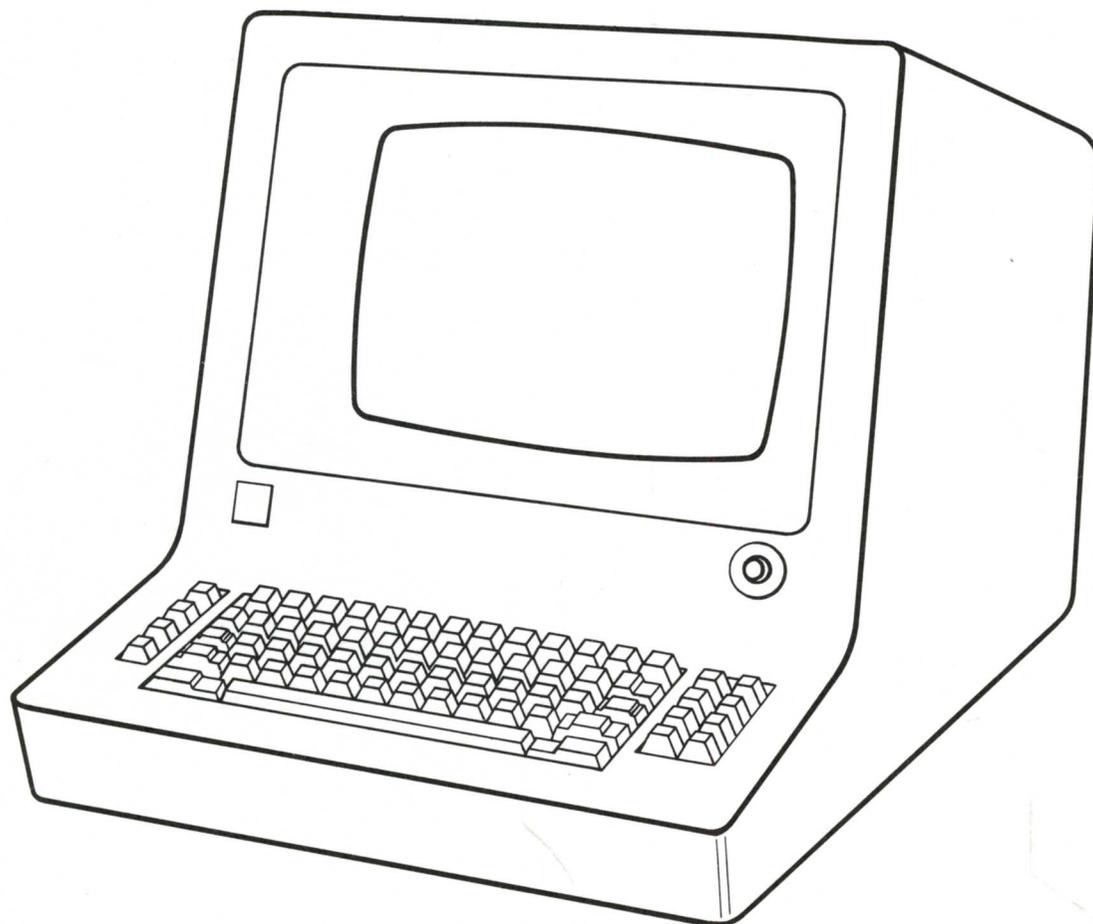


GA34-0026-1

File No. S1-03

IBM Series/1  
4979 Display Station  
Description



4979 DISPLAY DESCRIPTION

GA34-0026-1

File No. S1-03

IBM Series/1  
4979 Display Station  
Description

**4979 DISPLAY DESCRIPTION**

**Second Edition (March 1977)**

This is a major revision of, and obsoletes GA34-0026-0. Significant changes in this new edition include:

Restructuring of certain status words  
Changing the format of data flow information

Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters. Before using this publication in connection with the operation of IBM systems, have your IBM representative confirm editions that are applicable and current.

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, send your comments to IBM Corporation, Systems Publications, Department 27T, P.O. Box 1328, Boca Raton, Florida 33432. Comments become the property of IBM.

© Copyright International Business Machines Corporation 1976, 1977

<b>Preface</b>	v		
Prerequisite Knowledge	v		
Prerequisite Publications	v		
Related Publications	v		
<b>Chapter 1. Introduction</b>	1-1		
Display Station Functional Description	1-1		
Attachment Feature Functional Description	1-1		
Standard Features	1-2		
Cycle Steal	1-2		
Electronic Lock Out Feature	1-2		
Data Protect Feature	1-2		
Character Set	1-2		
Basic Components	1-2		
Display Screen	1-2		
Keyboard	1-2		
Controls	1-3		
<b>Chapter 2. Operation and Instruction</b>	2-1		
<b>Display Operations</b>	2-1		
Unformatted and Formatted Display Images	2-1		
Unformatted Mode	2-1		
Formatted Mode	2-1		
Data Representation	2-2		
Keyboard Operations	2-2		
Cursor	2-2		
Keyboard	2-3		
SHIFT/LOCK Keys	2-3		
Graphic Alphanumeric Keys	2-3		
Local Function Keys	2-5		
Interrupt Request Keys	2-9		
<b>Chapter 3. Programming Input/Output Operations</b>	3-1		
Data Transfer Operations	3-1		
Direct Program Control (DPC)	3-1		
Cycle Steal	3-1		
Initiating a Display Operation	3-1		
Operate I/O Instruction	3-1		
Using the IDCB	3-2		
Input/Output Commands and Display Operations	3-2		
Command Execution Under DPC Mode	3-2		
Command Execution in CS Mode	3-4		
Using the DCB	3-5		
Status Information	3-10		
Condition Codes	3-10		
Condition Code 2—Exception	3-10		
Condition Code 3—Device End	3-10		
Condition Code 4—Attention	3-11		
Interrupt Identification Word	3-11		
Interrupt Information Byte	3-11		
Interrupt Status Byte	3-11		
Cycle Steal Status Words	3-12		
Status After Power and Resets	3-13		
<b>Appendix A. Characters—EBCDIC Hexadecimal Equivalents</b>	A-1		
<b>Appendix B. Start I/O Examples</b>	B-1		
DCB Start I/O Examples	B-1		
START (Read From Buffer) Input	B-1		
START (Set Cursor and Device Status) Output	B-1		
START (Write to Buffer/Protected) Output	B-1		
START (Write/Erase-After to End of Line) Output	B-2		
START (Shift Lines/Write Protected/Erase After) Output	B-2		
START (Erase/Unprotected) Output	B-2		
<b>Index</b>	X-1		



