	9	10	11	12
24	24	22	18	16	15	12	7	8	26	19	21	23	25
25	25	24	21	20	20	18	14	16	9	10	11	12	13
26	26	26	24	24	25	24	21	24	18	20	22	24	26

• **Figure 23. Disk Record Sequences**

Any or all of the data in step 3 may be left blank. If the volume ID is left blank, the volume ID will be written as a blank field. If the sector sequence is left blank, the sequence of sectors will be in numeric order (1, 2, 3, ...25, 26). If no bad tracks are listed, the initialization operations will scan the disk to find any defective tracks marked with the bad track code. When one or two bad tracks are specified, initialization operations do not check for the bad track code and will try to initialize all other tracks.

When the initialization process is complete, the disk is positioned at track 00 and the label in sector 08 is displayed, unless errors were encountered. In case of errors, the following is displayed:

- Bad tracks that have been marked with the bad track code are displayed in the first data line. Positions 9 and 10 show the first bad track; 11 and 12 show the second bad track.

- Tracks that showed errors when checked are shown in data line 2 beginning in cursor position 41. Each track shown is written with three characters followed by a space. The first two characters indicate the track number and the third character indicates the type of error encountered:

I	=	Missing ID field
C	=	CRC error
S	=	Deleted record

Only 10 tracks with errors can be displayed; the first error encountered in a track is displayed.

If errors are displayed after initialization, the initialization was not successful. The disk should be removed to clear errors and initialization tried again until a successful operation occurs. If particular tracks show persistent errors in successive retries, those tracks should be considered defective. Initialization can be retried by specifying those tracks in the keyed data of step 3. Since the maximum number of bad tracks that can be specified is two, disks with persistent errors in more than two tracks cannot be used.

DISK CHECKING

The initialization feature provides a means of checking a disk for tracks marked with the bad track code and for tracks exhibiting errors. The procedure for checking is as follows:

1. Insert a diskette in the primary station drive. Make sure a diskette is not in the secondary drive.

2. Press FUNCT SEL lower and DELETE REC.

3. Press FUNCT SEL lower.

4. Press the I character.

At the end of the scan, a report is posted on the display in the same format as used for reporting errors after initializing. If errors are reported, the disk should be checked repeatedly and only tracks that show persistent errors in repeated operations should be considered bad.

Figure 24 shows the errors and the recovery action for disk initialization errors.

Code	Type	Causes	What You Do
'	Input specification	Track 00 or 78 to 99 were specified as a bad track. Sequence is greater than 13. A character other than 0-9 or blank was entered or an odd number of characters entered in positions 7-12.	Reset** and correct error.
=	Bad track specification	Bad tracks are not entered in ascending sequence.	Reset** and reenter in ascending sequence.
*	Bad tracks	More than two bad tracks are on the disk.	Reset** and retry disk initialization. If error persists, discard disk.
(	Write error	Disk error occurred while initializing the disk.	Reset** and retry initialization.
&	Sector not deleted	One of the normally deleted sectors (9-26) on the index track was not deleted. Sequence number keyed was 00.	Reset** and retry initialization.

**Initialization errors are not reset by the reset key. The disk is removed and reinserted; then the initialization setup must be done over.

Figure 24. Disk Initialization Errors

Chapter 8. Self-Check Feature

The self-check feature provides a method for checking the entry in numeric fields that contain a precomputed self-check digit. The self-check digit is the last digit in the field, and has been precomputed by using either of two algorithms, modulus 10 or modulus 11.

In the modulus 11 operation, basic numbers that require a check digit of 10 cannot be used as self-check numbers. The accounting system must be adjusted to eliminate such numbers from codes that are to be self-checked. If an operator uses a basic number requiring a check digit of 10, when punching or verifying check digits the machine indicates an error condition.

