SB21-0621-0

Field Developed Program

3270 Format Macro For Output Data Streams

Program Number: 5798-AKL

Program Description/
Operations Manual

This manual describes the capabilities of the system and the programs. Discussion of design assumptions and potential modification areas are included. Record and file layouts are described and primary processing procedures specified. This manual is both a system description and an installation and operations reference document.

IBM

OFFERING PERIOD SERVICES

During a specified number of months immediately following initial availability of each licensed program, designated as the Offering Period, the customer may submit documentation to a designated IBM location when he encounters a problem which his diagnosis indicates is caused by a licensed program error. During this period only, IBM through the program author(s) will, without additional charge, respond to an error in the current unaltered release of the licensed program by issuing known error correction information to the customer reporting the problem and/or issuing corrected or notice of availability of corrected code. However, IBM does not guarantee service results or represent or warrant that all errors will be corrected. Any onsite programming services or assistance will be provided at a charge.

WARRANTY

EACH LICENSED PROGRAM IS DISTRIBUTED ON AN 'AS IS' BASIS WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS OR IMPLIED.

Requests for copies of IBM publications should be made to your IBM Representative or to the IBM Branch Office serving your locality.

Address comments concerning the contents of this publication to IBM Corporation, Midwest Region Education Center, 1 IBM Plaza, 20th Floor, Chicago, Illinois 60611.

TABLE OF CONTENTS

Introduction -

This Program Description/Operations Manual provides Screen Designers, Systems Programmers and Assembler Language Applications Programmers with the information they need to use the 3270 FORMAT Macro for Output Data Streams in their Teleprocessing System. All readers should have a specific knowledge of the operation of the 3270 and in particular, the formatting of output Write and Write Erase data streams to the Local and Remote 3270 Printer and Screen devices. An excellent educational offering that gives the student this knowledge is course G3687, 3270 Operation and Design, taught at most IBM Education Centers.

The Systems Programmer's main concern is with the requirements necessary to include the FORMAT Macro support in his system. This information is found in the 'Installation Instructions' section.

The Screen Designer is responsible for laying out on a Screen Design Layout Form all of the necessary information for a Write or Write Erase output data stream. To assist the designer, Screen Design Layout Forms for both the 480 and 1920 size 3270 screens are provided in Appendix A. The use of these forms is found in the section 'Operations Instructions - Using the Screen Design Layout Form'.

The Application Programmer is responsible for writing the FORMAT Macro in his Assembler Language program by coding a series of FORMAT operands using the Screen Design Layout Form as a guide. The Macro Generator will check coding of the FORMAT Macro operands and will provide an output data stream acceptable for transmission to the 3270. The Screen Designer and Application Programmer work together in removing unacceptable or 'abort' macro MNOTES created by the Macro Generator. Appendix E contains a list of MNOTES in numerical order. Information on coding the FORMAT Macro is contained in the section 'Operations Instructions - Coding the FORMAT Macro'.

Appendix B and D contain sample FORMAT Macro Screen Layouts and expansions. Appendix C contains a quick summary of the FORMAT Macro Operands.

System Overview

The 3270 FORMAT Macro allows the Assembler Language Programmer to create write and write erase output data streams for the Local and Remote 3270 display screen and printer devices. Through a high level series of coded operands, the Programmer transfers a screen design layout form into Macro operands. The macro generator checks the operands and creates a string of constants comprising an output data stream.

In converting the operands to an output data stream, numerous checking and utility functions are performed such as computing, checking and translating Set Buffer Addresses from Row and Column co-ordinates as well as computation, checking and translation of Attribute Characters and Write Control Characters. The following functions/advantages are included in the FORMAT Macro:

+ Convert high level operands coded in sequence from a screen design layout form into acceptable command and order sequences.

+ Allows the Programmer to specify absolute decimal addresses for Set Buffer Address, Repeat to Address, etc., sequences, or to use the easy to code Row and Column notation, with checking, conversion and translation provided for all addresses by FORMAT.

+ Computes the Write Control Character utilizing high level character notation, combining multiple functions into a single Write Control Character with checking and translation provided by FORMAT.

+ Checks for coded inconsistencies in the Write Control Character. The following inconsistencies are checked:

    + That Restore Keyboard is not specified for a printer device.

    + That two printer commands such as honor NL and EM and honor 40 character line are not incorporated into the same Write Control Character.

    + That Reset Modified Data Tags is not used with an Erase Write, warning.

+ Computes, checks and translates the attribute character from a high level character string allowing default values for the attributes. The following inconsistencies are checked:

    + That Selector Pen Detect and Invisible are not both coded.

    + That High Intensity and Invisible are not both coded.

    + That Protected or Numeric are not coded when Autoskip is.

+ Examines Selector Pen Detectable fields as specified by the Programmer for adherence to the following conventions:

    + A valid first character, i.e., null, blank, GT (greater than) or ?

    + A minimum of one detectable character beyond the first character.

    + Warning when appropriate on minimum size of the detectable field.

    + Warning when Programmer may be assuming an erase write and has not so specified.

    + Warning when a field extends beyond the end of a Row and into the following Row.

    + Warns the Programmer on inconsistent use of the Modified Data Tag in conjunction with Selector Pen Detect fields.

    + Warns the Programmer when three nulls or blanks are not included at the end of the Selector Pen data field.

+ Sequence checks the order of coded operands for inconsistencies in:

    + Invalid Buffer Address specifications.

    + Out of sequence Buffer Address specifications both explicit and implicit. This capability may be supressed by the user.

    + Checks for screen wrap operations that overlay previously coded fields regardless of starting location on screen. This capability may be supressed by the user.

    + Checks for missing or incorrectly specified operands.

    + Checks configuration specifications against coded operands, for example:

        + Using STX, ESC or ETX sequences for a Local device.

        + Using Erase Unprotected to Address or Programmed Tab together with Erase Write.

+ Checks Repeat to Address, Erase Unprotected to Address, Programmed Tab, Insert Cursor and other character sequences for:

    + Missing, inconsistent or incorrectly specified information.

    + Invalid Buffer Addresses.

- Screen Wrap operations that overlap on previously coded fields. This capability may be supressed by the user.

+ Provides the Programmer with flexibility in coding the Macro operands so that he may:

  - Default to next buffer location addresses for Set Buffer Address Sequences (implicit specification), explicitly use Set Buffer Addresses sequences or eliminate Set Buffer Address sequences.

  - Allow the use of field length decimal numbers in conjunction with specifying data fields or allow the Macro to compute field lengths from the supplied data, the former method providing for a more rigid control over Macro field checking.

  - Incorporate Start of Text, End of Text, New Line and end of Medium characters in the data stream.

  - Incorporate or eliminate Start Field orders in output data streams.

  - Utilize the Insert Cursor Order anywhere in the output data stream.

  - Allow the Macro Operands to be coded in any sequence or require the Programmer to code the operands in ascending sequence by buffer address.

+ Provides the Programmer with 61 numbered MNOTE's to inform him of errors or warnings.

  - MNOTES are numbered and pinpoint the operand and sub-operand number causing the MNOTE.

  - MNOTES occur after code generation allowing the Programmer to inspect the faulty generated code in most cases.

  - Some mnotes issue warnings only and document possible problems, for example:

    - Blanks or nulls missing at the end of a Selector Pen data field.

    - Selector Pen fields extending from one Row into the next Row.

    - Use of a New Line in output data streams to a screen device.

    - Using a Programmed Tab or Erase Unprotected to Address operation following an erase write.

+ Other unnumbered MNOTES document useful information for the Programmer, for example:

  + SBA sequence (start buffer address) addresses are MNOTed when not used at the beginning of a definition or order to allow the Programmer to see where the current buffer address is by refering to his program listing.

  + Beginning and ending addresses for a screen.

  + Number of operands processed.

  + Whether or not wrap occurred.

  + The total size of the screen definition in core.

The applications Programmer using the FORMAT Macro should first familiarize himself with the section entitled "Using the Screen Design Layout Form". The examples and illustrations in this Guide assume that a Screen Design Layout Form is the starting point for the Programmer's job and the conventions set up in the section, "Using the Screen Design Layout Form", should be understood by the Programmer before going on.

Program Description

FORMAT is an Assembler Macro which when included in an OS or DOS Assembler program or made available to any Assembler thru a Macro Library, allows the Programmer to create output data streams. Fourteen different types of high level operands coded in sequence for each output data stream are checked by the Macro Generator and result in a series of define constant statements together with documentary statements. The resulting data stream can be written to a Local or Remote 3270 Display Screen or Printer.

In converting the operands to an output data stream, numerous checking and utility functions are performed such as computing, checking and translating Set Buffer Addresses from Row and Column co-ordinates as well as computation, checking and translation of Attribute Characters and Write Control Characters.

FORMAT works with any level of DOS or OS Assembler and can be included with the Assembler program or made available to the Assembler thru a Macro Library. The data streams can be compiled separately or in the user's program.

Installation Instructions

FORMAT is distributed in 80 column cards with the first card a MACRO card and the last card a MEND card. It is thus ready for immediate use in any OS or DOS Assembler program merely by including the macro at the start of the user's source cards. Optionally, FORMAT may be placed in the OS Macro Library or the DOS Source Statement Library by the Systems Programmer.

The FORMAT Macro consists of about 1526 cards. Each card has a sequence number beginning with #1, the MACRO card. The sequence number is in columns 77-80. Comments are included in FORMAT to document its method of operation. These comments together with the Systems Guide for FORMAT allow the user to make modifications or understand its method of operation.

OPERATIONS INSTRUCTIONS - USING THE SCREEN DESIGN LAYOUT FORM

Any Screen Design Layout Form can be used with FORMAT such as those used in Appendix A or the IBM 3270 Layout Form, GX27-2951.

Screen Name

The screen name should be 1 to 8 characters, the first character must be alphabetic. In order to identify the information returned and to assist the Programmer as well during error recovery operations, the screen designer might consider placing the screen name or a unique screen number on the screen at a pre-defined field location marked Protected, Invisible (non display/non print) with the Modified Data Tag set on. Each input data stream would then be uniquely identified. Placing this identification in the upper left hand field of the screen would allow early detection of the screen type in a scan of the input data stream where the identification is also to be used as a transaction code by the Programmer. One must be careful to avoid a "Locked Buffer" if the Copy Command is to be used.

Write or Erase Write

The designer should be aware that if he does not specify erase write, the buffer is not cleared to nulls and the Buffer Address and Cursor Locations are not set to 0. As a consequence, data character entry can begin, in the absence of a SBA sequence, at the buffer address containing the cursor or the current buffer address. The FORMAT Macro requires that the Programmer specify a SBA starting sequence when a write command is used instead of an erase write. For the most part, write commands will be used to send data to a previously for-

matted screen, i.e., one in which attribute characters have already been stored in the buffer. For this reason, the FORMAT Macro will MNOTE Erase Unprotected to Address and Programmed Tab Orders when specified with an erase write.

Write Control Character for Printer or Screen Operation

Certain items such as honor NL and EM codes, 40, 64 and 80 character lines and other operations should be checked. Note that these items are mutually exclusive and will cause a Macro Abort if incorrectly specified.

Defining Fields

The Designer should now, as shown in Figure 1, enter the fields and attribute characters for fields that comprise his screen. Figure 1 assumes that an initial format is to be sent to the screen with an erase write command and that all fields will be preceded with attribute characters. For data involving a write operation, there would probably be few attribute characters sent, if any, and the placing of data within an existing field would be more common. The placing of data within existing fields can be done by a) SBA sequences followed by data, b) Erase Unprotected to Address sequences, c) Repeat to Address sequences followed or not by data and d) Programmed tab sequences followed or not by data.

OPERATIONS INSTRUCTIONS - CODING THE FORMAT MACRO

Each output data stream that you will code with the FORMAT Macro will consist of a single FORMAT statement appropriately labeled. Following the FORMAT statement will appear one or more operands. Each operand must be enclosed in parenthesis. Thus, the general appearance of the macro statement will be:

```
NAME        FORMAT ( ),( ),( ), Etc.
```

All operands must be separated by commas with no intervening blanks. The NAME which is required may be any valid label in your program. It will be used in an EQU * statement to identify the beginning of your output data stream.

Within each operand there may be suboperands. For example:

```
SCREEN      FORMAT  (WCC,EW,RR),(PH,R1C1,'NAME'),(HN,42), Etc.
```

WCC, EW and RR are the 1st, second and third suboperands respectively of operand 1, (WCC,EW,RR). PH, R1C1 and 'NAME' are the first, second and third suboperands of operand 2, etc. Note that a suboperand can be enclosed in quotes, 'NAME', can

be a pure decimal number, 42, or can be letters and numbers mixed and not enclosed in quotes, PH, R1C1, etc. Regardless of how specified, each suboperand is still separated by commas and no intervening blanks are allowed between suboperands. The number of commas plus one within any given operand equals the number of suboperands within an operand. Later, reference will be made to the effect that four suboperands are implied. All this means is that there are three commas within the operand.

You may have 100, 200 or 240 operands in any output data stream depending on the level of DOS or OS Assembler you are using. (See the section, System Capabilities and Limitations.) There are 14 different types of operands and the discussion that follows gives the rules for coding them. An operand may be split at any point on a source statement card in your program merely by coding a non blank character in column 72 and continuing the operand in column 16 of the next card. If you do not desire to split an operand, you may finish the operand on one card with a right parenthesis, code a comma and a non blank character in column 72 and start the next operand in column 16 of the following card. In this way, one operand per card could be coded. The only rule to watch out for is that an operand may not exceed 127 or 255 characters depending on the level of DOS or OS Assembler you are using. This figure includes beginning and ending parenthesis, quotes and the commas used to separate the operands. Later in this Guide, it will be shown how to overcome the limitation of 127 characters in a suboperand if this is encountered. A higher level version of the Assembler can also be used in which case each of the 200 or 240 operands may be up to 255 characters in total length. (See the section, System Capabilities and Limitations.)

The sequence in which you code the operands depends on your screen design layout. A certain few of the 14 operand types must come first (unless coded in a previous FORMAT Macro in which case they can be omitted), then the rest can be coded in any order so long as ascending order is maintained from order to order in a wrap fashion in your output data stream. Alternatively, you may code the CHECKING=NO operand as outlined in Appendix I and can then code the operands in any sequence.

Within an operand, the sequence of suboperands is always very important. If you omit a suboperand you must never omit the commas that separate the suboperands. In the previous example, if you omitted the first suboperand of the second operand you must code (,R1C1,'NAME'). If you were to omit both the first and second suboperands you would code (,,'NAME'). If, however, you do not use a third operand you merely code (PH,R1C1) or (,R1C1), i.e., you do not need a comma to indicate suboperands missing from the right hand end.

With one of the operand types, you will find that if you omit the first suboperand you must not omit the 2nd or if you

omit the third suboperand you must not omit the 4th. Rules of this sort must be followed.

If, within a suboperand that begins and ends in single quote marks,', the data you desire to use is an apostrophe (single quote) or an ampersand, &, then you must code two apostrophes in a row or two ampersands in a row within the single quotes. For example, a suboperand consisting of ABC' would be coded as 'ABC'''.

Format uses MNOTES to document three types of conditions, abort, warning and information useful to the programmer. In the later category are included statistics on a) starting and ending buffer locations, b) number of operands processed, c) whether or not wrap occured, d) omitted SBA sequences and e) the number of bytes of core taken up by each output data stream. Warning MNOTES will appear in the diagnostic listing even with perfect assemblies. If this is disturbing, it is suggested that all output data streams be coded within a separate CSECT together with ENTRY statements for each screen name. V type address constants or A type address constants with EXTRN's can then be used in other CSECTs that have need to refer to the output data stream.

Configuration Operands (Local and Remote)

The FORMAT Macro needs to know the buffer size, type of configuration and device type in order to adequately check other operands and warn you of errors or abort the macro expansion if necessary. If the configuration operands are not coded as the first operands the FORMAT Macro will assume that the device is a Local Screen utilizing a 480 character buffer. The Macro assumes that the first three configuration operands are:

```
NAME    FORMAT (DEVICE=SCREEN),(BUFFER=480),(TYPE=LOCAL)
```

Since there are no commas, each of the three configuration operands contain a single suboperand. If you have a 480 character local screen you could omit all three unless you wanted to document the use of the macro. On the other hand, if your output data stream were for a 1920 character remote printer, you would be required to code:

```
NAME    FORMAT (DEVICE=PRINTER),(BUFFER=1920),(TYPE=REMOTE)
```

since in all three areas you differ from the default values of screen, 480 character buffer and local configuration.