This manual contains reference material and is a source of information about the IBM Series/1 4979 Display Station Model 1, and the 4979 Display Attachment Feature.

Chapter 1 is an introduction to the general characteristics and features of the 4979.

Chapter 2 contains information on the functions and operation of the display screen and its associated features.

Chapter 3 discusses the data flow between the processor and the 4979. Specific topics are:

- I/O Commands
- I/O Operations
- Status Information

Appendix A contains EBCDIC characters and hexadecimal equivalents.

Appendix B contains specific examples of different Start commands.

## Prerequisite Knowledge

This book assumes the reader has a background in data processing concepts and is familiar with the hexadecimal numbering system as used in IBM systems. It is assumed the reader has a basic understanding of displays, keyboards, and their relationship to a processor and an understanding of stored program concepts.

## Prerequisite Publications

*IBM Series/1 Model 5 4955 Processor and Processor Features Description*, GA34-0021

*IBM Series/1 Model 3 4953 Processor and Processor Features Description*, GA34-0022

*IBM Series/1 System Summary*, GA34-0035

## Related Publications

*IBM Series/1 Installation Manual--Physical Planning*, GA34-0029-1

*IBM Series/1 Configurator*, GA34-0042

*IBM Series/1 Operator's Guide*, GA34-0039



The IBM Series/1 4979 Display Station (Figure 1-1) is a table top unit that serves as a communication link between the user and the system. It provides image display of data transmitted to and from the processor. The display station enables the user to enter, modify, or delete data on the display, and to cause the revised display to be returned to the processor for storage or additional processing.

Manual operations (such as displaying and altering data, and loading programs), are performed by keying characters from the keyboard into the display video screen. The screen is used to display all actions. There are no indicator lights to be decoded, because all information is displayed on the screen in clear text. A console printer, the 4974 is available as an optional feature to record messages as hard copy.

This chapter is an introduction to the basic components and features of the IBM 4979 Display Station.

The display station connects to the attachment card (a prerequisite for the 4979) which is located in one of the following rack mounted units:

- IBM Series/1 Model 5 4955 Processor
- IBM Series/1 Model 3 4953 Processor
- IBM Series/1 4959 Input/Output Expansion Unit

## Display Station Functional Description

The IBM 4979 Display Station:

- Consists of a 12-inch video monitor attached to the processor (through the device attachment card).
- Contains an alpha-numeric keyboard.
- Is a table mounted unit.
- Displays numeric characters, upper case alphabetic characters, and special symbols.
- Is equipped with manual intensity adjustment.
- Is used for displaying and selecting operating modes and operator entries.

## Attachment Feature Functional Description

The attachment feature card:

- Serves as the connection between the processor and the display station.
- Interprets and executes commands from the processor.
- Provides a path for data between the processor and the display station.
- Transfers data in parallel by byte.
- Furnishes status information to the processor and reports condition codes after I/O instructions and during an interrupt.
- Provides a buffer for storing data.

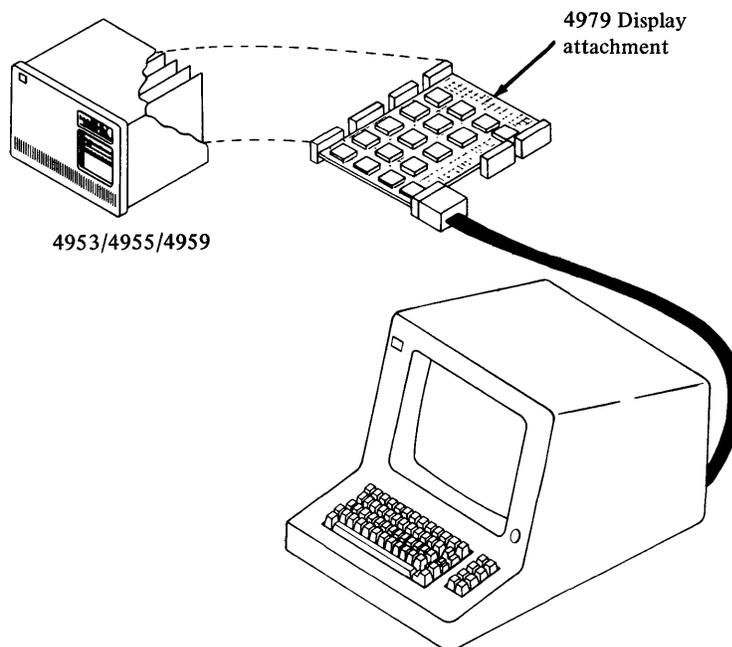


Figure 1-1. IBM Series/1 4979 Display Station and attachment

## Standard Features

The 4979 has the following standard features:

### Cycle Steal

The display station operates in *cycle steal* mode. In cycle steal mode I/O operations are overlapped with processing operations, so that processing can continue while I/O operations are in progress.