Self-check fields are specified in the program by the use of the following characters:

H – Modulus 10
C – Modulus 11
F – Modulus 10 skip
G – Modulus 11 skip
L – Modulus 10 duplicate
M – Modulus 11 duplicate

Fields defined by the characters F and G act as skip fields when the AUTO DUP/SKIP switch is on, and as manual self-check fields when the AUTO DUP/SKIP switch is off. Fields defined by the characters L and M act as duplicate fields when the AUTO DUP/SKIP switch is on and as manual self-check fields when it is off.

When duplicate skip and self-check are combined in a field, the self-check is performed after the automatic function. If a self-check error is found, turn off the AUTO DUP/SKIP switch to backspace, and correct the self-check number.

Blank fields check correctly in the enter mode, but cause a C error in verify or field correct mode. This allows a self-check field to be skipped in enter mode and entered later in verify mode.

Numerals 0 through 9 and all other EBCDIC characters with the values 0 through 9 in the low-order hex digit are totaled by the low-order hex digit value. All other characters are totaled as the value zero.

If self-check errors occur, three methods of error recovery are:

1. Press RESET to unlock the keyboard and remove the error condition. This permits entry of a new character or backspacing to correct a preceding character.

2. Press the dash key with either shift key to (a) remove error condition, (b) overscore the units position with the high order hex D, and (c) exit from the field.

3. Press NUM SHIFT and the & key to (a) remove error condition, (b) overscore the units position with the high order hex C, and (c) exit from the self-check field.

COMPUTING THE SELF-CHECK DIGIT

Modulus 10

Modulus 10 can be computed for any number from 2 to 127 digits long. To compute modulus 10, do the following:

1. Multiply the units position and every alternate position of the basic number, excluding the self-check digit, by 2.

2. Add the digits in the products to the digits in the basic number that were not multiplied.

3. Subtract the sum from the next higher number ending in zero.

The difference is the self-check digit.

For example:

Basic number	6	1	2	4	8
Units position and every alternate position of basic number	6		2		8
Multiply	12		4		16
Digits not multiplied		1		4	
Add	1+2+1+4+4+1+6=19				
Next higher number ending in 0				20	
Subtract				-19	
Self-check digit				1	

Modulus 11

To compute modulus 11, do the following:

1. Assign a weighting factor to each digit position of the basic number. These weighting factors are 2-7, 2-7, etc. starting with the units position of the number, excluding the self-check digit, and progressing toward the high-order digit.

2. Multiply each digit by its weighting factor.

3. Add the products.

4. Divide this sum by 11.

5. Subtract the remainder from 11.

The difference is the self-check digit.

For example:

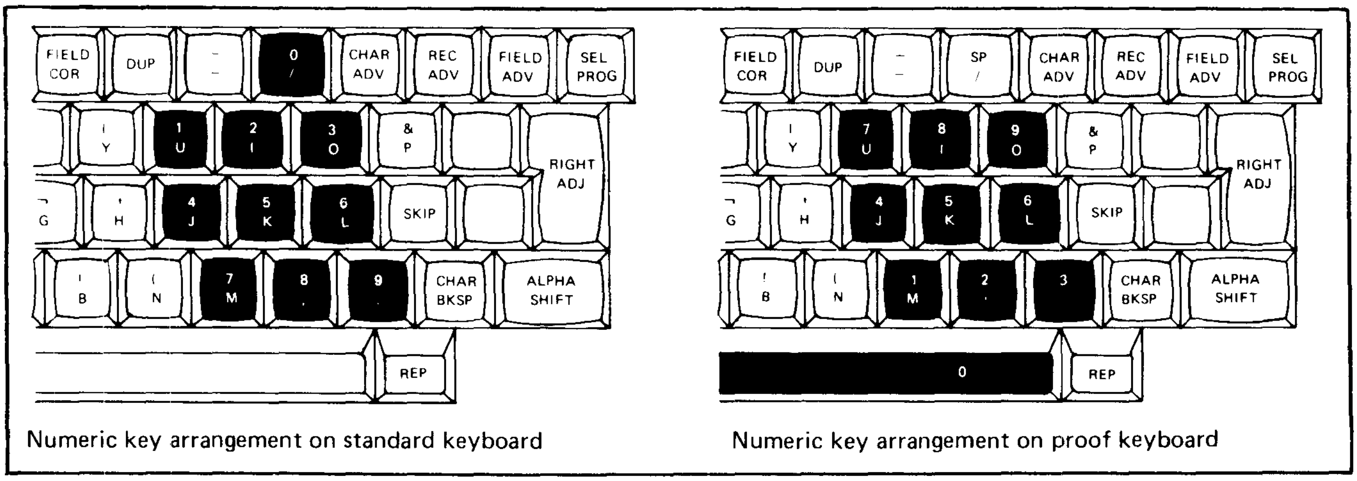
Basic number	1	2	3	4	5
Weighting factors	6	5	4	3	2
Multiply	6	10	12	12	10
Add	6+10+12+12+10=50				
Divide	50÷11=4 plus a remainder of 6				
Subtract	11-6=5				
Self-check digit	5				