Once you have specified the three values and are coding another FORMAT Macro, the three previous values will be in effect and you need not code them again in the same program unless one of the values changes.

In summary then, there are six configuration operands which may or may not have to be used. If they are used, they must always come as the first operands. From one to three configuration operands may be used for any one screen. Subsequent screens need not use the operands if all three items remain the same.

Example: A Programmer starts an output data stream to a 480 character screen at a remote location and does not desire to document the Macro. For his first FORMAT Macro usage, he codes:

```
SCREEN      FORMAT (TYPE=REMOTE),
```

The default values for a 480 character buffer and a screen device are automatically in effect. For the next FORMAT Macro usage in the same program, he can omit (TYPE=REMOTE). Only if he changes one of the three items need he code a configuration operand again in a subsequent FORMAT Macro in the same program.

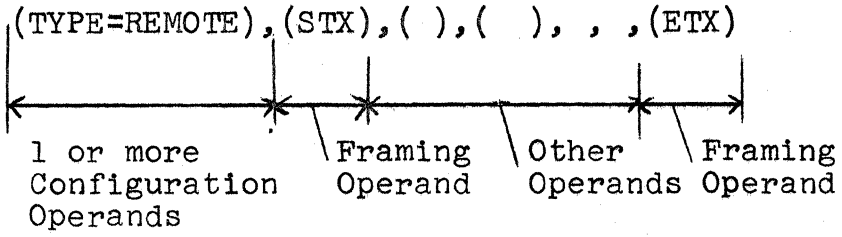
Framing Operands (Remote Only), Optional

If you are working with a remote device you may want to frame your output data stream by coding the (STX) operand and the (ETX) operand.

To use the framing operands merely insert the (STX) following the configuration operands and put the (ETX) operand as the last operand in the Macro.

Example: A Programmer knows that his output data stream will be sent as a single block transmission ending in ETX. He codes his macro as follows:

```
SCREEN    FORMAT (TYPE=REMOTE),(STX),( ),(  ), , ,(ETX)
```



The configuration operand (TYPE=REMOTE) is required when using the (STX) or (ETX) operands.

Write & Erase Write Command Operand (Remote)

Following the configuration and framing operand (STX) if used, you should code the write or erase write command operand for the remote. The format of this operand is as follows:

(ESC, [EW, / W,] 1 or more suboperands separated by commas) to represent bits to be set and translated by FORMAT in the Write Control Character. See Table of Suboperands for the Write Control Character.

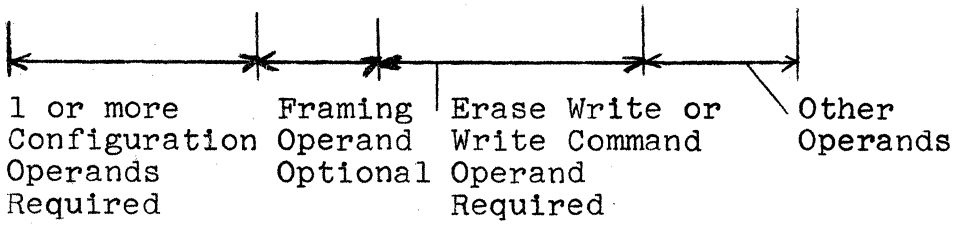
Table of Suboperands for the Write Control Character

One or more suboperands may be coded beginning with the third suboperand of a write or erase write command operand for a remote or local device.

Suboperand Codes	Meaning	Comment
MDT	Reset Modified Data Tags	Should not be used with erase write.
RR	Restore Keyboard	Should not be used when sent to printer.
SP	Start Printer	
80	Print 80 Character Line	If used, avoid also using 64, 40 and NL.
64	Print 64 Character Line	If used, avoid also using 80, 40 and NL.
40	Print 40 Character Line	If used, avoid also using 80, 64 and NL.
NL	Honor NL & EM Codes	If used, avoid also using 80, 64 and 40.
A	Sound Audible Alarm	

The first suboperand, ESC, must always be coded. The second suboperand must either be an E or EW. The third suboperand is also required. The third and subsequent suboperands must be valid suboperand codes taken from the table of suboperands for the write control character. The location of the write and erase write command operand should immediately follow the framing operand, if used, in the following fashion:

```
SCREEN    FORMAT (TYPE=REMOTE),(STX),(ESC,EW,RR,A),( ),( ),,(ETX)
```



Although FORMAT will not Abort if the write or erase write command operand is omitted, FORMAT does make use of the fact that an erase is being done in its checking of subsequent

operands and will assume a write is being done if no write or erase write command operand is given. FORMAT will also check that two printer techniques have not both been coded, i.e., that only one suboperand from 80, 64, 40 and NL is used.

Example: A Programmer desires to issue a write command that restores the keyboard but does not reset modified data tags. He codes:

```
SCREEN    FORMAT (TYPE=REMOTE),(STX),(ESC,W,RR),(  ),
```

The generated code would be:

```
1977+SCREEN    EQU   *
1978+          DC    X'02' STX
1979+          DC    X'27' ESC
1980+          DC    X'F1' WRITE
1981                 *, WRITE CONTROL CHARACTER IS HEX 02
1982+          DC    AL1(194) ADDRESS, ATTRIBUTE OR WCC
```

The Write Control Character will always be documented in a HEX value as shown in statement 1981. Following the statement will be listed the translated value for the WCC. The translated value, statement 1982, will be in the form of a single byte address constant and will be labelled ADDRESS, ATTRIBUTE OR WCC. This is because the translation routine in FORMAT is common to translating addresses, attributes and write control characters. Backward reference to statement 1981 serves to pinpoint 1982 as a write control character in this example.

Example: A Programmer desires to issue an erase write command to a remote printer that will start the printer and print an 80 character line. He codes:

```
SCREEN    FORMAT (TYPE=REMOTE),(DEVICE=PRINTER),(STX),(ESC,EW,SP,80),
```

The generated code would be:

```
1977+SCREEN    EQU   *
1978+          DC    X'02' STX
1979+          DC    X'27' ESC
1980+          DC    X'F5' ERASE WRITE
1981                 *,WRITE CONTROL CHARACTER IS HEX 38
1982+          DC    AL1(248) ADDRESS, ATTRIBUTE OR WCC
```

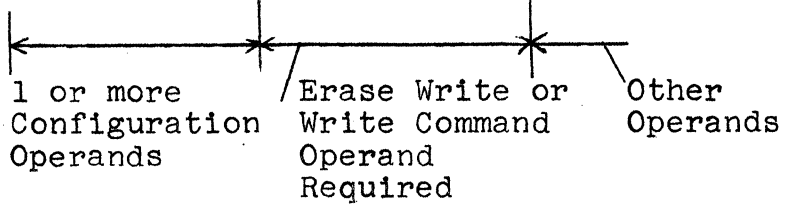
Write and Erase Write Command Operand (local)

Following the configuration operand, if used, you should code the write or erase write command operand for the local. The format of the operand is as follows:

(WCC, [EW, / W,] 1 or more suboperands separated by commas) to represent bits to be set and translated by FORMAT in the Write Control Character. See Table of Suboperands for the Write Control Character in the section entitled "Write & Erase Write Command Operand (remote)".

The first suboperand, WCC, must always be coded. The second suboperand must be either an E or an EW. The third suboperand is required. The third suboperand and subsequent suboperands must be valid suboperand codes taken from the Table of Suboperands for the Write Control Character. The location of the write and erase write command operand should immediately follow the configuration operands, if used, in the following fashion:

```
SCREEN    FORMAT (BUFFER=1920),(WCC,EW,RR,A),(  ),
```



Although FORMAT will not terminate if the write or erase write command operand is omitted, FORMAT does make use of the fact that an erase is being done in its checking of subsequent operands and will assume a write is being done if no write or erase write command operand is given. FORMAT will also check that two printer techniques have not both been coded, i.e., that only one suboperand from 80, 64, 40 and NL is used.

Example: A Programmer desires to issue a write command that restores the keyboard but does not reset modified data tags. He codes:

```
SCREEN    FORMAT (WCC,W,RR), (·),
```

The generated code would be:

```
1980+SCREEN    EQU   *
1981                 *,WRITE CONTROL CHARACTER IS HEX 02
1982+          DC    AL1(194) ADDRESS, ATTRIBUTE OR WCC
```

The Write Control Character will always be documented in a HEX value as shown in statement 1981. Following the statement will be listed the translated value for the WCC. The translated value, statement 1982, will also be in the form of a single byte address constant and will be labeled ADDRESS, ATTRIBUTE OR WCC. This is because the translation routine in FORMAT is common to translating addresses, attributes and write control characters. Backward reference to statement 1981 serves to pinpoint 1982 as a write control character.

Example: A Programmer desires to issue an erase write command to a local printer that will start the printer and print an 80 character line. He codes:

```
SCREEN    FORMAT (DEVICE=PRINTER),(WCC,EW,SP,80),(  ),
```

The generated code would be:

```
1980+SCREEN    EQU   *
1981                 *,WRITE CONTROL CHARACTER IS HEX 38
1982+          DC    AL1(248) ADDRESS, ATTRIBUTE OR WCC
```

Basic Operand (Local & Remote)

The basic operand is the most flexible and most often used operand in any output data stream. With the basic operand, the Programmer can:

- Set up a start field sequence with or without a preceding Set Buffer Address order and with or without data following the start field order.
- Write data into the output data stream (without a preceding start field order) with or without a preceding Set Buffer Address order.

The flexibility is also indicated by the numerous entries in Appendix C, FORMAT Macro Instruction Format Summary, under the Basic Operand. The format of the operand is as follows:

```
Suboperand #1,          Suboperand #2,          Suboperand #3
Attribute               Set Buffer              Data
Character               Address                 Suboperand
Suboperand              Suboperand

(  [X,                  [Row & Column Sequence, [Data in Single
    Omitted,             Decimal Number 0-1919,  Quotes,
    Attribute Sequence,  X,                      Omitted,
    (See Table of        Omitted,               ]
    Attribute Characters)]

Suboperand #4,
Field Length
Suboperand

   [Length of Field
    in Decimal Number,
    Not Used            ]      )
```

In coding a basic operand, you must always code two or more suboperands, i.e., a minimum of one comma is mandatory. If you omit the second suboperand and only decide to use the first suboperand, you must still code the second suboperand as an X.

Prior to examining each of the suboperands, it is well to keep in mind the most common usage of the basic operand.

Required: Set up a SBA order at Row 3, Column 15 and place an attribute character for high intensity, protected data in that location. Fill the field with the data, 'NAME'. Translate the attribute character and convert and translate the SBA order address.

Solution: Code the basic operand as (PH,R3C15,'NAME')

The first suboperand sets up a protected, high intensity attribute character following the SBA sequence specified by the second operand. Data, 'NAME', follows in the output data stream. The sequence in which one codes the basic operands and intermixes them with RA, Repeat to Address operands, EUA, Erase Unprotected to Address operands and PT, Programmed Tab operands must be in ascending order beginning with a buffer address or defaulting to the first location in the buffer and continuing in a wrap fashion up to but not beyond the starting point. If you do not desire to code the FORMAT Operands in ascending sequence, you should code the CHECKING=NO operand as outlined in Appendix I. For example, I may choose to start my output data stream at the 4th row, column 1 of a 480 character buffer, code successively higher and higher basic, RA and EUA opeaands and wrap to the first row of the screen, ending with a basic operand defining a field on the third row. The FORMAT macro has its own internal current buffer address which is incremented just as the hardware CBA is incremented when data is written into the buffer. FORMAT keeps track of where you start your coding and if you wrap and come back and begin to exceed your starting address, Macro abort occurs with an appropriate mnote message. In order to use FORMAT then, you must code operands in ascending sequence from your starting operand location. You may wrap but must not exceed this starting address. The only exception is the Insert Cursor operand which can be coded to be placed in sequence or out of sequence if desired. If you specify CHECKING=NO as outlined in Appendix I, you can code all of the operands in any sequence.

At the end of a successful expansion, FORMAT prints out the starting 3270 buffer address and ending 3270 buffer address. Note that this is quite different from the number of core positions used as the later includes control characters, addresses, orders, etc. FORMAT will also print out the total number of operands processed and whether or not wrap, as defined above, occurred. If you are using a write command instead of an erase write, FORMAT will require that you begin your output data stream with an SBA sequence whether you start the output data stream with a basic operand, RA, EUA or insert cursor operand. This means that if you start with a basic operand after your write command operand, a valid buffer address entry in the sec-

ond suboperand of the basic operand is required. The same holds true if you were to start with a RA, EUA or IC Operand.

The first suboperand is used to set up an attribute character and will generate a start field order followed by a checked and translated attribute character. If the suboperand is coded X, no start field order will be generated. If omitted, a default attribute character defined as unprotected, alphameric, normal intensity, non selector pen detectable with the modified data tag turned off will be generated. These are all of the default values for the attribute character. The user may alter the default values by setting up his own attribute character by using an attribute sequence and refering to the table of attribute characters below.

Table of Attribute Characters

Value Code	Attribute	Comment
P	Protected	Not to be used with a value code A.
N	Numeric	Not to be used with value code A.
A	Autoskip	A=P+N, neither P nor N should be used.
S	Selector Pen Detectable	Not to be used with I.
H	High Intensity	Not to be used with I.
M	Modified Data Tag	
I	Invisible	Not to be used with S or H.

For example, (AS, R1C1) defines an attribute character as a selector pen detectable, autoskipped field. If the first character of the field is a ø,ø, greater than or ? then it will truly be selector pen detectable. The default values of normal intensity and no modified data tag are also in effect. (HMN,R3C4) defines an unprotected numeric field of high intensity with the modified data tag set on. If the first character of data in the field is a ø, ø, greater than or question mark then it will also be selector pen detectable as well. This is because of the way in which the attribute character is calculated. The high intensity bit will also make the field selector pen detectable. It is the presence of one of 4 designator characters (?, greater than, null or blank) that makes the field truly detectable. (,R1C1) defines a default attribute: unprotected, alphameric, normal intensity, non selector pen detectable with the modified data tag turned off. The code generated for the start field order is:

For a first suboperand coded N (numeric):

```
2180+          DC    X'1D' SF
2181                 *,ATTRIBUTE CHARACTER IS HEX 10
2182+          DC    AL1(80) ADDRESS, ATTRIBUTE OR WCC
```

For a first suboperand coded NS (numeric, detectable):

```
1910+          DC    X'1D' SF
1911                 *,ATTRIBUTE CHARACTER IS HEX 14
1912+          DC    AL1(212) ADDRESS, ATTRIBUTE OR WCC
```

As with the write control character, the calculated bit configuration of the untranslated attribute character is given in hexidecimal form. The translated value is also given in both its decimal and hexidecimal equivalent at the left hand side of the listing. The table of attribute characters lists various invalid combinations that cause Macro Abort conditions. Note that Autoskip, A, implies P and N and that P and A, N and A, and P and N, are invalid.

The second suboperand, (R1C1), defines an SBA sequence which will precede the start field sequence. If the start field sequence has been omitted then the SBA sequence only will be generated, for example (X, R1C3). The SBA sequence can be omitted by coding an X or by a ,, (comma, comma) sequence. Whenever coding the basic operand, at least two suboperands must be used. A start field sequence with no preceding SBA sequence would be coded (IP,X) and would generate only a start field order followed by an attribute character defining an invisible, protected, alphanumeric field. If you use three or more suboperands you may use either X or ,, to omit the second suboperand. You cannot code (IP,,) because the third suboperand or data operand, if implied, as it is in this case due to the presence of two commas, must never be omitted unless a valid 4th suboperand is coded which it isn't. Both (IP,,'DATA') and (IP,X,'DATA') would achieve the effect of omitting the SBA suboperand when three suboperands are used.

In coding the second suboperand, you may use Row and Column notation or decimal notation. 480 character buffers have 12 lines (rows) with 40 positions per line (columns). Valid specifications are rows 1 to 12 and columns 1-40. Any other rows or columns are invalid. 1920 character buffers have 24 lines (rows) with 80 positions per line (columns). Valid specifications are rows 1-24 and columns 1-80. Alternately, a decimal number of 0 thru 479 or 0 thru 1919 may be specified. The vertical column on the lefthand side of the Screen Design Layout Form is designed to assist one in using decimal numbers. All digits must be numeric and the decimal number must be within the correct buffer size range as specified in the buffer configuration operand or defaulted to, i.e., 480. A basic operand coded (SH,R7C35,'?bTYPEbbb') generates:

```
1440+           DC      X'11' SBA
1441                    *,SBA BUFFER ADDRESS IS 274 DECIMAL
1442+           DC      AL1(196) ADDRESS, ATTRIBUTE OR WCC
1443+           DC      AL1(210) ADDRESS, ATTRIBUTE OR WCC
1444+           DC      X'1D' SF
1445                    *,ATTRIBUTE CHARACTER IS HEX 08
1446+           DC      AL1(200) ADDRESS, ATTRIBUTE OR WCC
1447+           DC      CL9'?bTYPEbbb'
```

The SBA address order precedes the start field order. Two address bytes follow the SBA order. (The HEX representations of the A type Addresses appear at the left hand margin.)