### Electronic Lock Out Feature

An electronic lock out feature protects against unauthorized operator entry. This feature is provided under program control. See "Interrupt Request Keys" in Chapter 2.

### Data Protect Feature

The data protect feature prohibits the operator from altering protected data. This feature is provided under program control.

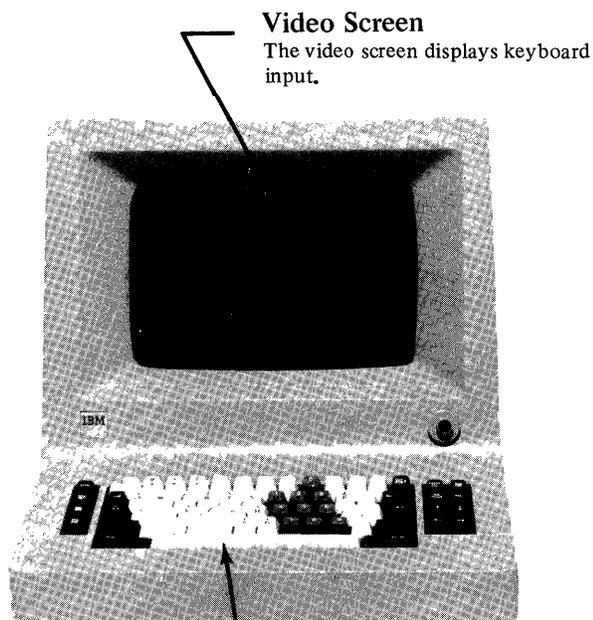
### Character Set

The display station uses a 64 character subset of EBCDIC (Extended Binary Coded Decimal Interchange Code).

## Basic Components

The display station has two basic components:

- Display screen
- Console keyboard



### Console Keyboard

The keyboard is used for keying information into the system, altering data, starting and stopping programs, and so on.

Figure 1-2. Basic Components (screen, keyboard)

## Display Screen

The display screen (Figure 1-2) is a 12-inch CRT (cathode ray tube) similar to that of a television receiver. It has a capacity of 1920 characters arranged in a format of 24 lines of 80 characters each. The display provides a buffer for storing data, the capacity of the buffer is the same as the character capacity of the screen.

Characters displayed consist of dots generated in a pseudo 7 x 7 dot matrix on a raster CRT scanned screen.

All information displayed on the screen must be regenerated periodically for retention of the image. The speed at which the display is reproduced is called the "refresh" rate. The 4979 is designed to "refresh" the display approximately 60 times per second, creating an easy to read display under normal lighting conditions.

## Keyboard

The display station keyboard (Figure 1-2) is similar to that of a data entry style key layout. The alphanumeric, punctuation, and special character keys form the main body of the keyboard with special control keys on each side.

The keyboard is arranged in four different key groups:

- Shift/Lock Keys
- Graphic Alphanumeric
- Local Function
- Interrupt Request

All keys are color coded for ease of use:

- Alphabetic and graphic keys are white keys with black letters.
- Numeric keys are light gray keys with uppershift numbers in white and lowershift characters in black.
- Local Function and Interrupt keys are charcoal gray keys with white letters or symbols.

ATTN	(	#	'	\$	*	<	-	0	/				DUPL	ERASE	EOS
	)	@	%	+	^	>	_	1	2	3	&	P	DUPL	EOL	EOF
PF4	PF1	PF5	PF2	PF6	PF3									INS	DEL
														MODE	
														↑	↓
														←	→
															ENTER

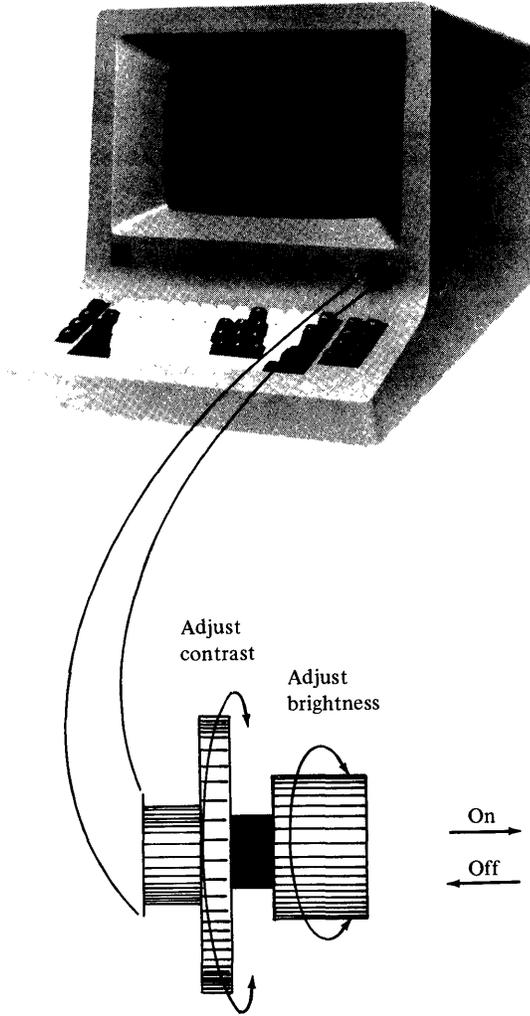
The keyboard character set is:

A through Z  
 0 through 9  
 \$ , . + - \* / % @ & ' #  
 ( ) { } \  
 = > <  
 ! " ?  
 - | : ; ↵  
 Space

## Controls

The display station has three external off-keyboard operator controls (located on one stem):

- ON-OFF
- Brightness
- Contrast



These controls, located just above the keyboard on the front of the unit, allow for (1) powering the unit ON/OFF and (2) adjustment of the brightness and contrast of the display for a comfortable viewing level.



.

v



.

.



### Display Operations

The 4979 provides a buffer for storing data. The capacity of this buffer is the same as the character capacity of the display screen, i.e., 1920 characters. Display data that is stored in the buffer is presented to the operator on the display screen in the form of alphanumeric characters and symbols. The display is refreshed continuously (regenerated) from the buffer to maintain a constant image on the screen.

The following sections provide information on the functions and operations of the display screen and its associated features.

### Unformatted and Formatted Display Images

There is a fixed relationship between each buffer storage location and its related character position on the display screen. Buffer address locations are referenced from 0 (the first displayable character location in the upper left corner of the screen) to 1919 (the last displayable character location in the bottom right corner of the screen).

By using these address locations under appropriate commands, a program can load a display station buffer with a display image that exactly satisfies the requirement.

An application program can present a display to the operator in one of the following modes (See Figure 2-1):

- Unformatted mode
- Formatted mode

#### *Unformatted Mode*

In unformatted mode, the display station is left unformatted and the display user uses the screen in a free-form manner.

#### *Formatted Mode*

In formatted mode, the display image is completely or partially formatted by the application program. There are two types of data fields on a formatted display. They are:

- Unprotected data field
- Protected data field

#### **Unprotected Data Field**

All operator *entries* will go into unprotected fields known as input fields. The fields are highlighted on the screen by characters of brighter intensity. Data can be entered, modified, or erased by keyboard action in any unprotected input field.

#### **Protected Data Field**

The protected data field is primarily for the use of the program, and data cannot be entered into this field by the operator. Protected data fields are comprised of contiguous protected character positions designated by the application program. These fields are distinguishable on the screen by a lower intensity.

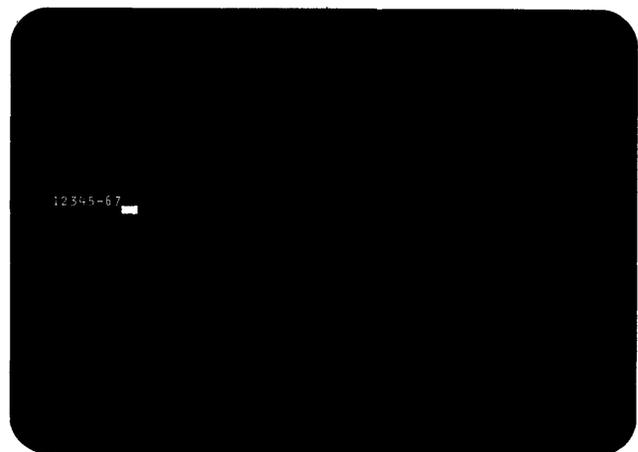
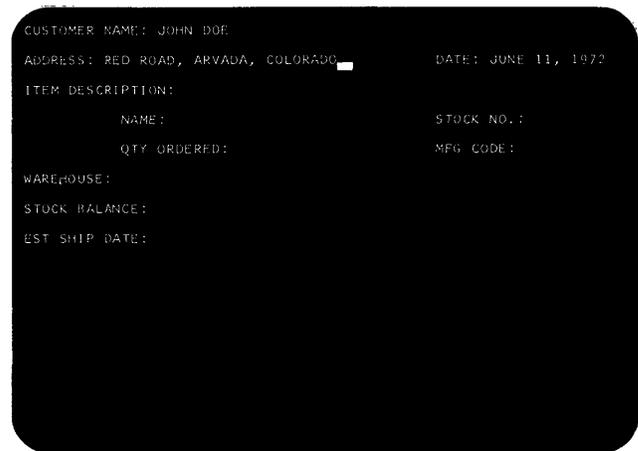


Figure 2-1. The image on your screen might look like one of the above

## Data Representation

All data is transmitted from the processor to the display station buffer in EBCDIC or the applicable character set (see Figure 2-2). The 64 EBCDIC characters include the upper case alphanumeric characters, space, special graphic symbols. Characters transmitted from the processor that are "outside" the 64-character EBCDIC subset are accepted, converted to a NULL, and displayed on the screen as a blank.

When a character is transferred to the buffer . . .