Chapter 9. Proof Keyboard Feature

The 3742 proof keyboard differs from a standard 3742 keyboard only by a rearrangement of the numeric keys. The proof keyboard numeric key arrangement allows the operator, who is familiar with the adding machine numeric key arrangement, to enter data more effectively.

Figure 25 shows the numeric keys on a proof keyboard and the numeric keys on a standard keyboard. In alpha shift, both the standard keyboard and the proof keyboard operate the same.

Note: A 3742 can have either the proof keyboard numeric key arrangement or the standard keyboard numeric key arrangement, but both numeric key arrangements are not available on the same 3742.



Numeric key arrangement on standard keyboard

Numeric key arrangement on proof keyboard

Figure 25. Proof Keyboard and Standard Keyboard Numeric Key Arrangement

Appendix A. Timings

The following chart provides approximate timing information for search, copy, and initialization operations on the 3742. Timings are shown for the normal sector sequence (1, 2, 3, ...25, 26) and for the alternate sector sequence (1, 3, 5, . . .25, 2, 4, . . .24, 26). The timings provided in the chart should serve as a useful guide when relative job timing must be considered.

The timings are based on 80 character records. Timing can be affected by such things as disk speed, length and location of search masks, and the location of data sets on the disk. Also, search time will vary if the other keyboard is being operated during the search operation.

TIMING CHART

Operation	Number of Records	Normal Sector Sequence		Alternate Sector Sequence	
		Disk 1	Disk 2	Disk 1	Disk 2
Search end of data	1898 (Assume EOD is 74001)	7 sec	7 sec	7 sec	7 sec
	949 (assume EOD is 37014)	4 sec	4 sec	4 sec	4 sec
	400 (assume EOD is 16001)	2 sec	2 sec	2 sec	2 sec
Search on sequential content	1898	9 sec	9 sec	9 sec	9 sec
	949	7 sec	7 sec	7 sec	7 sec
	400	4 sec	4 sec	4 sec	4 sec
Search on content	1898	4 min	6 min	5 min	6 min
	949	2 min	3 min	3 min	3 min
	400	1 min	1 min 30 sec	1 min	1 min 30 sec
Disk Initialization	-------	2 min	------	2 min	------
Image copy	1898 plus index track	4 min	------	10 min	------
Data set copy	949	11 min	------	9 min	------
Data set copy	400	4 min	------	4 min	------

Appendix B. Glossary

accumulator: A counting device that retains an accumulated sum until reset.

alpha shift: The A and Z keys plus the bottom letters on all the other keys.

alphameric: Any of the 26 characters of the alphabet or the 10 numeric digits 0 through 9.

automatic field: A field that is completed by the machine automatically such as an auto dup or auto skip field which the operator does not have to key manually.

begin field code: Synonymous with field definition character.

blank character: An unused character position in a record. Results when the space bar is pressed.