If the Programmer had coded (SH,274,'?bTYPEbbb'), the results would have been identical. If the current buffer address is set at the desired location by a previous operand, the SBA sequence is not required. For example, the prior operand may specify a repeat to address to location 274 or an erase unprotected to address to location 274 or insert data into location 273. The Programmer could code (SH,,'?bTYPEbbb') or (SH,X,'?bTYPEbbb') and the following would result:

```
1440                    *,MNOTE 55, SBA SEQUENCE AT 274, OMITTED
1441+           DC      X'1D' SF
1442                    *,ATTRIBUTE CHARACTER IS HEX 08
1443+           DC      AL1(200) ADDRESS, ATTRIBUTE OR WCC
1444+           DC      CL9'?bTYPEbbb'
```

MNOTE 55 is generated to document the omitted SBA sequence. When data is read back from this field, it will be necessary to know the address plus one of the attribute character. If the Programmer had been working with Row and Column notation or had omitted the SBA sequence, he could use FORMAT MNOTE 55 in the later case, as input to routines that examine input data streams for incoming SBA sequences. With Row and Column notation, he merely has to examine his listing to locate the proper decimal value in the two byte address constant. One Assembly documents the address which can be utilized in the next Assembly of the program.

The Programmer can code (,,'DATA') in which case he will generate:

```
1440                    *,MNOTE 55, SBA SEQUENCE AT 274, OMITTED
1441+           DC      X'1D' SF
1442+                   *,ATTRIBUTE CHARACTER IS HEX 00
1443+           DC      AL1(64) ADDRESS, ATTRIBUTE OR WCC
1444+           DC      CL4'DATA'
```

This results in an omitted SBA sequence and a default attribute character value. The same result can be achieved by coding (,X,'DATA').

Also possible at this point would be (X,,'DATA') or (X,X,'DATA') in which case FORMAT would generate:

```
1440                    *,MNOTE 55, SBA SEQUENCE AT 274, OMITTED
1441+          DC       CL4'DATA'
```

Now, both the SBA and start field orders have been omitted.

At this point, read APPENDIX F, MNOTES for SBA Sequences - Discussion.

The third suboperand of the basic operand is used to place data in the write output data stream. The data is enclosed in single quotation marks. The single quote marks are mandatory. If data within single quotes is omitted in the third suboperand and a fourth suboperand is implied then the 4th suboperand must be present and must contain a valid decimal number. (SH,R1C1,,) is illegal, so is (SH,R1C1,). If the attribute code specifies selector pen detectable then the third suboperand must be present.

The data in quotes is placed into the output data stream. The length of the data within the quote marks will always be used to define the character constant. If the data is so long that the entire operand exceeds 127 or 255 characters including parenthesis, quotes and commas, then the data can be truncated and split into two pieces. Assume that you must split the first 10 characters of the alphabet and place the entire 10 characters starting at location 2. You would code: (X,2,'ABCDE'), (X,,'FGHIJ').

The generated code would be:

```
1440+          DC       X'11' SBA
1441                    *,SBA BUFFER ADDRESS IS 2 DECIMAL
1442+          DC       AL1(64) ADDRESS, ATTRIBUTE OR WCC
1443+          DC       AL1(194) ADDRESS, ATTRIBUTE OR WCC
1444+          DC       AL5'ABCDE'
1445                    *,MNOTE 55,.SBA SEQUENCE AT 7, OMITTED
1446+          DC       CL5'FGHIJ'
```

Although only ten characters were split, the method may be applied any number of times to larger character strings.

If you have specified selector pen detect in the attribute character then you must code a third operand in single quotes and the data within the quotes must conform to the following conventions:

- If the first character of the data is not a ?, >, null or blank you will abort with Mnote 26.

- If the last three positions of the field do not contain three blanks or nulls, you will be warned with Mnote 56.

- If the selector pen data field extends from one row into the next, you will be warned with Mnote 51.

- If > has been specified as the first data character and no modified data tag has been set, you will be warned with Mnote 52.

- If a null, blank or ? has been specified in the first data character together with the modified data tag in the attribute character, you will be warned with Mnote 53.

- If the length of the data between the quotes is only one character, you will abort with Mnote 27.

- If the data between quotes is not at least five characters you will be warned with Mnote 28. (This assumes a designator character, a single displayable character and three nulls or blanks will be the most common form of minimum data.)

The fourth suboperand is optional except when the third suboperand is omitted and a 4th suboperand has been implied by use of three commas. It is used to tell FORMAT the total length of the field when the third suboperand has data in it which is less than the total field length. For example, assume that only the first five characters of a field are to be placed into the field, the rest to contain nulls. For more precise checking by FORMAT you may code (PH,R3C4,'NAMES',10). You could just as easily have coded (PH,R3C4,'NAMES'). In the first case, FORMAT will increase the buffer location by 10 whereas in the later case, FORMAT will increase the buffer location by 5. Suppose now that you make a mistake and code (I,R3C12,'JOHN') as the next operand. FORMAT will catch this error in the first case but not in the second. The next operand should be (I,R3C15,'JOHN').

When you use the 4th operand to tell FORMAT that the data of the third operand does not fill up the entire field, the 4th operand must contain all decimal numbers. For a selector pen detectable field with the 4th operand specified, FORMAT will assume that the remaining positions of the field are to be set to nulls and will warn you with Mnote 29 if you have not specified an erase write.

Format will also protect you in one other way in conjunction with using the 4th operand. Consider the following sequence:

```
1433 T90    FORMAT  (WCC,EW,RR),(X,,'DATA'),(,,'2',10),
                    (N,,'3')
1434+T90    EQU     *
1435                *,WRITE CONTROL CHARACTER IS HEX 02
1436+       DC      AL1(194) ADDRESS, ATTRIBUTE OR WCC
1437                *,MNOTE 55, SBA SEQUENCE AT 0, OMITTED
1438+       DC      CL4'DATA'
```

```
1439                    *,MNOTE 55, SBA SEQUENCE AT 4, OMITTED
1440+           DC      X'1D' SF
1441                    *,ATTRIBUTE CHARACTER IS HEX 00
1442+           DC      AL1(64) ADDRESS, ATTRIBUTE OR WCC
1443+           DC      CL1'2'
1444+           DC      X'11' SBA
1445                    *,SBA BUFFER ADDRESS IS 15 DECIMAL
1446+           DC      AL1(64) ADDRESS, ATTRIBUTE OR WCC
1447+           DC      AL1(79) ADDRESS, ATTRIBUTE OR WCC
1448+           DC      X'1D' SF
1449                    *,ATTRIBUTE CHARACTER IS HEX 10
1450+           DC      AL1(80) ADDRESS, ATTRIBUTE OR WCC

1451+           DC      CL1'3'
1452                    *,STARTING ADDRESS 4, ENDING ADDRESS 16
1453                    *,4 OPERANDS PROCESSED
1454                    *,TOTAL CORE FOR FORMAT T90 WAS 14 BYTES
```

Because an erase write was specified, FORMAT will allow omission of the SBA sequence in the 2nd operand. The default attribute character and omission of the third operand is also OK. Note, however, that the Programmer wishes to omit the SBA order for the 4th operand. If FORMAT allowed this, the start field order would come immediately after the CL1'2' statement. Under these circumstances, FORMAT will not allow omission of the SBA order sequence for the 4th operand and supplies it of its own accord as it has done following the CL1'2'. He cannot omit the SBA order sequence for the 4th operand by coding an X.

Finally, it must be said that (,,,10) and (X,,,10) are allowed where no data is to be written into a field with or without an attribute character, but the field length checking capability is desired. At this point it is well to review APPENDIX G - MNOTES in conjunction with Illegal Wrap Operations - Discussion.

Erase Unprotected to Address Operand (Local & Remote)

The Erase Unprotected to Address Operand is used to place an EUA order in the output data stream with or without a preceding SBA order sequence. It can also be used to erase all unprotected data in an entire buffer. The format of the operand is as follows:

Suboperand #1, Identifier	Suboperand #2 Set Buffer Address Suboperand	Suboperand #3, Erase Unprotected to Address Address Suboperand
(EUA,	[Row & Column Sequence, Decimal Number 0-1919, X, Omitted,]	[Row & Column Sequence, Decimal Number 0-1919])

In coding an EUA operand you must always code three suboperands. The second suboperand only can be omitted, i.e., the first and third suboperands are required. The purpose of the EUA operand is to set up an EUA order in the output data stream and the operand must be coded in its correct ascending screen location sequence intermixed with basic operands, RA, Repeat to Address operands and other operands in an increasing fashion allowing wrap, as described in APPENDIX G and in the discussion of the basic operand. (If CHECKING=NO is specified as per Appendix I, this later requirement is not necessary.) To illustrate its usage assume the following:

Required: Set the buffer address to Row 1, Column three and erase all unprotected data from that location up to but not including location 285.

Solution: Code the EUA operand as (EUA,R1C3,285)

The first suboperand identifies the operand as an EUA operand. The second operand sets up the SBA sequence prior to the EUA order and the third operand sets up the EUA address as required by the order. The generated code would look like:

```
2056+      DC    X'11' SBA
2057             *,SBA BUFFER ADDRESS IS 2 DECIMAL
2058+      DC    AL1(64) ADDRESS, ATTRIBUTE OR WCC
2059+      DC    AL1(194) ADDRESS, ATTRIBUTE OR WCC
2060+      DC    X'12' EUA
2061             *,EUA BUFFER ADDRESS IS 285 DECIMAL
2062+      DC    AL1(196) ADDRESS, ATTRIBUTE OR WCC
2063+      DC    AL1(93) ADDRESS, ATTRIBUTE OR WCC
```

The second suboperand defines an SBA order which, if not omitted thru the use of ,, or X, will precede the EUA order. If omitted as in the following example, MNOTE 55, will be given to pinpoint the value of the omitted SBA sequence. If you coded (N,R1C2,'DATA'), (EUA,,R2C2), the generated code would be:

```
2929 +     DC    X'11' SBA
2930             *,SBA BUFFER ADDRESS IS 1 DECIMAL
2931 +     DC    AL1(64) ADDRESS, ATTRIBUTE OR WCC
2932 +     DC    AL1(193) ADDRESS, ATTRIBUTE OR WCC
2933 +     DC    X'1D' SF
2934             *,ATTRIBUTE CHARACTER IS HEX 10
2935 +     DC    AL1(80) ADDRESS, ATTRIBUTE OR WCC
2936 +     DC    CL4'DATA'
2937             *,MNOTE 55, SBA SEQUENCE AT 6, OMITTED
2938 +     DC    X'12' EUA
2939             *,EUA BUFFER ADDRESS IS 41 DECIMAL
2940 +     DC    AL1(64) ADDRESS, ATTRIBUTE OR WCC
2941 +     DC    AL1(233) ADDRESS, ATTRIBUTE OR WCC
```

The third suboperand which cannot be omitted is used to set up the EUA buffer address and can be specified as a decimal number or with row and column notation. The discussions of APPENDIX F and APPENDIX G apply to both the second and third suboperands of an EUA operand.

If an EUA is specified in an erase write output data stream, a warning is issued. If the programmer desires to erase all unprotected data by coding an EUA address equal to the current buffer location, FORMAT will not consider this a wrap. Any way of bringing the current buffer address to equal the EUA address is legitimate.

Repeat to Address Operand (Local & Remote)

The repeat to address operand is used to place a RA order into the output data stream, with or without a preceding SBA sequence, together with the repeat to address character and optionally, followed by additional data. It may also be used to propagate a single character into all locations of an entire buffer. The format of the operand is as follows:

Suboperand #1, Identifier	Suboperand #2, Set Buffer Address Suboperand	Suboperand #3, Repeat to Address Address Suboperand
(RA,	Row & Column Sequence, Decimal Number 0-1919, X, Omitted,	Row & Column Sequence, Decimal Number 0-1919,

Suboperand #4, Repeat to Address Character	Suboperand #5 Data Suboperand	Suboperand #6 Field Length Suboperand
Single Data Character in Single Quotes, Single Data Character without Quotes,	Data in Single Quotes, Not Used	Length of Field in Decimal Number, Not Used)

In coding a RA operand you must always code four or more suboperands. If you code less than four suboperands, FORMAT will abort the Macro expansion with Mnote 30. The second suboperand can be omitted but the 1st, 3rd and 4th are required. The purpose of the RA operand is to set up a RA order in the output data stream and the operand must be coded in its correct ascending screen location sequence intermixed with basic operands, EUA operands and other operands as described in the discussion on using the basic operand. (If CHECKING=NO is specified as

per Appendix I, this later requirement is not necessary.) To illustrate the usage of the RA operand assume that the following is required.

Required: A large autoskip field on the first 80 character row of a Model 2 screen is to contain the first 10 letters of the alphabet, 54 asterisks, the next 10 letters of the alphabet and 6 nulls, all in high intensity.

Solution: (AH,R24C80,'ABCDEFGHIJ'),(RA,,R1C65,*,'KLMNOPQRST',16)

The first operand sets an attribute character in the last position in the buffer and writes 10 characters in the first ten locations of row 1. The expansion of the RA operand would be:

```
2139                *,MNOTE 55, SBA SEQUENCE AT 10, OMITTED
2140+       DC      X'3C' RA
2141                *,RA BUFFER ADDRESS IS 64 DECIMAL
2142+       DC      AL1(64) ADDRESS, ATTRIBUTE OR WCC
2143+       DC      AL1(127) ADDRESS, ATTRIBUTE OR WCC
2144+       DC      CL1'*'
2145+       DC      CL10'KLMNOPQRST'
```

The second suboperand in the RA operand defines an SBA sequence which if not omitted through the usage of ,, or X, will precede the RA sequence. If the output data stream in the example was not positioned as it was with the first operand that wrote data into row 1, columns 1 thru 10, then the RA operand could have been coded (RA,R1C11,R1C65,*,'KLMNOPQRST',16). If omitted, as in the example, MNOTE 55 will pinpoint the omitted value.

The third suboperand which cannot be omitted, is used to set up the RA buffer address and can be specified as a decimal number or using row and column notation. The discussions in APPENDIX F and G apply to both the second and third suboperands of a RA operand.

The fourth suboperand cannot be omitted and it specifies the single character to be propagated into all buffer locations up to but not including the third suboperand buffer address value. It must be in quotes if it is a blank or a comma. If a character of the alphabet, a number or a special character, you may omit the framing quotes.

Suboperand #5, which is optional, allows you to follow the single propagated data character with data. The data, if specified, must be in single quotes. This data will be written into the buffer starting in the address specified by suboperand #3 which is the 'Stop' address for the Repeat to Address Order. You may specify suboperand #5 by itself or you

may specify both suboperand #5 and #6. You may not omit suboperand #5 and code suboperand #6 by itself. Coding suboperand #6, if used, tells FORMAT that the data field that follows the Repeat to Address sequence, extends beyond the data specified in suboperand #5. The decimal value specified in suboperand #6 includes the data length in suboperand #5 plus any remaining nulls to finish out the length of the field. In the example, there are 16 positions ending the first row but only the first 10 of these have data written into them. Suboperand #6 is used for more precise checking purposes as with suboperand #4 in the basic operand.

FORMAT will not abort the Macro generation if a Repeat to Address order for a length greater than 480 characters is specified and is followed by more data in the output data stream. Such a condition may cause overrun at the 3270 and should be avoided.