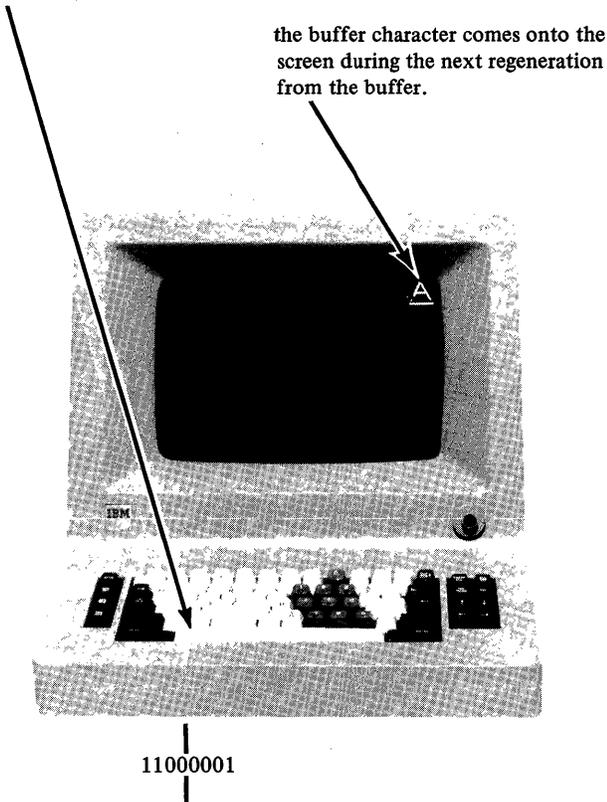


Figure 2-2. Data representation

## Keyboard Operations

The keyboard enables the user to change, edit, or create character displays except within fields protected from keyboard operations by the program. As messages are being composed or modified by keyboard operations, the changes are inserted in the buffer and displayed on the subsequent display fresh cycle.

When the user completes an operation and presses one of the interrupt request keys, for example the ENTER key, an interruption informs the program; the program may then read the modified data fields from the display buffer.

## Cursor

A special symbol (that resembles an underscore), called a cursor (Figure 2-3), is displayed beneath a character or character position on the display screen to indicate where the next character entered from the keyboard will be stored (Figure 2-4). For example, when the cursor is displayed under one character in a line of characters, that character can be changed or deleted by keyboard action unless it is a protected character. Also, if the cursor is displayed under a position without a display character, a character can be inserted in that position by keyboard action unless it is a protected space character. However, when the cursor appears beneath a protected character or a protected space character position, that position cannot be modified by keyboard action. Attempting to do so causes the cursor to be moved one position to the right.

One, and only one, cursor is capable of being displayed on the screen. When the display is turned on, the cursor is automatically generated and displayed in the first location on the screen (Figure 2-3), following a command from the application program to unlock the keyboard. A power up of the display station causes an interrupt to the application program. The cursor can be repositioned by the keyboard operator and also by the program. Following a depression of any interrupt key, the cursor is blanked as an indication to the operator that the keyboard is locked. A command from the application program is required to unlock the keyboard and allow the cursor to reappear. The cursor is flashed during Insert Mode operation (see Insert Mode).

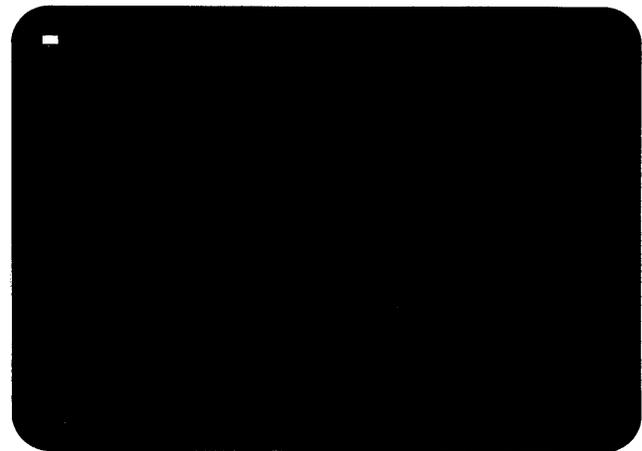


Figure 2-3. Cursor

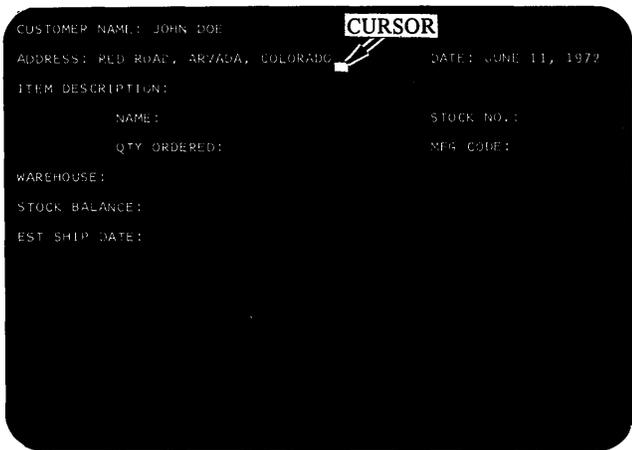


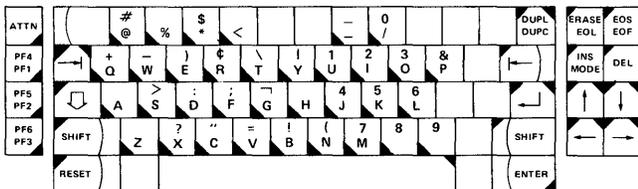
Figure 2-4. The cursor looks like an underscore

### Keyboard

The display station has a Data Entry—typewriter style keyboard.