BOE (beginning of extent): The beginning of a data set, defined by a track and sector number; pertains to positions 29 through 33 in the data set label.

bypass: Moving the cursor across a field or portion of a record without changing the content in the field or portion of the record.

continuation character: Synonymous with continuation field code.

continuation field code: Characters following the field definition characters in the program and indicate the program shift (alpha or numeric) for the remaining positions in the program.

control statement: Synonymous with mask statement.

copy: The duplication of data from one disk onto another disk.

counters: Devices that count keystrokes, records, and corrections (made while verifying). These counters are used for production statistics.

current field: The field being processed by the operator.

cursor: The underscore character (_) on the display screen that indicates where the next character will be entered (referred to as the cursor position).

data: Information that is keyed from a source document.

data set: The various types of information related to a particular job and entered as a group of records.

data set label: The information in track 00 (index track) that identifies a particular data set.

disk: The magnetic media on which data is recorded.

disk error: An error condition caused by the disk or machine. Disk error numbers are 1, 2, 3, 4, 5, 7.

diskette: The square plastic container and its enclosed disk.

duplicate fields: Fields programmed to repeat information from the same field in the previous record.

end programming code: Synonymous with end-of-program character.

end-of-program character: The letter (E) used to make the end of the program.

EOD (end of data): The address of the next available sector on the disk for a record. This address is in positions 75 through 79 of a data set label on the index track.

error code: A character displayed in the status line to indicate the type of error occurring.

field: A specific number of character positions reserved by means of a program for a particular category of data for a record(s). For example, the position in which the cost of a part is entered.

field definition character: The letter used in a program to mark the beginning of a data field. The letter also defines the shift and the type of field, such as R for a right adjust field, S for a skip field, etc.

fill character: Either a blank or zero. Used in right adjust fields, from the first position to the first non-fill character entered in the field.

index track: The first track on the disk (track 00).

load: To enter a program into a storage area.

machine mode: Pertains to what the machine is set up to do (that is, enter, verify) and is indicated in the status line.

machine status: Refers to the code in the status line that tells you whether the machine is ready (R) or not ready (N) or requires you to wait (W).

manual fields: Any field that the machine does not automatically process.

mask: Compare field for search operations.

mask statement: A set of instructions stored in the program area of the machine when doing field totals operations.

numeric shift: The top character on each key (except A and Z).

offline: An operation done with a previously keyed job when doing a field totals operation.

program: A set of instructions that tell the machine what to do with data as you key it.

program chaining: A method used for automatic program selection.

program code: The characters that make up a program.

program field: A field containing program code characters to specify the type of field, program shift, and length of the field.

program number: The number in the status line that gives the storage area number of the program currently being used.

program shift code: The letter in the status line that indicates the program shift (alpha or numeric) of the character to be keyed.

record: A piece of data no more than 80 characters long (128 characters with the 128 feature) that is usually only part of a related set of data.

record address: The track and sector number of a record.

right adjust: To enter characters at the right end of a field.

sector: A section in a track on the disk for a record.

skip field: A field in which data is blanked.

source document: Material containing the information to be keyed.

status line: The top line on the display screen. Gives information such as cursor position, record address, and machine mode.

verify bypass field: A field that is bypassed during verify mode but not during enter or update mode.

verify indicator field: A single character field indicating whether or not a data set is verified. The field contains a V if verified; it contains a blank if not verified.

verify mark: Synonymous with verify indicator field.

Index

IBM®

International Business Machines Corporation

General Systems Division
5775D Glenridge Drive N. E.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM 3742 Dual Data Station Reference Manual Printed in USA GA21-9184-3

READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). **No reply.**

Page Number *Error*

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number *Comment*

Note: All comments and suggestions become the property of IBM.

- No postage necessary if mailed in the U.S.A.

Name ____

Address ____

____

IBM 3742
Dual Data Station
Reference Manual

GA21-9184-3

GA21-9184-3

Cut Along Line

Fold

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM Corporation
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesota 55901

Fold

Fold

IBM®

International Business Machines Corporation

General Systems Division
5775D Glenridge Drive N. E.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM 3742 Dual Data Station Reference Manual Printed in USA GA21-9184-3