Insert Cursor Operand (Local & Remote)

The insert cursor operand is used to place an insert cursor order into the output data stream, with or without a preceding SBA sequence. By coding the insert cursor operand with an X in the 3rd suboperand, FORMAT will allow you to place the IC operand anywhere in the sequence of operands. FORMAT will not alter the value of its own CBA when the third suboperand is specified with an X. Coding of the insert cursor operand under all circumstances does not cause FORMAT to increase the CBA. The format for the insert cursor operand is:

Suboperand #1, Identifier	Suboperand #2, Set Buffer Address Suboperand	Suboperand #3, Out of Sequence Specification Suboperand
(IC,	[Row & Column Sequence, Decimal Number 0-1919, Not Used]	[X, Not Used])

If you desire to insert the cursor order as the very next constant in the output data stream, without any need to set the buffer address, you need only code (IC). If you wish to precede the IC order with an SBA order sequence you should code (IC,R1C3) or (IC,285). You must put the operand in its proper ascending sequence of operands if you use the two later methods. (If CHECKING=NO is specified as per Appendix I, this later requirement is not necessary.) If you do not desire to code the IC operand in its sequential ascending operand order you should code (IC, 285,X) or (IC,R1C3,X). FORMAT will now omit all checking for sequence. This will occur even though CHECKING= YES was coded or defaulted to as outlined in Appendix I. If you use an X in the third operand, you must specify a second

operand. After coding the IC operand with an X in the third operand you must pick up where you left off in the ascending sequence of operands. For example, (IC,55) will generate:

```
1750+       DC      X'11' SBA
1751                *,SBA BUFFER ADDRESS IS 55 DECIMAL
1752+       DC      AL1(64) ADDRESS, ATTRIBUTE OR WCC
1753+       DC      AL1(247) ADDRESS, ATTRIBUTE OR WCC
1754+       DC      X'13' IC
```

To illustrate coding the IC operand with an X in the third sub-operand location, consider the following sequence of operands: (AH,R8C5,'DATA2'),(IC,R7C3,X),(N,R8C15,'X')

The generated code will be:

```
2492+       DC      X'11' SBA
2493                *,SBA BUFFER ADDRESS IS 564 DECIMAL
2494+       DC      AL1(200) ADDRESS, ATTRIBUTE OR WCC
2495+       DC      AL1(244) ADDRESS, ATTRIBUTE OR WCC
2496+       DC      X'1D' SF
2497                *,ATTRIBUTE CHARACTER IS HEX 38
2498+       DC      AL1(248) ADDRESS, ATTRIBUTE OR WCC
2499+       DC      CL5'DATA2'
2500+       DC      X'11' SBA
2501                *,SBA BUFFER ADDRESS IS 482 DECIMAL
2502+       DC      AL1(199) ADDRESS, ATTRIBUTE OR WCC
2503+       DC      AL1(226) ADDRESS, ATTRIBUTE OR WCC
2504+       DC      X'13' IC
2505+       DC      X'11' SBA
2506                *,SBA BUFFER ADDRESS is 574 DECIMAL
2507+       DC      AL1(200) ADDRESS, ATTRIBUTE OR WCC
2508+       DC      AL1(126) ADDRESS, ATTRIBUTE OR WCC
2509+       DC      X'1D' SF
2510                *,ATTRIBUTE CHARACTER IS HEX 10
2511+       DC      AL1(80) ADDRESS, ATTRIBUTE OR WCC
2512+       DC      CL1'X'
```

The third operand (N,R8C15,'X') picks up where the first operand left off and no problems arise due to the second operand being out of sequence with respect to the first and third operands.

Programmed Tab Operand (Local & Remote)

A PT operand is used to put the PT order in the output data stream followed by data or by itself. The format for the PT operand is as follows: (PT) or

Suboperand #1, Identifier	Suboperand #2, Expected Tab Address Suboperand	Suboperand #3 Data Suboperand

(PT, [X, Row & Column Sequence, Decimal Number 0-1919,] [X, Data in Single Quotes])

The second suboperand if not omitted by coding an X is used internally by FORMAT. It contains the expected address of the first character of the data in quotes in the 3rd suboperand. That is to say, it is the address of the first character location in the next highest sequential unprotected field that will be found in a forward scan. The expansion for a PT operand (PT,R5C8,'DATA␣TO␣TABBED'), follows.

```
1434+          DC      X'05' PT
1435+          DC      CL17'DATA␣TO␣TABBED'
```

The second suboperand if used must be in ascending sequence of addresses and can therefore cause any of the messages as described in APPENDIX G. (If CHECKING=NO is specified as per Appendix I, this later requirement is not necessary.) FORMAT will not warn the Programmer when the PT operation causes true wrap from the bottom row to the top row. The only warnings will be as discussed in APPENDIX G. If the Programmer merely desires to clear the end of a field to nulls and not follow the PT order with data, he should code (PT). Note that two (PT)'s in a row may require an intervening null.

New Line and End of Medium Operands (Local & Remote)

By coding a (NL) or (EM) operand, the Programmer may format output to a printer (must be used for a dedicated Model 3 printer).

System Capabilities & Limitations

100 to 240 operands with 127 or 255 characters per operand for each use of the Macro depending on whether a high level of the Assembler is used or not as per the following table:

Assembler	Maximum Number of Operands	Maximum Number of Characters Per Operand
DOS Assembler D	100	127
DOS Assembler F	200	255
OS Assembler F	200	255
OS Assembler H*	240	255

*IBM Product Program

Information in Appendix A thru E - Discussion

Appendix A contains the Screen Design Layout Forms for the 480 and 1920 character buffer devices.

Appendix B contains sample Screen Design Layout forms, in particular, Figures 1 & 2, discussed in the section "Using the Screen Design Layout Form". In addition, a test screen, Figure 3, is given to illustrate the way in which FORMAT will Mnote errors. All three screens, Figures 1 thru 3, have been coded and submitted to FORMAT and the resultant macro expansions are listed in Appendix D.

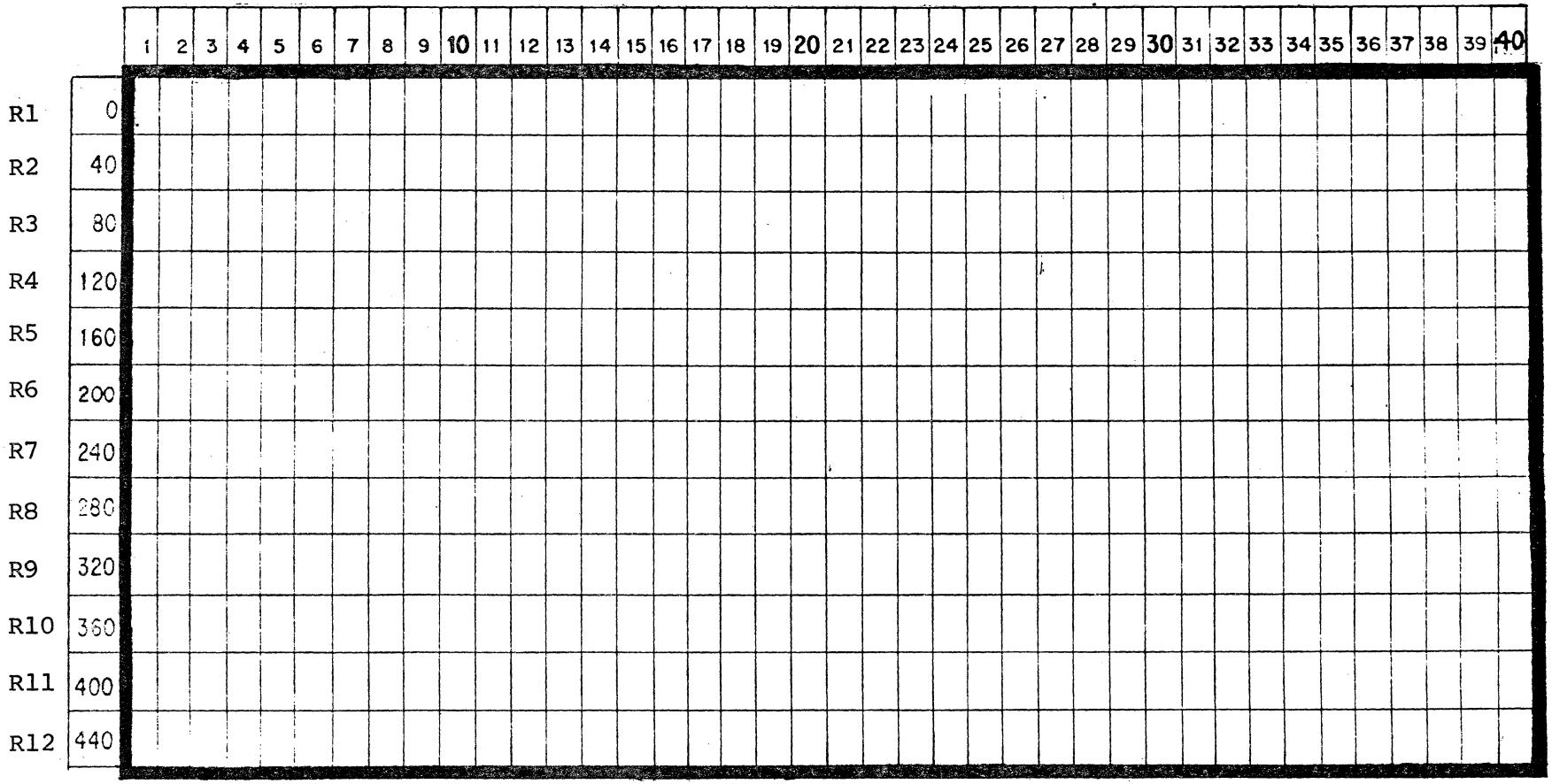
Appendix C contains a listing of all of the FORMAT Macro Instruction Operands in all of the various possible combinations in the form of a Macro Instruction Summary. It can be used as a quick cross check as to possible methods of coding each of the 14 different types of FORMAT operands.

Appendix D contains FORMAT Macro Instruction listings for the three screens of Figures 1, 2 and 3 shown in Appendix B. The discussion of Figures 1 and 2 is contained in the section "Using the Screen Design Layout Form". Coding of the FORMAT Macro Operands is discussed in the section "Coding the FORMAT Macro".

Appendix E contains a listing of all of the MNOTES referred to in the section "Coding the FORMAT Macro". The MNOTES are numbered and appear in numerical order for easy reference. Where information is supplied by the FORMAT Macro at the time the MNOTE is produced in the output listing, Appendix E substitutes three underlines, ___, to indicate that the value will be put into the MNOTE by FORMAT.

Screen Name_____ SCREEN DESIGN LAYOUT FORM - 480 CHARACTER SCREEN

- - - Columns - - -



Check Appropriate Box

△ WRITE △ ERASE WRITE

△ LOCAL △ REMOTE

Write Control Character for Printer Operation

△ START PRINTER

△ HONOR NL & EM CODES

△ 40 CHARACTER LINE

△ 64 CHARACTER LINE

△ 80 CHARACTER LINE

△ RESET MODIFIED DATA TAG

Start Coding Output Stream at Location - _____

Insert Cursor at Location -_____

Write Control Character for Screen Operations

△ RESET MODIFIED DATA TAGS

△ RESTORE KEYBOARD AND RESET AID

△ SOUND AUDIBLE ALARM

Color Code for Attribute Character

P - PROTECTED - BROWN

N - NUMERIC - BLUE

H - HIGH INTENSITY - RED

I - INVISIBLE - YELLOW

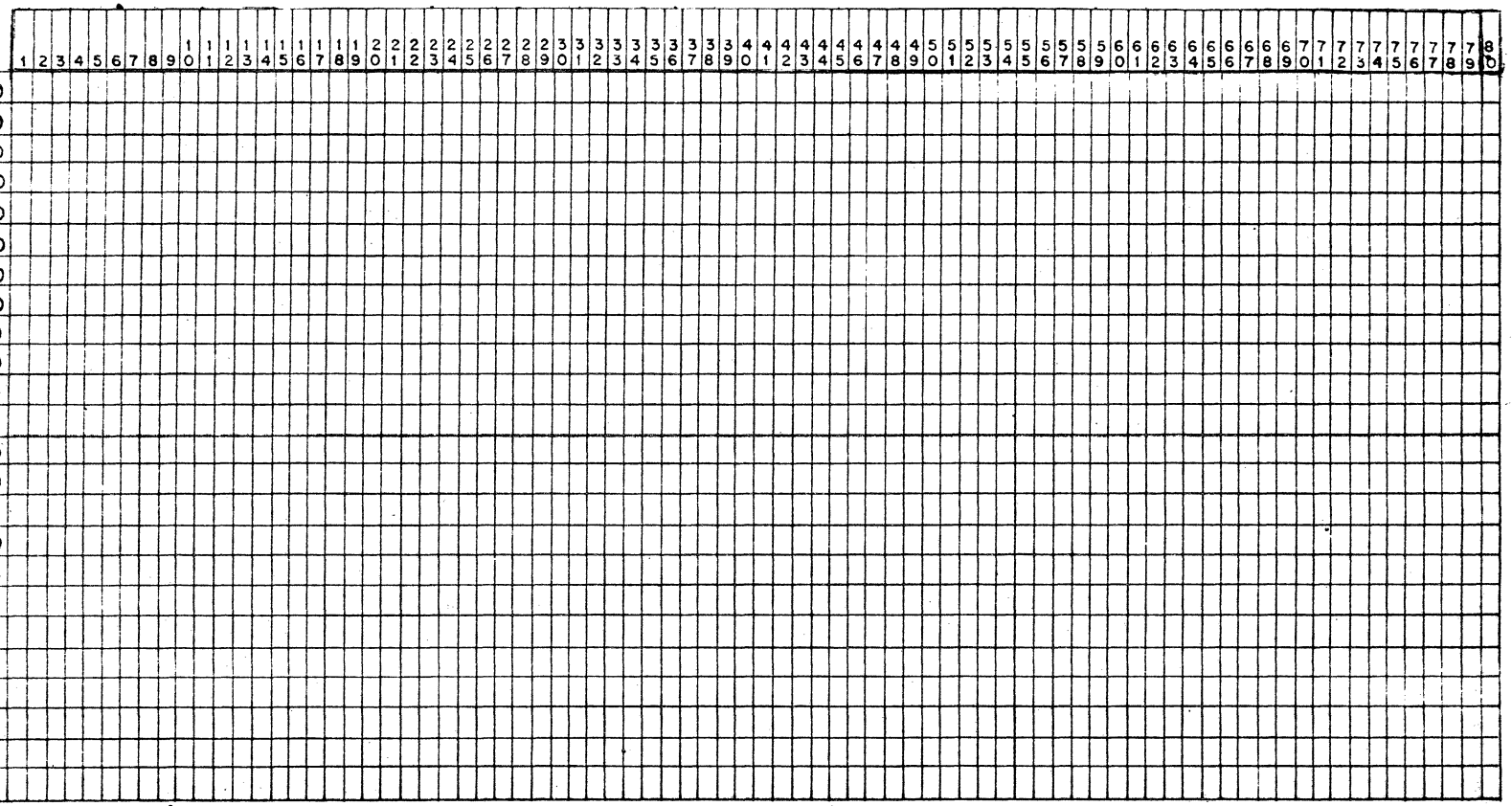
S - SELECTOR PEN DETECT - ORANGE

M - MODIFIED DATA TAG - GREEN

A - AUTOSKIP - PINK (P+N)