The keys on the keyboard are divided into four main groups. Refer to keyboard and legend in Figure 2-5.

- Shift/Lock Keys
- Graphic Alphanumeric Keys
- Local Function Keys
- Interrupt Request Keys

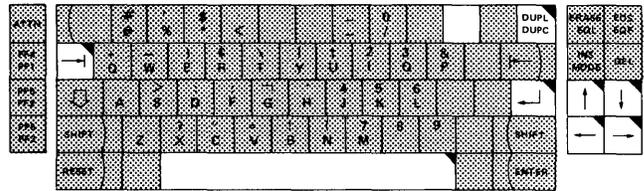


#### Legend

- (Typamatic action)
- Graphic alphanumeric
- Local function & Shift Lock
- Interrupt request

Figure 2-5. Keyboard groups

Certain keys within the Graphic Alphanumeric and Local Function categories have *typamatic* action (Figure 2-6). This means that they have the ability to repeat their character or operation automatically if held down. The typamatic keys increase operating speed. After an initial pause after the first key entry, the typamatic keys operate at approximately 10 key entries per second. The typamatic action of individual keys is discussed in greater detail under the individual key descriptions in the following sections.



#### Legend

- Typamatic action

Figure 2-6. Typamatic keys repeat their function when held down

### SHIFT/LOCK Keys

The SHIFT and LOCK keys operate the same as on a standard typewriter keyboard (Figure 2-7). When either the left or the right SHIFT key is pressed, the upper symbols shown on the dual character keys are displayed when their respective keys are pressed. If the character key has no upper symbol (such as some alphabetic keys), the SHIFT key has no effect on the character displayed. The SHIFT keys are nonlocking and must be held down.

Pressing the LOCK key (wide arrow) electronically locks the keyboard in a shifted (upshift) condition. When the LOCK key is used, holding the SHIFT key down is unnecessary, and both hands are free for typing. Press either SHIFT key when ready to return the keyboard to the unshifted (downshift) condition. This releases the electronic lock.

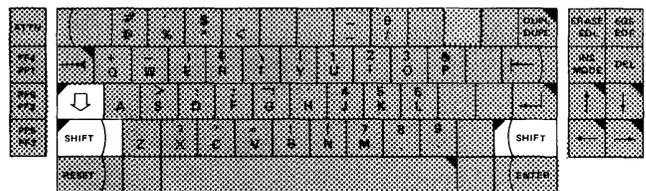


Figure 2-7. Shift/lock keys

### Graphic Alphanumeric Keys

These keys (Figure 2-8) represent the 64 printable alphanumeric, space, and graphic symbols contained within the EBCDIC subset table. Each key entry within this group causes the appropriate symbol to be displayed on the screen in the position marked by the cursor. When the character is displayed, the cursor is moved one position to the right. This routine is repeated for each key entry. The end of the display line will wrap to the beginning of the next lower line (see discussion of cursor wrap under Cursor Positioning Keys this chapter) and the end of the bottom display line wraps to the beginning of the top line.

Keys that have two characters can display either of them, depending on the position of the SHIFT key. The lower character displays when the character key is pressed and the keyboard is in the down shifted mode (the SHIFT key not pressed). To display the upper character, hold the SHIFT key down while pressing the character key or press and release the LOCK key before pressing the character key.

Remember, none of the Graphic Alphanumeric keys will cause an interrupt to the program.

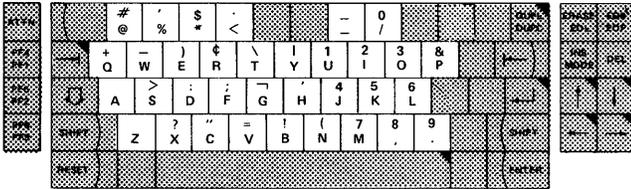


Figure 2-8. Graphic alphanumeric keys

The graphic alphanumeric keys are:

<i>Lower shift symbol</i>	<i>Upper shift symbol</i>
Q	+ = plus
W	_ = underscore
E	) = right parenthesis
R	¢ = cent sign
T	\ = reverse slash
Y	= logical OR
U	1 = numeric one
I	2 = numeric two
O	3 = numeric three
P	& = ampersand
A	blank
S	> = greater than sign
D	: = colon
F	; = semicolon
G	¬ = logical NOT
H	' = prime apostrophe
J	4 = numeric four
K	5 = numeric five
L	6 = numeric six
Z	blank
X	? = question mark
C	" = quotation mark
V	= = equal sign
B	! = exclamation point
N	( = left parenthesis
M	7 = numeric seven
@ = at sign	# = number sign
% = percent sign	, = comma
* = asterisk	\$ = dollar sign
< = less than sign	. = period
- = minus sign	- = minus sign
/ = slash	0 = zero
, = comma	8 = numeric eight
. = period	9 = numeric nine

To familiarize yourself with the keyboard, it may be convenient to look at the Graphic Alphanumeric keys in groups:

- Alphabetic Characters
- Numeric Characters
- Symbol and Punctuation Characters
- Space Bar

### Alphabetic Characters

All but two of the alphabetic keys (the A and Z) are dual character keys (Figure 2-9). The alphabetic characters on the faces of these keys are always displayed as capitals (uppercase characters) when the key is pressed. When the key is pressed with the keyboard in upshifted mode, the numeric or symbol (on the upper half of the key face) is displayed.