Screen Name_____ SCREEN DESIGN LAYOUT FORM - 1920 CHARACTER SCREEN

- - - Columns - - -



Check Appropriate Box

△ WRITE △ ERASE WRITE

△ LOCAL △ REMOTE

Write Control Character for Printer Operation

△ START PRINTER

△ HONOR NL & EM CODES

△ 40 CHARACTER LINE

△ 64 CHARACTER LINE

△ 80 CHARACTER LINE

△ RESET MODIFIED DATA TAG

Start Coding Output Stream at Location - _____

Insert Cursor at Location - _____

Write Control Character for Screen Operations

△ RESET MODIFIED DATA TAGS

△ RESTORE KEYBOARD AND RESET AID

△ SOUND AUDIBLE ALARM

Color Code for Attribute Character

P - PROTECTED - BROWN

N - NUMERIC - BLUE

H - HIGH INTENSITY - RED

I - INVISIBLE - YELLOW

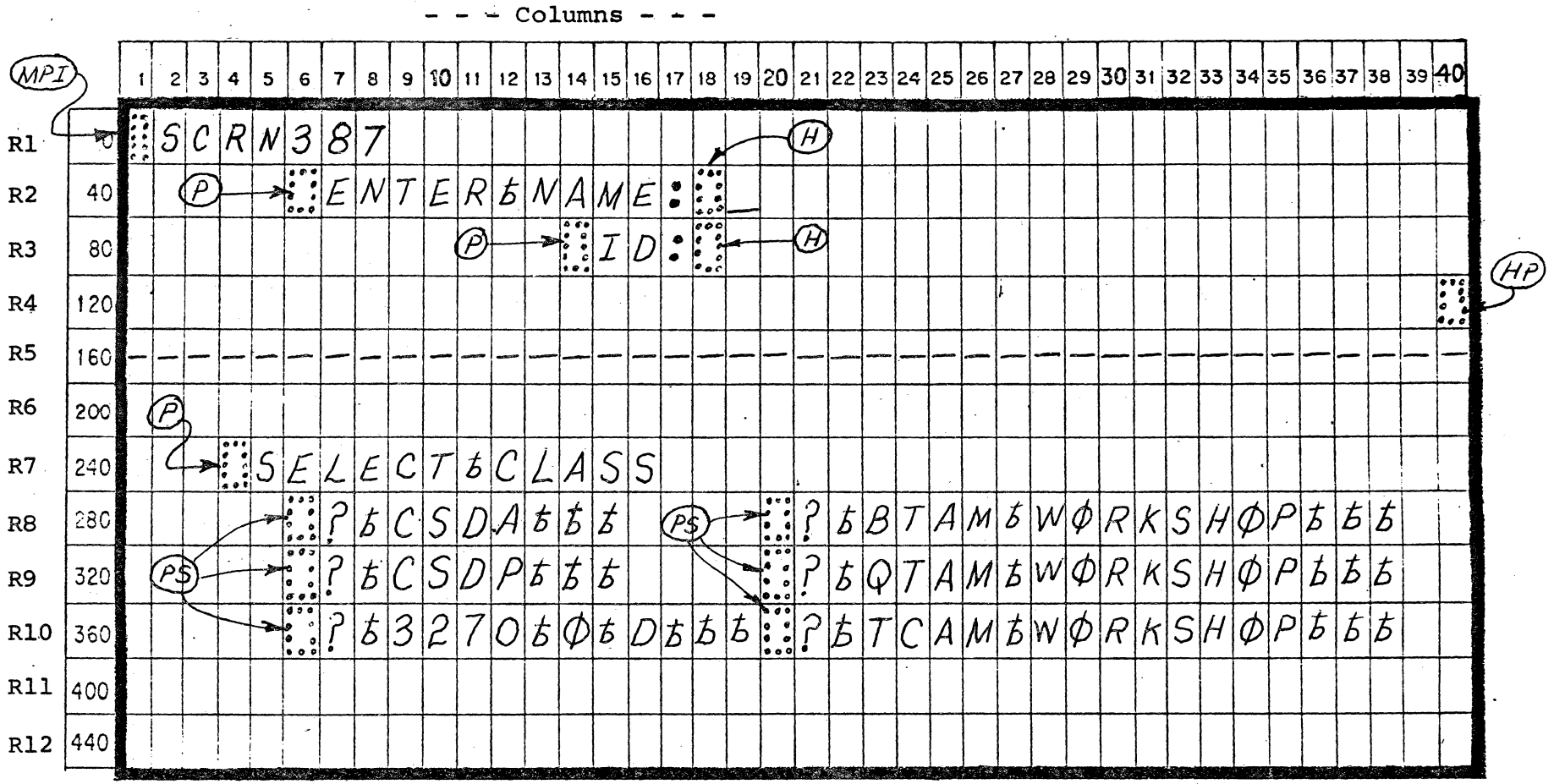
S - SELECTOR PEN DETECT - ORANGE

M - MODIFIED DATA TAG - GREEN

A - AUTOSKIP - PINK (P+N)

Screen Name SCRN387

SCREEN DESIGN LAYOUT FORM - 480 CHARACTER SCREEN



Check Appropriate Box

△ WRITE ☒ ERASE WRITE

△ LOCAL ☒ REMOTE

Write Control Character for Printer Operation

△ START PRINTER

△ HONOR NL & EM CODES

△ 40 CHARACTER LINE

△ 64 CHARACTER LINE

△ 80 CHARACTER LINE

△ RESET MODIFIED DATA TAG

Start Coding Output Stream at Location - R1C1

Insert Cursor at Location - R2C19

Write Control Character for Screen Operations

△ RESET MODIFIED DATA TAGS

☒ RESTORE KEYBOARD AND RESET AID

☒ SOUND AUDIBLE ALARM

Color Code for Attribute Character

P - PROTECTED - BROWN

N - NUMERIC - BLUE

H - HIGH INTENSITY - RED

I - INVISIBLE - YELLOW

S - SELECTOR PEN DETECT - ORANGE

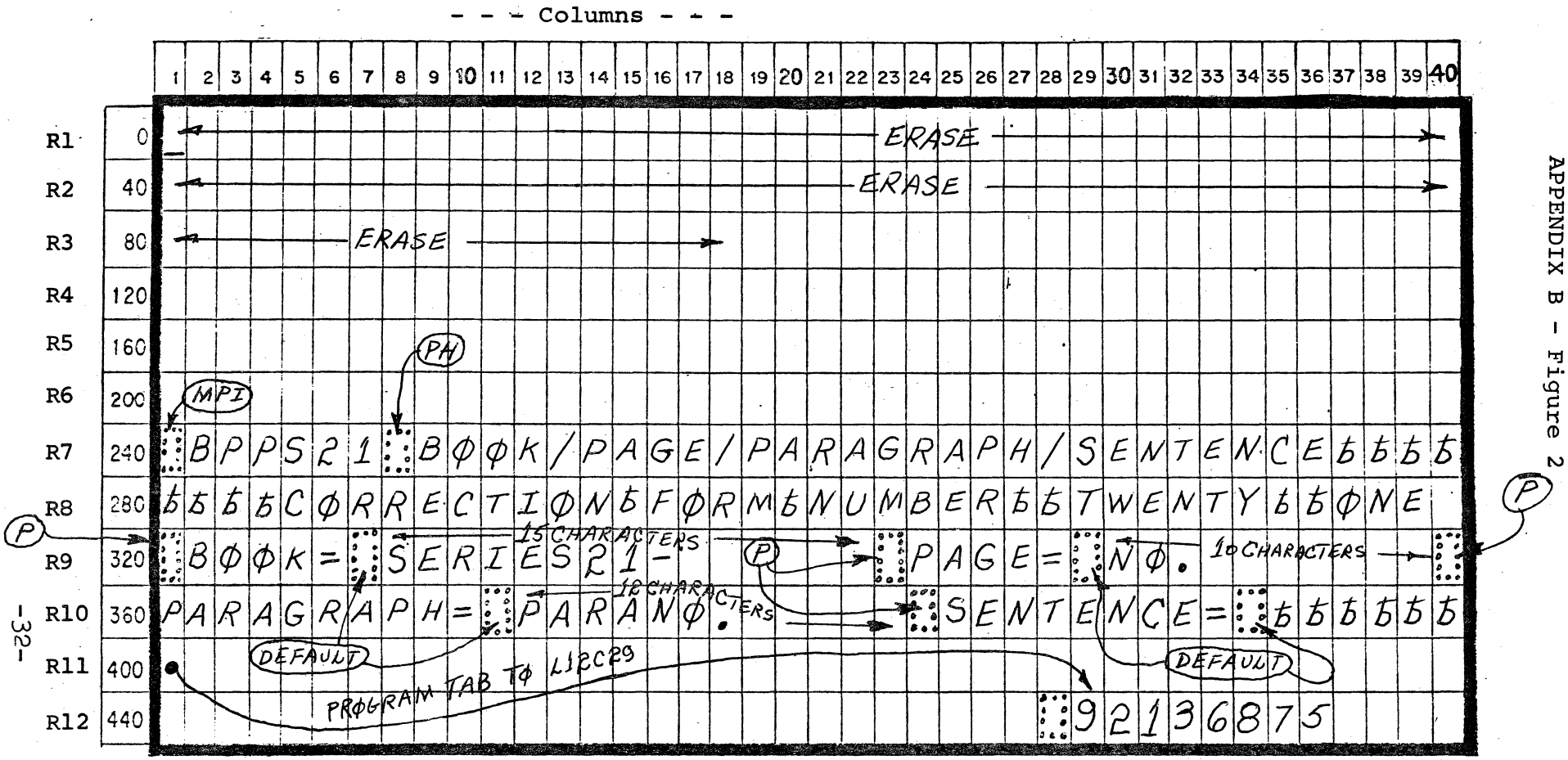
M - MODIFIED DATA TAG - GREEN

A - AUTOSKIP - PINK (P+N)

APPENDIX B - Figure 1

Screen Name BPPS21

SCREEN DESIGN LAYOUT FORM - 480 CHARACTER SCREEN



Check Appropriate Box

☒ WRITE ☐ ERASE WRITE

☒ LOCAL ☐ REMOTE

Write Control Character for Printer Operation

☐ START PRINTER

☐ HONOR NL & EM CODES

☐ 40 CHARACTER LINE

☐ 64 CHARACTER LINE

☐ 80 CHARACTER LINE

☐ RESET MODIFIED DATA TAG

Start Coding Output Stream at Location - R7C1

Insert Cursor at Location - R1C1

Write Control Character for Screen Operations

☒ RESET MODIFIED DATA TAGS

☒ RESTORE KEYBOARD AND RESET AID

☒ SOUND AUDIBLE ALARM

Color Code for Attribute Character

P - PROTECTED - BROWN

N - NUMERIC - BLUE

H - HIGH INTENSITY - RED

I - INVISIBLE - YELLOW

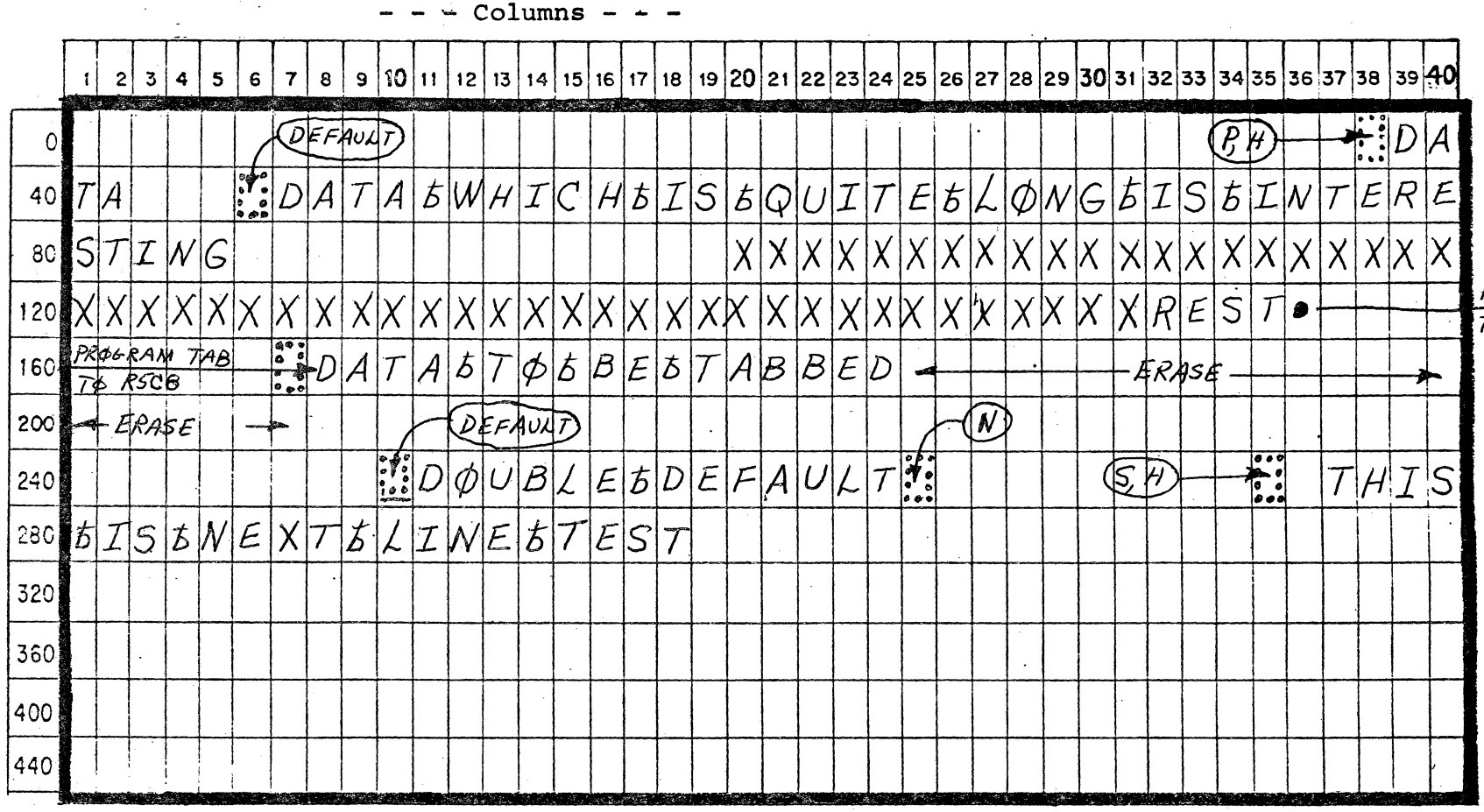
S - SELECTOR PEN DETECT - ORANGE

M - MODIFIED DATA TAG - GREEN

A - AUTOSKIP - PINK (P+N)

Screen Name TEST1

SCREEN DESIGN LAYOUT FORM - 480 CHARACTER SCREEN



Check Appropriate Box

△ WRITE ☒ ERASE WRITE

△ LOCAL ☒ REMOTE

Write Control Character for Printer Operation

△ START PRINTER

△ HONOR NL & EM CODES

△ 40 CHARACTER LINE

△ 64 CHARACTER LINE

△ 80 CHARACTER LINE

△ RESET MODIFIED DATA TAG

Start Coding Output Stream at Location - R1C38

Insert Cursor at Location - R7C10

Write Control Character for Screen Operations

☒ RESET MODIFIED DATA TAGS

☒ RESTORE KEYBOARD AND RESET AID

△ SOUND AUDIBLE ALARM

Color Code for Attribute Character

P - PROTECTED - BROWN

N - NUMERIC - BLUE

H - HIGH INTENSITY - RED

I - INVISIBLE - YELLOW

S - SELECTOR PEN DETECT - ORANGE

M - MODIFIED DATA TAG - GREEN

A - AUTOSKIP - PINK (P+N)

APPENDIX B - Figure 3

APPENDIX C, FORMAT MACRO INSTRUCTION FORMAT SUMMARY

Configuration Operands

(DEVICE=SCREEN) (DEVICE=PRINTER)

(TYPE=LOCAL) (TYPE=REMOTE)

(BUFFER=480) (BUFFER=1920)

Framing Operands (Remote Only)

(STX) (ETX)

Write and Erase Write Command Operand (Remote)

(ESC,EW, see table for codes) (ESC,W,see table for codes)

Suboperand Codes	Meaning	Comment
MDT	Reset Modified Data Tags	Should not be used with erase write.
RR	Restore Keyboard	Should not be used when sent to printer.
SP	Start Printer	
80	Print 80 Character Line	If used, avoid also using 64, 60 and NL.
64	Print 64 Character Line	If used, avoid also using 80, 40 and NL.
40	Print 40 Character Line	If used, avoid also using 80, 64 and NL.
NL	Honor NL & EM Codes	If used, avoid also using 80, 64 and 40.
A	Sound Audible Alarm	

Write and Erase Write Command Operand (Local)

(WCC,EW,see table for codes) (WCC,W, see table for codes)

Basic Operand (Local & Remote)

Table of Attribute Characters

Value Code	Attribute	Comment
P	Protected	Not to be used with a value code A.
N	Numeric	Not to be used with value code A.
A	Autoskip	A=P+N, neither P nor N should be used.

S	Selector Pen Detectable	Not to be used with I.
H	High Intensity	Not to be used with I.
M	Modified Data Tag	
I	Invisible	Not to be used with S or H.

(see table,X) Start Field Order and Attribute from Table
(,X) Default value of Start Field/Attribute
(see table,address,data) SBA, Start Field Order and Attribute from Table, Data
(see table,address,data,field length) Same as above only with Field Length
(,address,data) SBA,Default value of Start Field/Attribute, Data
(,address,data,field length) Same as above only with Field Length
(see table,,data) Omit SBA, Start Field Order and Attribute from Table, Data
(see table,X,data) Same as above.
(,,data) Omit SBA, Default value of Start Field/Attribute data
(X,,data) Omit SBA, Omit Start Field Order, Data
(X,X,data) Same as above.
(X,address,data) SBA, Omit Start Field, Data
(X,address,data,field length) Same as above only with Field length
(see table,,data,field length) Omit SBA, Start Field Order and Attribute from Table,Data with Field Length.
(see table,,,field length) Omit SBA, Start Field Order and Attribute from Table, Omit Data, Field Length.
(,,data,field length) Omit SBA, Default value of Start Field/Attribute, Data with Field Length
(,,,field length) Omit SBA, Default value of Start Field/Attribute, Omit Data, Field Length
(,X,,field length) Same as above.
(X,,,field length) Omit SBA, Omit Start Field, Omit Data, Field Length (rarely used)
(X,X,data,field length) Omit SBA, Omit Start Field, Data with Field Length
(X,,data,field length) Same as above.

Erase Unprotected to Address Operand (Local & Remote)

(EUA, SBA address,EUA address) SBA, EUA Order, EUA Address
(EUA,,EUA address) Omit SBA, EUA Order, EUA Address
(EUA,X,EUA address) Same as above.

Repeat to Address Operand (Local & Remote)

(RA,SBA address,RA address,RA character) SBA, RA Order, RA Address, RA Character
(RA,,RA address, RA character) Omit SBA, RA Order, RA Address, RA Character
(RA,X,RA address,RA character) Same as above.
(RA,SBA address,RA address,RA character,data) SBA, RA Order, RA Address, RA Character, Data
(RA,,RA address,RA character,data) Omit SBA, RA Order, RA Address, RA Character, Data
(RA,X,RA address,RA character,data) Same as above.
(RA,SBA address,RA address,RA character,data,field length) SBA, RA Order, RA Address, RA Character, Data with Field Length
(RA,,RA address,RA character,data,field length) Same as above.

Insert Cursor Operand (Local & Remote)

(IC) IC Order
(IC, SBA address) SBA, IC Order
(IC,SBA address,X) SBA, IC Order (Does not have to appear in sequence.)

Programmed Tab Operand (Local & Remote)

(PT) PT Order
(PT,PT address,data) PT Order, Data
(PT,X,data) PT Order, Data
(PT,PT address,X) PT Order

New Line & End of Medium Operands (Local & Remote)

(NL) New Line Order
(EM) End of Medium Order

Checking Operand (Local & Remote)

(CHECKING=YES)
(CHECKING=NO)

APPENDIX D - SAMPLE FORMAT MACRO INSTRUCTION EXPANSIONS

```
1491 SCRN387 FORMAT (TYPE=REMOTE),(STX),(ESC,EW,RR,A),(MPI,R1C1,'SCRN387'),X
                    (P,R2C6,'ENTER NAME '),(H,X),(IC),(P,R3C14,'ID '),(H,X),X
                    (HP,R4C40),(RA,,R6C1,'-'),(P,R7C4,'SELECT CLASS'),(PS,R8X

                     C6,'? CSDA  '),(PS,R8C20,'? BTAM WORKSHOP   '),(PS,R9C6X
                     ,'? CSDP   '),(PS,R9C20,'? QTAM WORKSHOP   '),(PS,R10C6,X
                     '? 3270 O D   '),(PS,R10C20,'? TCAM WORKSHOP   '),(ETX)

0091                 1492+SCRN387  EQU  *
0091 02              1493+         DC   X'02' STX
0092 27              1494+         DC   X'27' ESC
0093 F5              1495+         DC   X'F5' ERASE WRITE
                     1496               *,WRITE CONTROL CHARACTER IS HEX 06
0094 C6              1497+         DC   AL1(198) ADDRESS, ATTRIBUTE OR WCC
0095 11              1498+         DC   X'11' SBA
                     1499               *,SBA BUFFER ADDRESS IS 0 DECIMAL
0096 40              1500+         DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
0097 40              1501+         DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
0098 1D              1502+         DC   X'1D' SF
                     1503               *,ATTRIBUTE CHARACTER IS HEX 2D
0099 6D              1504+         DC   AL1(109) ADDRESS, ATTRIBUTE OR WCC
009A E2C3D9D5F3F8F7  1505+         DC   CL7'SCRN387'
00A1 11              1506+         DC   X'11' SBA
                     1507               *,SBA BUFFER ADDRESS IS 45 DECIMAL
00A2 40              1508+         DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
00A3 6D              1509+         DC   AL1(109) ADDRESS, ATTRIBUTE OR WCC
00A4 1D              1510+         DC   X'1D' SF
                     1511               *,ATTRIBUTE CHARACTER IS HEX 20
00A5 60              1512+         DC   AL1(96) ADDRESS, ATTRIBUTE OR WCC
00A6 C5D5E3C5D940D5C1 1513+        DC   CL11'ENTER NAME '
                     1514               *,MNOTE 55, SBA SEQUENCE AT 57, OMITTED
00B1 1D              1515+         DC   X'1D' SF
                     1516               *,ATTRIBUTE CHARACTER IS HEX 08
00B2 C8              1517+         DC   AL1(200) ADDRESS, ATTRIBUTE OR WCC
00B3 13              1518+         DC   X'13' IC
00B4 11              1519+         DC   X'11' SBA
                     1520               *,SBA BUFFER ADDRESS IS 93 DECIMAL
```

```
00B5 C1                   1521+         DC   AL1(193) ADDRESS, ATTRIBUTE OR WCC
00B6 5D                   1522+         DC   AL1(93) ADDRESS, ATTRIBUTE OR WCC
00B7 1D                   1523+         DC   X'1D' SF
                          1524               *,ATTRIBUTE CHARACTER IS HEX 20
00B8 60                   1525+         DC   AL1(96) ADDRESS, ATTRIBUTE OR WCC
00B9 C9C47A               1526+         DC   CL3'ID '
                          1527               *,MNOTE 55, SBA SEQUENCE AT 97, OMITTED
00BC 1D                   1528+         DC   X'1D' SF
                          1529               *,ATTRIBUTE CHARACTER IS HEX 08
00BD C8                   1530+         DC   AL1(200) ADDRESS, ATTRIBUTE OR WCC
00BE 11                   1531+         DC   X'11' SBA
                          1532               *,SBA BUFFER ADDRESS IS 159 DECIMAL
00BF C2                   1533+         DC   AL1(194) ADDRESS, ATTRIBUTE OR WCC
00C0 5F                   1534+         DC   AL1(95) ADDRESS, ATTRIBUTE OR WCC
00C1 1D                   1535+         DC   X'1D' SF
                          1536               *,ATTRIBUTE CHARACTER IS HEX 28
00C2 E8                   1537+         DC   AL1(232) ADDRESS, ATTRIBUTE OR WCC
                          1538               *,MNOTE 55, SBA SEQUENCE AT 160, OMITTED
00C3 3C                   1539+         DC   X'3C' RA
                          1540               *,RA BUFFER ADDRESS IS 200 DECIMAL
00C4 C3                   1541+         DC   AL1(195) ADDRESS, ATTRIBUTE OR WCC
00C5 C8                   1542+         DC   AL1(200) ADDRESS, ATTRIBUTE OR WCC
00C6 60                   1543+         DC   CL1'-'
00C7 11                   1544+         DC   X'11' SBA
                          1545               *,SBA BUFFER ADDRESS IS 243 DECIMAL
00C8 C3                   1546+         DC   AL1(195) ADDRESS, ATTRIBUTE OR WCC
00C9 F3                   1547+         DC   AL1(243) ADDRESS, ATTRIBUTE OR WCC
00CA 1D                   1548+         DC   X'1D' SF
                          1549               *,ATTRIBUTE CHARACTER IS HEX 20
00CB 60                   1550+         DC   AL1(96) ADDRESS, ATTRIBUTE OR WCC
00CC E2C5D3C5C3E340C3     1551+         DC   CL12'SELECT CLASS'
00D8 11                   1552+         DC   X'11' SBA
                          1553               *,SBA BUFFER ADDRESS IS 285 DECIMAL
00D9 C4                   1554+         DC   AL1(196) ADDRESS, ATTRIBUTE OR WCC
00DA 5D                   1555+         DC   AL1(93) ADDRESS, ATTRIBUTE OR WCC
00DB 1D                   1556+         DC   X'1D' SF
                          1557               *,ATTRIBUTE CHARACTER IS HEX 24
00DC E4                   1558+         DC   AL1(228) ADDRESS, ATTRIBUTE OR WCC
00DD 6F40C3E2C4C14040     1559+         DC   CL9'? CSDA   '
00E6 11                   1560+         DC   X'11' SBA
                          1561               *,SBA BUFFER ADDRESS IS 299 DECIMAL
00E7 C4                   1562+         DC   AL1(196) ADDRESS, ATTRIBUTE OR WCC
```

```
00E8 6B                   1563+          DC   AL1(107) ADDRESS, ATTRIBUTE OR WCC
00E9 1D                   1564+          DC   X'1D' SF
                          1565                *,ATTRIBUTE CHARACTER IS HEX 24
00EA E4                   1566+          DC   AL1(228) ADDRESS, ATTRIBUTE OR WCC
00EB 6F40C2E3C1D440E6     1567+          DC   CL18'? BTAM WORKSHOP   '
00FD 11                   1568+          DC   X'11' SBA
                          1569                *,SBA BUFFER ADDRESS IS 325 DECIMAL
00FE C5                   1570+          DC   AL1(197) ADDRESS, ATTRIBUTE OR WCC
00FF C5                   1571+          DC   AL1(197) ADDRESS, ATTRIBUTE OR WCC
0100 1D                   1572+          DC   X'1D' SF
                          1573                *,ATTRIBUTE CHARACTER IS HEX 24
0101 E4                   1574+          DC   AL1(228) ADDRESS, ATTRIBUTE OR WCC
0102 6F40C3E2C4D74040     1575+          DC   CL9'? CSDP   '
010B 11                   1576+          DC   X'11' SBA
                          1577                *,SBA BUFFER ADDRESS IS 339 DECIMAL
010C C5                   1578+          DC   AL1(197) ADDRESS, ATTRIBUTE OR WCC
010D D3                   1579+          DC   AL1(211) ADDRESS, ATTRIBUTE OR WCC
010E 1D                   1580+          DC   X'1D' SF
                          1581                *,ATTRIBUTE CHARACTER IS HEX 24
010F E4                   1582+          DC   AL1(228) ADDRESS, ATTRIBUTE OR WCC
0110 6F40D8E3C1D440E6     1583+          DC   CL18'? QTAM WORKSHOP   '
0122 11                   1584+          DC   X'11' SBA
                          1585                *,SBA BUFFER ADDRESS IS 365 DECIMAL
0123 C5                   1586+          DC   AL1(197) ADDRESS, ATTRIBUTE OR WCC
0124 6D                   1587+          DC   AL1(109) ADDRESS, ATTRIBUTE OR WCC
0125 1D                   1588+          DC   X'1D' SF
                          1589                *,ATTRIBUTE CHARACTER IS HEX 24
0126 E4                   1590+          DC   AL1(228) ADDRESS, ATTRIBUTE OR WCC
0127 6F40F3F2F7F040D6     1591+          DC   CL13'? 3270 0 D   '
0134 11                   1592+          DC   X'11' SBA
                          1593                *,SBA BUFFER ADDRESS IS 379 DECIMAL
0135 C5                   1594+          DC   AL1(197) ADDRESS, ATTRIBUTE OR WCC
0136 7B                   1595+          DC   AL1(123) ADDRESS, ATTRIBUTE OR WCC
0137 1D                   1596+          DC   X'1D' SF
                          1597                *,ATTRIBUTE CHARACTER IS HEX 24
0138 E4                   1598+          DC   AL1(228) ADDRESS, ATTRIBUTE OR WCC
0139 6F40E3C3C1D440E6     1599+          DC   CL18'? TCAM WORKSHOP   '
014B 03                   1600+          DC   X'03' ETX
                          1601                *,STARTING ADDRESS 0, ENDING ADDRESS 397
                          1602                *,19 OPERANDS PROCESSED
                          1603                *,TOTAL CORE FOR FORMAT SCRN387 WAS 187 BYTES
```

```
                1604 BPPS21   FORMAT (TYPE=LOCAL),(WCC,W,MDT,RR,A),(MPI,R7C1,'BPPS21'),     X
                                    (PH,,'BOOK/PAGE/PARAGRAPH/SENTENCE        CORRECTION FORX
                                    M NUMBER  TWENTY  ONE'),(P,R9C1,'BOOK='),(,,'SERIES21-',X
                                    15),(P,,'PAGE='),(,,'NO.',10),(P,,'PARAGRAPH='),(,,'PARAX
                                    NO.',12),(P,,'SENTENCE='),(,,'      '),(PT,R12C29,'92136X
                                    875'),(EUA,R1C1,R3C19),(IC,R1C1,X)

014C                    1605+BPPS21   EQU  *
                        1606               *,WRITE CONTROL CHARACTER IS HEX 07
014C C7                 1607+         DC   AL1(199) ADDRESS, ATTRIBUTE OR WCC
014D 11                 1608+         DC   X'11' SBA
                        1609               *,SBA BUFFER ADDRESS IS 240 DECIMAL
014E C3                 1610+         DC   AL1(195) ADDRESS, ATTRIBUTE OR WCC
014F F0                 1611+         DC   AL1(240) ADDRESS, ATTRIBUTE OR WCC
0150 1D                 1612+         DC   X'1D' SF
                        1613               *,ATTRIBUTE CHARACTER IS HEX 2D
0151 6D                 1614+         DC   AL1(109) ADDRESS, ATTRIBUTE OR WCC
0152 C2D7D7E2F2F1       1615+         DC   CL6'BPPS21'
                        1616               *,MNOTE 55, SBA SEQUENCE AT 247, OMITTED
0158 1D                 1617+         DC   X'1D' SF
                        1618               *,ATTRIBUTE CHARACTER IS HEX 28
0159 E8                 1619+         DC   AL1(232) ADDRESS, ATTRIBUTE OR WCC
                        1620+         DC   CL71'BOOK/PAGE/PARAGRAPH/SENTENCE   CORRECTION FORMX
015A C2D6D6D261D7C1C7      +                NUMBER  TWENTY  ONE'
01A1 11                 1621+         DC   X'11' SBA
                        1622               *,SBA BUFFER ADDRESS IS 320 DECIMAL
01A2 C5                 1623+         DC   AL1(197) ADDRESS, ATTRIBUTE OR WCC
01A3 40                 1624+         DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
01A4 1D                 1625+         DC   X'1D' SF
                        1626               *,ATTRIBUTE CHARACTER IS HEX 20
01A5 60                 1627+         DC   AL1(96) ADDRESS, ATTRIBUTE OR WCC
01A6 C2D6D6D27E         1628+         DC   CL5'BOOK='
                        1629               *,MNOTE 55, SBA SEQUENCE AT 326, OMITTED
01AB 1D                 1630+         DC   X'1D' SF
                        1631               *,ATTRIBUTE CHARACTER IS HEX 00
01AC 40                 1632+         DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
01AD E2C5D9C9C5E2F2F1   1633+         DC   CL9'SERIES21-'
01B6 11                 1634+         DC   X'11' SBA
                        1635               *,SBA BUFFER ADDRESS IS 342 DECIMAL
```

```
01B7 C5                  1636+         DC   AL1(197) ADDRESS, ATTRIBUTE OR WCC
01B8 D6                  1637+         DC   AL1(214) ADDRESS, ATTRIBUTE OR WCC
01B9 1D                  1638+         DC   X'1D' SF
                         1639               *,ATTRIBUTE CHARACTER IS HEX 20
01BA 60                  1640+         DC   AL1(96) ADDRESS, ATTRIBUTE OR WCC
01BB D7C1C7C57E          1641+         DC   CL5'PAGE='
                         1642               *,MNOTE 55, SBA SEQUENCE AT 348, OMITTED
01C0 1D                  1643+         DC   X'1D' SF
                         1644               *,ATTRIBUTE CHARACTER IS HEX 00
01C1 40                  1645+         DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
01C2 D5D64B              1646+         DC   CL3'NO.'
01C5 11                  1647+         DC   X'11' SBA
                         1648               *,SBA BUFFER ADDRESS IS 359 DECIMAL
01C6 C5                  1649+         DC   AL1(197) ADDRESS, ATTRIBUTE OR WCC
01C7 E7                  1650+         DC   AL1(231) ADDRESS, ATTRIBUTE OR WCC
01C8 1D                  1651+         DC   X'1D' SF
                         1652               *,ATTRIBUTE CHARACTER IS HEX 20
01C9 60                  1653+         DC   AL1(96) ADDRESS, ATTRIBUTE OR WCC
01CA D7C1D9C1C7D9C1D7    1654+         DC   CL10'PARAGRAPH='
                         1655               *,MNOTE 55, SBA SEQUENCE AT 370, OMITTED
01D4 1D                  1656+         DC   X'1D' SF
                         1657               *,ATTRIBUTE CHARACTER IS HEX 00
01D5 40                  1658+         DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
01D6 D7C1D9C1D5D64B      1659+         DC   CL7'PARANO.'
01DD 11                  1660+         DC   X'11' SBA
                         1661               *,SBA BUFFER ADDRESS IS 383 DECIMAL
01DE C5                  1662+         DC   AL1(197) ADDRESS, ATTRIBUTE OR WCC
01DF 7F                  1663+         DC   AL1(127) ADDRESS, ATTRIBUTE OR WCC
01E0 1D                  1664+         DC   X'1D' SF
                         1665               *,ATTRIBUTE CHARACTER IS HEX 20
01E1 60                  1666+         DC   AL1(96) ADDRESS, ATTRIBUTE OR WCC
01E2 E2C5D5E3C5D5C3C5    1667+         DC   CL9'SENTENCE='
                         1668               *,MNOTE 55, SBA SEQUENCE AT 393, OMITTED
01EB 1D                  1669+         DC   X'1D' SF
                         1670               *,ATTRIBUTE CHARACTER IS HEX 00
01EC 40                  1671+         DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
01ED 404040404040        1672+         DC   CL6'      '
01F3 05                  1673+         DC   X'05' PT
01F4 F9F2F1F3F6F8F7F5    1674+         DC   CL8'92136875'
01FC 11                  1675+         DC   X'11' SBA
                         1676               *,SBA BUFFER ADDRESS IS 0 DECIMAL
```

```
01FD 40                 1677+          DC    AL1(64) ADDRESS, ATTRIBUTE OR WCC
01FE 40                 1678+          DC    AL1(64) ADDRESS, ATTRIBUTE OR WCC
01FF 12                 1679+          DC    X'12' EUA
                        1680                 *,EUA BUFFER ADDRESS IS 98 DECIMAL
0200 C1                 1681+          DC    AL1(193) ADDRESS, ATTRIBUTE OR WCC
0201 E2                 1682+          DC    AL1(226) ADDRESS, ATTRIBUTE OR WCC
0202 11                 1683+          DC    X'11' SBA
                        1684                 *,SBA BUFFER ADDRESS IS 0 DECIMAL
0203 40                 1685+          DC    AL1(64) ADDRESS, ATTRIBUTE OR WCC
0204 40                 1686+          DC    AL1(64) ADDRESS, ATTRIBUTE OR WCC
0205 13                 1687+          DC    X'13' IC
                        1688                 *,STARTING ADDRESS 240, ENDING ADDRESS 97
                        1689                 *,WRAP OCCURRED
                        1690                 *,15 OPERANDS PROCESSED
                        1691                 *,TOTAL CORE FOR FORMAT BPPS21 WAS 186 BYTES
```