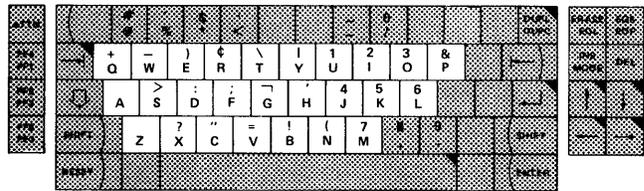


Figure 2-9. Alphabetic keys

### Numeric Characters

With the keyboard in upshifted mode, Graphic Alphanumeric keys display the numeric symbol shown on their upper half (Figure 2-10).

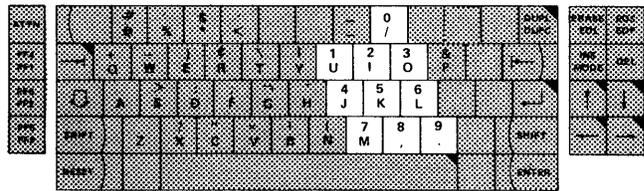


Figure 2-10. Numeric keys

### Graphic Characters

Graphic characters are located on the upper half of most of the alphabetic keys, and on the lower half of two of the numeric keys. The graphic that appears on the upper part of the key displays only with the keyboard in shifted mode.

## Space Bar

A space is considered an actual character that occupies a position or location on the face of the screen. When the space bar (Figure 2-11) is pressed in upshifted mode, a space code or blank is entered on the screen and will replace whatever character is presently in that position (a space can never be inserted in place of a protected character when the keyboard is in the upshifted mode). The space bar can be used to cause cursor advancement to the right, through unprotected positions, if the keyboard is in the down shifted position (SHIFT key down). If a space is desired at a position occupied by any unprotected character other than a NULL code, the keyboard must be upshifted.

Remember, in upshifted mode the space bar replaces all unprotected positions with blanks (space codes). In down-shift mode, the space bar will advance the cursor without altering data except NULLS. The space bar has typamatic action.

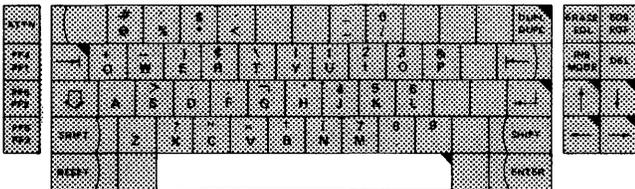


Figure 2-11. Space bar

## Local Function Keys

These keys (Figure 2-12), the third of the four main groups, cause movement of data characters or cursor within the attachment but they do not cause an interrupt request. The keys in this group are:

- |                  |            |
|------------------|------------|
| TAB (forward)*   | DUPL/DUPC* |
| TAB (backward)   | Erase EOL  |
| New Line Cursor* | EOS/EOF    |
| Up Cursor*       | INS Mode   |
| Down Cursor*     | DEL Mode   |
| Left Cursor*     | Reset      |
| Right Cursor*    |            |

\*These keys are typamatic keys.

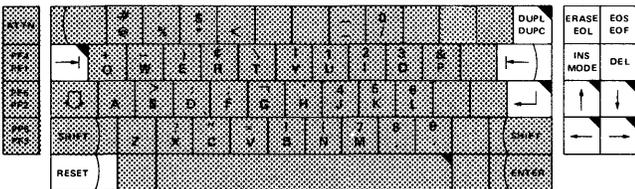
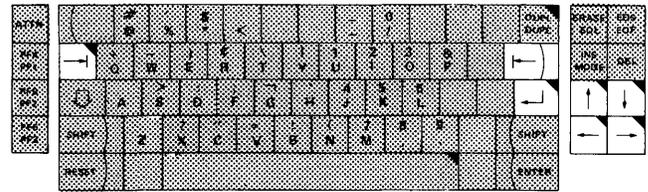


Figure 2-12. Local function keys

## Cursor Positioning Keys



- |                    |                     |
|--------------------|---------------------|
| → Tab              | ← Move cursor left  |
| ↵ New Line         | → Move cursor right |
| ↑ Move cursor up   | ← Back tab          |
| ↓ Move cursor down |                     |

Figure 2-13. Use the cursor control keys to reposition the cursor

Because the position of the cursor determines where the next character keyed will be entered, it is important to know how to move the cursor to any location on the screen. The operator will want to be able to move the cursor around quickly to positions where data is to be entered or altered. These keys are part of the Local Function keys. They provide a means of rapid positioning of the cursor to any character position without affecting any of the information already on the screen. Cursor positioning keys (Figure 2-13) move the cursor in a horizontal (right or left) or vertical (up or down) direction.