To illustrate a badly coded screen, consider TEST1, in Figure 3 of APPENDIX B. An erase write causes mnote 59, 57 and 61. The selector pen detect field causes mnotes 56 and 51. Note that the mnotes pinpoint both the operand and sub-operand in the case of mnotes 59 and 56.

The Programmed Tab operation assumes that a previously defined unprotected field attribute character has been defined at R5C7. The erase write mnote #57, Statement #1450, alerts the programmer that this command will have wiped out the assumed attribute character.

Mnote 61 at Statement 1453 warns the Programmer of the possible error in using an erase unprotected to address sequence when an erase write has been specified. Mnote 59 at Statement 1420 is the first programmer warning that he has incorrectly specified an EW when he really wants W in operand #3. By referring to mnotes 59, 57 and 61, the Programmer should conclude that a write is what he had in mind.

Statements 1467 and 1472 illustrate the omission of two SBA sequences and the documentation by mnote 55. The final address referred to in Statement 1488 can be calculated by adding the 23 bytes of Statement 1483 to the SBA address at Statement 1477, i.e., 274 + 23 = 297.

The total core of 143 bytes can be checked by subtracting the STX address, 002 HEX from the ETX address, 090 HEX, and adding 1, i.e., (090-002) + 1 = 08D + 1 = 08E HEX = 143 Decimal.

```
                1415 TEST1   FORMAT (TYPE=REMOTE),(STX),(ESC,EW,RR,MDT),(PH,R1C38,'DATA'), X
                                    (,R2C6,'DATA WHICH IS QUITE LONG IS INTERESTING'),(RA,1OX
                                    O,R4C32,X,'REST',10),(PT,R5C8,'DATA TO BE TABBED'),(EUA,X
                                    ,R6C8),(IC,R7C10),(,,'DOUBLE DEFAULT'),(N,),(SH,R7C35,' X
                                    THIS IS NEXT LINE TEST'),(ETX)

0002                1416+TEST1   EQU  *
0002 02             1417+        DC   X'02' STX
0003 27             1418+        DC   X'27' ESC
0004 F5             1419+        DC   X'F5' ERASE WRITE
                    1420              MNOTE 59, OPERAND NO. 3, SUBOPERAND 4, YOU HAVE
                                      SPECIFIX ED RESET MODIFIED DATA TAGS ON AN ERASE
                                      WRITE
                    1421+*            OPERATION, WARNING
                    1422              *,WRITE CONTROL CHARACTER IS HEX 03
0005 C3             1423+        DC   AL1(195) ADDRESS, ATTRIBUTE OR WCC
0006 11             1424+        DC   X'11' SBA
                    1425              *,SBA BUFFER ADDRESS IS 37 DECIMAL
0007 40             1426+        DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
0008 E5             1427+        DC   AL1(229) ADDRESS, ATTRIBUTE OR WCC
0009 1D             1428+        DC   X'1D' SF
                    1429              *,ATTRIBUTE CHARACTER IS HEX 28
000A E8             1430+        DC   AL1(232) ADDRESS, ATTRIBUTE OR WCC
000B C4C1E3C1       1431+        DC   CL4'DATA'
000F 11             1432+        DC   X'11' SBA
                    1433              *,SBA BUFFER ADDRESS IS 45 DECIMAL
0010 40             1434+        DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
0011 6D             1435+        DC   AL1(109) ADDRESS, ATTRIBUTE OR WCC
0012 1D             1436+        DC   X'1D' SF
                    1437              *,ATTRIBUTE CHARACTER IS HEX 00
0013 40             1438+        DC   AL1(64) ADDRESS, ATTRIBUTE OR WCC
0014 C4C1E3C140E6C8C9 1439+      DC   CL39'DATA WHICH IS QUITE LONG IS INTERESTING'
003B 11             1440+        DC   X'11' SBA
                    1441              *,SBA BUFFER ADDRESS IS 100 DECIMAL
003C C1             1442+        DC   AL1(193) ADDRESS, ATTRIBUTE OR WCC
003D E4             1443+        DC   AL1(228) ADDRESS, ATTRIBUTE OR WCC
003E 3C             1444+        DC   X'3C' RA
                    1445              *,RA BUFFER ADDRESS IS 151 DECIMAL
003F C2             1446+        DC   AL1(194) ADDRESS, ATTRIBUTE OR WCC
0040 D7             1447+        DC   AL1(215) ADDRESS, ATTRIBUTE OR WCC
```

```
0041 E7                   1448 +        DC    CL1'X'
0042 D9C5E2E3             1449 +        DC    CL4'REST'
                          1450                MNOTE 57, OPERAND NO.7, A PT HAS BEEN SPECIFIED
                                              WITHX AN ERASE WRITE, WARNING
0046 05                   1451 +        DC    X'05' PT
0047 C4C1E3C140E3D640     1452 +        DC    CL17'DATA TO BE TABBED'
                          1453                MNOTE 61, OPERAND NO. 8, AN EUA HAS BEEN
                                              SPECIFIED WITHX AN ERASE WRITE, WARNING
0058 11                   1454 +        DC    X'11' SBA
                          1455                *,SBA BUFFER ADDRESS IS 185 DECIMAL
0059 C2                   1456 +        DC    AL1(194) ADDRESS, ATTRIBUTE OR WCC
005A F9                   1457 +        DC    AL1(249) ADDRESS, ATTRIBUTE OR WCC
005B 12                   1458 +        DC    X'12' EUA
                          1459                *,EUA BUFFER ADDRESS IS 207 DECIMAL
005C C3                   1460 +        DC    AL1(195) ADDRESS, ATTRIBUTE OR WCC
005D 4F                   1461 +        DC    AL1(79) ADDRESS, ATTRIBUTE OR WCC
005E 11                   1462 +        DC    X'11' SBA
                          1463                *,SBA BUFFER ADDRESS IS 249 DECIMAL
005F C3                   1464 +        DC    AL1(195) ADDRESS, ATTRIBUTE OR WCC
0060 F9                   1465 +        DC    AL1(249) ADDRESS, ATTRIBUTE OR WCC
0061 13                   1466 +        DC    X'13' IC
                          1467                *,MNOTE 55, SBA SEQUENCE AT 249, OMITTED
0062 1D                   1468 +        DC    X'1D' SF
                          1469                *,ATTRIBUTE CHARACTER IS HEX 00
0063 40                   1470 +        DC    AL1(64) ADDRESS, ATTRIBUTE OR WCC
0064 C4D6E4C2D3C540C4     1471 +        DC    CL14'DOUBLE DEFAULT'
                          1472                *,MNOTE 55, SBA SEQUENCE AT 264, OMITTED
0072 1D                   1473 +        DC    X'1D' SF
                          1474                *,ATTRIBUTE CHARACTER IS HEX 10
0073 50                   1475 +        DC    AL1(80) ADDRESS, ATTRIBUTE OR WCC
0074 11                   1476 +        DC    X'11' SBA
                          1477                *,SBA BUFFER ADDRESS IS 274 DECIMAL
0075 C4                   1478 +        DC    AL1(196) ADDRESS, ATTRIBUTE OR WCC
0076 D2                   1479 +        DC    AL1(210) ADDRESS, ATTRIBUTE OR WCC
0077 1D                   1480 +        DC    X'1D' SF
                          1481                *,ATTRIBUTE CHARACTER IS HEX 08
0078 C8                   1482 +        DC    AL1(200) ADDRESS, ATTRIBUTE OR WCC
0079 40E3C8C9E240C9E2     1483 +        DC    CL23' THIS IS NEXT LINE TEST'
                          1484                MNOTE 56, OPERAND 12, SUBOPERAND 3, SELECTOR
                                              PEN DETECTX FIELD DOES NOT CONTAIN BLANKS OR
                                              NULLS IN LAST
```

```
                    1485+*              THREE POSITIONS, WARNING
                    1486                MNOTE 51, OPERAND NO. 12, SELECTOR PEN
                                        DETECT DATA  FIEX LD EXTENDS FROM ONE ROW
                                        INTO NEXT, WARNING
0090 03             1487+          DC   X'03' ETX
                    1488                *,STARTING ADDRESS 37, ENDING ADDRESS 297
                    1489                *,13 OPERANDS PROCESSED
                    1490                *,TOTAL CORE FOR FORMAT TEST1 WAS 143 BYTES
```

APPENDIX E, FORMAT MACRO INSTRUCTION MNOTE MESSAGES

Numbered Mnotes

MNOTE 1, NAME FIELD ON FORMAT MACRO MISSING, MACRO ABORT

MNOTE 2, ZERO OPERANDS CODED, MACRO ABORT

MNOTE 3, OPERAND NO. ___, IS NOT A SUBLIST, I.E., IN PARENTHESIS, MACRO ABORT

MNOTE 4, OPERAND NO. ___, SPELLING ERROR DETECTED, MACRO ABORT

MNOTE 5, OPERAND NO. ___, SUBOPERAND NO. ___, INVALID ATTRIBUTE CHARACTER CODED NOT PNHISM OR A, MACRO ABORT

MNOTE 6, OPERAND NO. ___, SUBOPERAND NO. ___, BOTH S AND I CODED IN ATTRIBUTE CHARACTER, MACRO ABORT

MNOTE 7, OPERAND NO. ___, SUBOPERAND NO. ___, BOTH H AND I CODED IN ATTRIBUTE CHARACTER, MACRO ABORT

MNOTE 8, OPERAND NO. ___, SUBOPERAND NO. ___, BOTH P AND N CODED IN ATTRIBUTE CHARACTER, USE A, MACRO ABORT

MNOTE 9, OPERAND NO. ___, HAS AN A AND A P IN THE ATTRIBUTE CHARACTER, MACRO ABORT

MNOTE 10, OPERAND NO. ___, SUBOPERAND NO. ___, THE SUBOPERAND HAS GREATER THAN 6 CHARACTERS FOR A SET BUFFER ADDRESS SEQUENCE, MACRO ABORT

MNOTE 11, OPERAND NO. ___, SUBOPERAND NO. ___, ALL DIGITS ARE NOT NUMERIC FOR SCREEN LOCATION, OR ARE NOT WITHIN AN ACCEPTABLE RANGE OF NUMERIC VALUES FOR THE SCREEN SIZE SPECIFIED, MACRO ABORT

MNOTE 12, OPERAND NO. ___, SUBOPERAND NO. ___, WHEN UTILIZING ROW-AND-COLUMN NOTATION EITHER THE 3RD OR 4TH CHARACTER MUST BE A C

MNOTE 13, OPERAND NO. ___, SUBOPERAND NO. ___, FOR A SCREEN SIZE OF 1920, THE ROW NUMBER MUST BE, IF TWO POSITIONS LONG, GREATER THAN 9 AND LESS THAN 25, OR IF A 480 SIZE SCREEN GREATER THAN 9 AND LESS THAN 13, ABORT

MNOTE 14, OPERAND NO. ___, SUBOPERAND NO. ___, FOR A SCREEN SIZE OF 1920, THE COLUMN NUMBER MUST BE GREATER THAN OR EQUAL TO 1 AND LESS THAN 81 OR IF A 480 SIZE SCREEN GREATER THAN OR EQUAL TO ONE AND LESS THAN 41 MACRO ABORT

MNOTE 15, OPERAND NO. ___, SUBOPERAND NO. ___, IF A SINGLE POSITION ROW NUMBER IS SPECIFIED, IT MUST BE GREATER THAN 0 AND LESS THAN 10, MACRO ABORT

MNOTE 16, OPERAND NO. ___, SUBOPERAND NO. ___, THE SCREEN SIZE OF 480 OR 1920 HAS BEEN EXCEEDED, THE MAXIMUM VALUE AS CALCULATED MUST BE 479 OR 1919, ABORT

MNOTE 17, OPERAND NO. ___, SUBOPERAND NO. ___, THE CALCULATED ADDRESS IS EQUAL TO THE PREVIOUS BUFFER LOCATION, OPERANDS MUST NOT OVERLAP, MACRO ABORT

MNOTE 18, OPERAND NO. ___, SUBOPERAND NO. ___, AN ILLEGAL WRAP OPERATION HAS OCCURRED IN THAT THE ORIGINAL STARTING SCREEN LOCATION FOR THIS SCREEN HAS NOW BEEN REACHED AFTER WRAPING THE SCREEN ONCE, MACRO ABORT

MNOTE 19, OPERAND NO. ___, SUBOPERAND NO. ___, YOU HAVE ALREADY WRAPPED ONCE AND ARE NOW ATTEMPTING TO WRAP AGAIN, MACRO ABORT

MNOTE 20, OPERAND NO. ___, SUBOPERAND ___, A WRAP OPERATION HAS JUST OCCURRED BUT HAS EXCEEDED THE START ADDRESS FOR THIS SCREEN, MACRO ABORT

MNOTE 21, OPERAND NO. ___, THE 3RD SUBOPERAND HAS BEEN OMITTED WHICH REQUIRES A 4TH SUBOPERAND, MACRO ABORT

MNOTE 22, OPERAND NO. ___, SUBOPERAND ___, THE 3RD SUBOPERAND CONTAINING DATA MUST BEGIN AND END IN SINGLE QUOTATION MARKS, MACRO ABORT

MNOTE 23, OPERAND NO. ___, SUBOPERAND ___, YOU HAVE OMITTED A SET BUFFER ADDRESS SEQUENCE IN THE SUBOPERAND AND HAVE NOT SPECIFIED AN ERASE WRITE. A WRITE MUST BEGIN WITH A SBA SEQUENCE, MACRO ABORT.

MNOTE 24, OPERAND NO. ___, YOU HAVE SPECIFIED SELECTOR PEN DETECT BUT HAVE OMITTED THE 3RD SUBOPERAND, MACRO ABORT

MNOTE 25, OPERAND NO. ___, THE 4TH SUBOPERAND DOES NOT HAVE A DECIMAL NUMBER IN IT, MACRO ABORT

MNOTE 26, OPERAND ___, SUBOPERAND ___, THE FIRST CHARACTER OF A SELECTOR PEN DETECT FIELD MUST CONTAIN A BLANK, GREATER THAN, NULL OR QUESTION MARK, MACRO ABORT

MNOTE 27, OPERAND ___, SUBOPERAND ___, SELECTOR PEN DETECT FIELD IS NOT 2 CHARACTERS WIDE, MACRO ABORT

MNOTE 28, OPERAND ___, SUBOPERAND ___, SELECTOR PEN DETECT FIELD MAY NOT BE LARGE ENOUGH, WARNING

MNOTE 29, OPERAND ___, SUBOPERAND ___, AN ERASE WRITE WAS NOT SPECIFIED YET A 4TH SUBOPERAND WAS CODED GIVING FIELD LENGTH FOR A SELECTOR PEN DETECT FIELD, WARNING

MNOTE 30, OPERAND ___ WAS CODED RA AND HAS LESS THAN 4 SUBOPERANDS, MACRO ABORT

MNOTE 31, OPERAND ___, WAS CODED RA AND HAS AN OMITTED 3RD SUBOPERAND, MACRO ABORT

MNOTE 32, OPERAND ___, WAS CODED RA AND HAS AN OMITTED 4TH SUBOPERAND, MACRO ABORT

MNOTE 33, OPERAND ___, SUBOPERAND ___, REQUIRES A SINGLE CHARACTER, MACRO ABORT

MNOTE 34, OPERAND NO. ___, AN EUA OPERAND MUST HAVE THREE SUBOPERANDS, MACRO ABORT

MNOTE 35, OPERAND NO. ___, AN EUA OPERAND MUST NOT HAVE AN OMITTED 3RD SUBOPERAND, MACRO ABORT

MNOTE 36, OPERAND NO. ___, A PT OPERAND MUST HAVE 3 SUBOPERANDS, MACRO ABORT

MNOTE 37, OPERAND NO. ___, A PT OPERAND CANNOT HAVE A SECOND SUBOPERAND MISSING, MACRO ABORT

MNOTE 38, OPERAND NO. ___, A PT OPERAND CANNOT HAVE A THIRD SUBOPERAND MISSING, MACRO ABORT

MNOTE 39, OPERAND NO. ___, AN EM HAS BEEN SPECIFIED AS A SINGLE SUBOPERAND WITHOUT SPECIFYING A PRINTER AS A DEVICE, WARNING

MNOTE 40, OPERAND NO. ___, A NL HAS BEEN SPECIFIED AS A SINGLE SUBOPERAND WITHOUT SPECIFYING A PRINTER AS A DEVICE, WARNING

MNOTE 41, OPERAND NO. ___, AN IC HAS BEEN SPECIFIED AS A SUB-OPERAND AND IS MISSING A REQUIRED 2ND SUBOPERAND, MACRO ABORT

MNOTE 42, OPERAND NO. ___, A STX HAS BEEN CODED AS A SINGLE SUBOPERAND YET A REMOTE DEVICE WAS NOT SPECIFIED MACRO ABORT

MNOTE 43, OPERAND NO. ___, AN ETX HAS BEEN CODED AS A SINGLE SUBOPERAND YET A REMOTE DEVICE WAS NOT SPECIFIED, MACRO ABORT

MNOTE 44, OPERAND NO. ___,AN ESC HAS BEEN CODED YET A REMOTE DEVICE WAS NOT SPECIFIED, MACRO ABORT

MNOTE 45, OPERAND NO. ___, SUBOPERAND ___, AN ESC OPERAND RE-QUIRES AN E OR EW IN THE SECOND SUBOPERAND LOCATION, MACRO ABORT

MNOTE 46, OPERAND NO. ___, AN ESC OPERAND REQUIRES 3 OR MORE SUBOPERANDS, MACRO ABORT

MNOTE 47, OPERAND NO. ___, SUBOPERAND ___, AN INVALID SUB-OPERAND, OTHER THAN A, NL, 40, 64, 80, SP, RR OR MDT HAS BEEN DETECTED, MACRO ABORT

MNOTE 48, OPERAND NO. ___, SUBOPERAND ___, A DUPLICATE PRINTER CODE HAS BEEN SPECIFIED, I.E., IF A 40 HAS ALREADY BEEN SET, A 64, NL, OR 80 HAS BEEN DETECTED MACRO ABORT

MNOTE 49, OPERAND NO. ___, CODED WCC YET A REMOTE CONFIGURATION WAS SPECIFIED, MACRO ABORT

MNOTE 50, OPERAND NO. ___, SUBOPERAND ___, WCC CODED AND DOES NOT HAVE AN EW OR W IN THE 2ND SUBOPERAND, MACRO ABORT

MNOTE 51, OPERAND NO. ___, SELECTOR PEN DETECT DATA FIELD EXTENDS FROM ONE ROW INTO NEXT, WARNING

MNOTE 52, OPERAND NO. ___, A GREATER THAN HAS BEEN SPECIFIED AS THE FIRST CHARACTER IN A SELECTOR PEN DETECT FIELD YET NO MODIFIED DATA TAG HAS BEEN SET IN THE ATTRIBUTE CHARACTER, WARNING

MNOTE 53, OPERAND NO. ___, A GREATER THAN HAS NOT BEEN SET IN THE FIRST CHARACTER OF A SELECTOR PEN DETECT FIELD YET THE MODIFIED DATA TAG HAS BEEN SET, WARNING

MNOTE 54, OPERAND ___, AN IC OPERAND, HAS BEEN CODED WITH 3 SUBOPERANDS YET THE FIRST CHARACTER OF THE 3RD SUBOPERAND IS NOT AN X, MACRO ABORT

MNOTE 55, SBA SEQUENCE AT ___, OMITTED

MNOTE 56, OPERAND ___, SUBOPERAND ___, SELECTOR PEN DETECT FIELD DOES NOT CONTAIN BLANKS OR NULLS IN LAST THREE POSITIONS, WARNING

MNOTE 57, OPERAND NO. ___, A PT HAS BEEN SPECIFIED WITH AN ERASE WRITE, WARNING

MNOTE 58, OPERAND NO. ___, SUBOPERAND NO. ___, HAS BOTH AN A AND N IN THE ATTRIBUTE CHARACTER, MACRO ABORT

MNOTE 59, OPERAND NO. ___, SUBOPERAND NO. ___, HAS SPECIFIED RESET MODIFIED DATA TAGS ON AN ERASE WRITE OPERATION, WARNING

MNOTE 60, OPERAND NO. ___, SUBOPERAND NO. ___, HAS SPECIFIED A RESTORE KEYBOARD FOR A PRINTER DEVICE, WARNING

MNOTE 61, OPERAND NO. ___, AN EUA HAS BEEN SPECIFIED WITH AN ERASE WRITE, WARNING

Unnumbered Mnotes

MNOTE FOR ATTRIBUTE CHARACTER

*,ATTRIBUTE CHARACTER IS HEX XX

MNOTE FOR WRITE CONTROL CHARACTER

```
*,WRITE CONTROL CHARACTER IS HEX XX
MNOTE FOR SBA BUFFER ADDRESSES
*,SBA BUFFER ADDRESS IS ____ DECIMAL
ENDING MNOTES
*,STARTING ADDRESS ____, ENDING ADDRESS ____
*, ____ OPERANDS PROCESSED
*,TOTAL CORE FOR FORMAT ____ WAS ____ BYTES
*,WRAP OCCURRED
```

APPENDIX F, "MNOTES FOR SBA SEQUENCES - DISCUSSION"

The largest string of characters for an SBA sequence is 6, i.e., R21C72. If a sequence of larger than 6 is detected, mnote 10 aborts the expansion. If the first character is not an R, a numeric operand is assumed and each character is checked to see that it is numeric. Mnote 11 results if all characters are not numeric. If the first character is an R then the third or fourth characters must be C. Failure to pass this test causes mnote 12. Each row and column number is checked for a proper numeric range and if not correct, mnote 11 is used to abort Macro expansion. The row number and column number is checked against valid ranges based on the buffer size and mnotes 13, 14 & 15 document errors detected at this point. The final decimal number is next calculated from row and column notation or from the decimal number, whichever is specified, and checked against the screen size. An error in an invalid decimal number specification at this point, i.e., specifying 482 for a model 1 screen, will cause Macro abort with mnote 16. This same sequence of checks is done for all addresses in all types of operands.

APPENDIX G, "MNOTES IN CONJUNCTION WITH ILLEGAL WRAP OPERATIONS - DISCUSSION"

When FORMAT puts data, NL or EM in the output data stream or an attribute character or when a 4th operand is specified as to field length, FORMAT increases its internal buffer address by the appropriate amount. FORMAT also keeps track of the starting address and ending address in the buffer and whether or not wrap has occurred. If a calculated address is equal to the previous buffer address as maintained by FORMAT, mnote 17 results. Exception is made for Repeat to Address orders used for filling an entire buffer with the same character and Erase Unprotected to Address orders used for erasing all unprotected data in the buffer. If wrap has already occurred and the current operand exceeds FORMAT's starting address then Macro abort occurs with mnote 18. If wrap has already occurred and the current operand is causing a second wrap, mnote 19 will abort Macro expansion. Finally, if the current operation causes a wrap to occur and if it exceeds the starting address of the screen, mnote 20 will abort expansion. The following two pages illustrate these 4 situations with test data. The erase write and write operands have been omitted in the four test cases.

If CHECKING=NO is specified, as outlined in Appendix I, Macro Abort cannot occur due to Mnotes 17, 18, 19 or 20.

```
3014 T97        FORMAT    (N,R2C2,'DATA'),(N,R2C6,'D')
3015+T97        EQU       *
3016+           DC        X'11' SBA
3017                      *,SBA BUFFER ADDRESS IS 81 DECIMAL
3018+           DC        AL1(193) ADDRESS, ATTRIBUTE OR WCC
3019+           DC        AL1(209) ADDRESS, ATTRIBUTE OR WCC
3020+           DC        X'1D' SF
3021                      *,ATTRIBUTE CHARACTER IS HEX 10
3022+           DC        AL1(80) ADDRESS, ATTRIBUTE OR WCC
3023+           DC        CL4'DATA'
3024+           DC        X'11' SBA
3025                      *,SBA BUFFER ADDRESS IS 85 DECIMAL
3026+           DC        AL1(193) ADDRESS, ATTRIBUTE OR WCC
3027+           DC        AL1(213) ADDRESS, ATTRIBUTE OR WCC
3028+           DC        X'1D' SF
3029                      *,ATTRIBUTE CHARACTER IS HEX 10
3030+           DC        AL1(80) ADDRESS, ATTRIBUTE OR WCC
3031                      MNOTE 17, OPERAND NO. 2, SUBOPERAND
                          NO. 2, THE CALCULATED ADDRESS IS
                          EQUAL TO THE PREVIOUS BUFFER
3032+*                    LOCATION, OPERANDS MUST NOT OVERLAP,
                          MACRO ABORT


1802 T18        FORMAT    (,R1C3,'O'),(N,R1C2,'Y')
1803+T18        EQU       *
1804+           DC        X'11' SBA
1805                      *,SBA BUFFER ADDRESS IS 2 DECIMAL
1806+           DC        AL1(64) ADDRESS, ATTRIBUTE OR WCC
1807+           DC        AL1(194) ADDRESS, ATTRIBUTE OR WCC
1808+           DC        X'1D' SF
1809                      *,ATTRIBUTE CHARACTER IS HEX 00
1810+           DC        AL1(64) ADDRESS, ATTRIBUTE OR WCC
1811+           DC        CL1'O'
1812+           DC        X'11' SBA
1813                      *,SBA BUFFER ADDRESS IS 1 DECIMAL
1814+           DC        AL1(64) ADDRESS, ATTRIBUTE OR WCC
1815+           DC        AL1(193) ADDRESS, ATTRIBUTE OR WCC
1816+           DC        X'1D' SF
1817                      *,ATTRIBUTE CHARACTER IS HEX 10
1818+           DC        AL1(80) ADDRESS, ATTRIBUTE OR WCC
1819+           DC        CL1'Y'
1820                      MNOTE 18, OPERAND NO. 2, SUBOPERAND
                          NO. 3, AN ILLEGAL WRAP OPERATION HAS
                          OCCURRED IN THAT THE ORIGINAL START-
1821+*                    ING SCREEN LOCATION FOR THIS SCREEN
1822+*                    HAS NOW BEEN REACHED AFTER WRAPING THE
                          SCREEN ONCE, MACRO ABORT
```

```
1414 T60        FORMAT    (N,R12C35,'DATA',20),(N,R1C4,'DATA')
1415+T60        EQU       *
1416+           DC        X'11' SBA
1417                      *,SBA BUFFER ADDRESS IS 474 DECIMAL
1418+           DC        AL1(199) ADDRESS, ATTRIBUTE OR WCC
1419+           DC        AL1(90) ADDRESS, ATTRIBUTE OR WCC
1420+           DC        X'1D' SF
1421                      *,ATTRIBUTE CHARACTER IS HEX 10
1422+           DC        AL1(80) ADDRESS, ATTRIBUTE OR WCC
1423+           DC        CL4'DATA'
1424+           DC        X'11' SBA
1425                      *,SBA BUFFER ADDRESS IS 3 DECIMAL
1426+           DC        AL1(64) ADDRESS, ATTRIBUTE OR WCC
1427+           DC        AL1(195) ADDRESS, ATTRIBUTE OR WCC
1428+           DC        X'1D' SF
1429                      *,ATTRIBUTE CHARACTER IS HEX 10
1430+           DC        AL1(80) ADDRESS, ATTRIBUTE OR WCC
1431                      MNOTE 19, OPERAND NO. 2, SUBOPERAND
                          NO. 2, YOU HAVE ALREADY WRAPPED
                          ONCE AND ARE NOW ATTEMPTING TO
1432+*                    WRAP AGAIN, MACRO ABORT


2561 T78        FORMAT    (,R5C5,'D'),(,R5C5,'D')
2562+T78        EQU       *
2563+           DC        X'11' SBA
2564                      *,SBA BUFFER ADDRESS IS 324 DECIMAL
2565+           DC        AL1(197) ADDRESS, ATTRIBUTE OR WCC
2566+           DC        AL1(196) ADDRESS, ATTRIBUTE OR WCC
2567+           DC        X'1D' SF
2568                      *,ATTRIBUTE CHARACTER IS HEX 00
2569+           DC        AL1(64) ADDRESS, ATTRIBUTE OR WCC
2570+           DC        CL1'D'
2571+           DC        X'11' SBA
2572                      *,SBA BUFFER ADDRESS IS 324 DECIMAL
2573+           DC        AL1(197) ADDRESS, ATTRIBUTE OR WCC
2574+           DC        AL1(196) ADDRESS, ATTRIBUTE OR WCC
2575+           DC        X'1D' SF
2576                      *,ATTRIBUTE CHARACTER IS HEX 00
2577+           DC        AL1(64) ADDRESS, ATTRIBUTE OR WCC
2578                      MNOTE 20, OPERAND NO. 2, SUBOPERAND
                          2, A WRAP OPERATION HAS JUST OCCURRED
                          BUT HAS EXCEEDED THE START
2579+*                    ADDRESS FOR THIS SCREEN, MACRO ABORT
```

APPENDIX H - INCLUDING LABELS IN A FORMAT OUTPUT DATA STREAM

The Programmer can create a label anywhere in the Output Data Stream by coding a label operand wherever he desires. The label operand is a single label enclosed in parenthesis and beginning with a dollar sign. Thus ($HERE) will generate:

```
$HERE     EQU     *
```

If the label operand preceded a basic operand that had a leading SBA sequence as well as a Start Field Order then the following offsets would be applicable:

Address	Offset to:
$HERE	SBA Order
$HERE+1	SBA Address
$HERE+3	Start Field Order
$HERE+4	Attribute Character
$HERE+5	1st byte of Output Data

APPENDIX I - USING THE CHECKING=NO OPERAND

If you do not desire FORMAT to check the ascending sequence of operands, you may code (CHECKING=NO) anywhere in the sequence of operands. As a result, FORMAT will not Abort with MNOTES 17 thru 20 as per Appendix G. FORMAT will refrain from any further checking until you code (CHECKING=YES). If the first use of FORMAT utilizes (CHECKING=NO) then it is necessary to code (CHECKING=YES) as the first operand of the next use of FORMAT if you desire checking to start again for that screen.

If you desire to add several fields to the end of a set of existing FORMAT macro operands, by removing the (ETX) operand and substituting (CHECKING=NO), you may now code any number of additional operands in any order without creating an Abort condition. Following the last of the added operands you would then code the (ETX) operand.

3270 Format Macro For Output Data Streams
Program Description/Operations Manual

Printed in U.S.A.

SB21-0621-0

SB21-0621-0

IBM
International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)