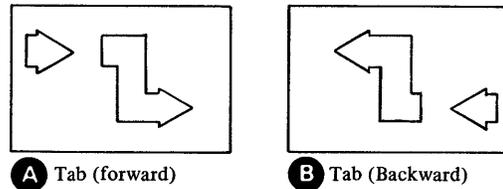
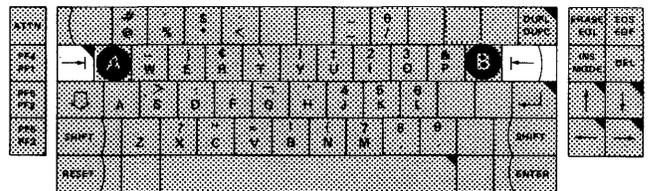
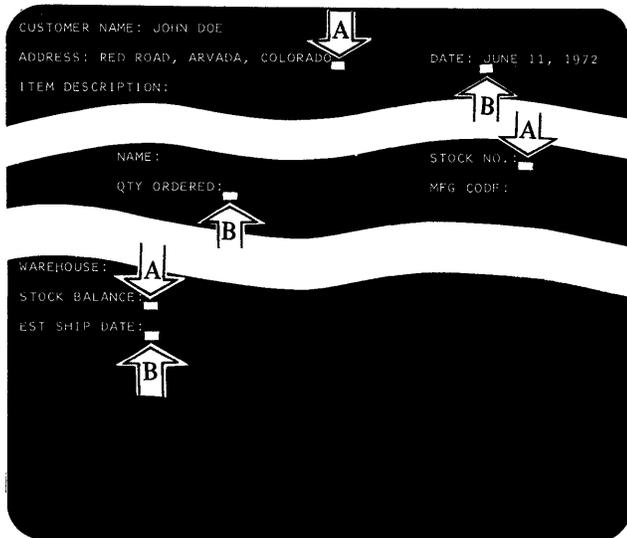


Figure 2-14. Tabulator keys

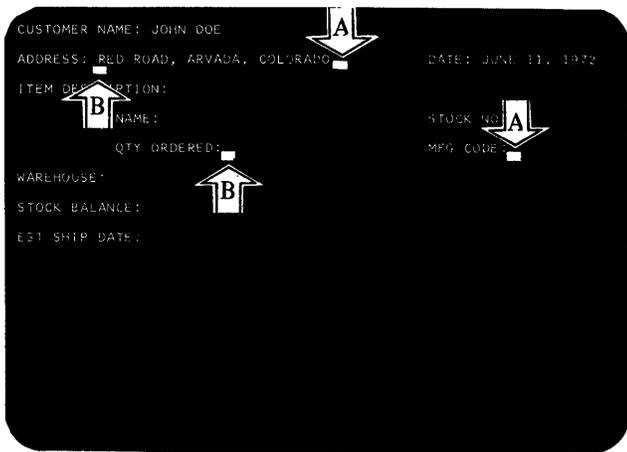
**Tab (Forward).** This key (Figure 2-14-A) causes the cursor to skip to the right to the first position of the first unprotected field it encounters; then, in typamatic mode, skip to all first character positions of unprotected fields encountered line-by-line in a left-to-right, top-to-bottom movement. The first position of an unprotected field is the first position encountered after a protected position. On reaching the bottom right hand corner of the screen, it wraps to the upper left corner of the display screen. On a display with no unprotected fields, the cursor is repositioned to character location zero.

**Tab (Backward).** This key (Figure 2-14-B) causes the cursor to skip to the leftmost position of the first unprotected character field encountered. The leftmost position of an unprotected field is the character to the immediate right of a protected character. If no unprotected position is encountered, the cursor wraps to the upper leftmost position of the display image.



The tab key moves the cursor to the next input field

A = Before tab      B = After tab



The back tab key moves the cursor back to the beginning of an input field (upper A-to-B) or the first character position of the preceding field (lower A-to-B).

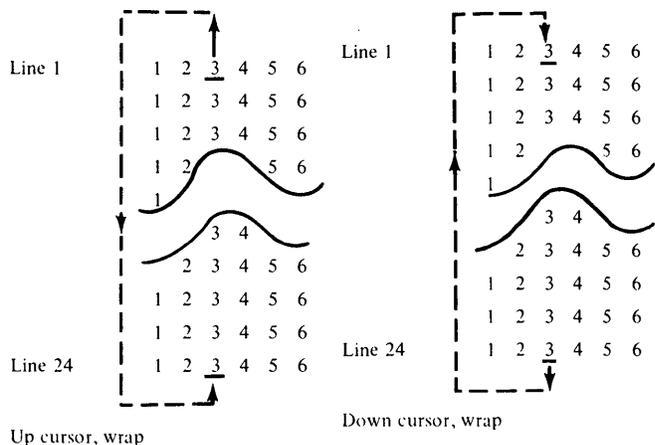
**New Line Cursor.** This key moves the cursor in the following manner:

- If the screen is unformatted, the cursor is moved to the first position in the next line.
- If the screen is formatted, the cursor moves to the first unprotected position on the first new line that contains unprotected characters.

If all character positions on the screen are protected, the cursor wraps to the first character position on line one, that is, the upper left corner of the display.

The New Line Cursor key is typamatic.

**Up Cursor.** This key moves the cursor upward (bottom to top) one line from its current position, but in the same character column. On reaching the top of the screen, it wraps to the same position on the bottom and continues upward again. This key is typamatic; it will continue cursor movement, line-by-line, until the key is released.

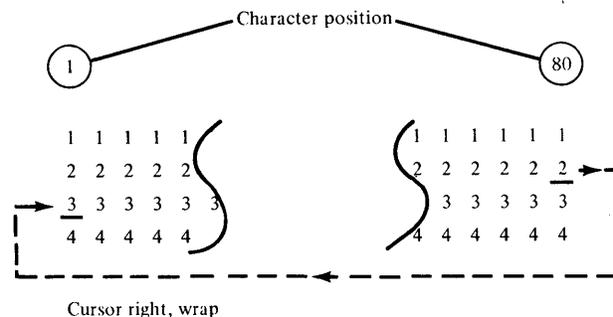


Up cursor, wrap

Down cursor, wrap

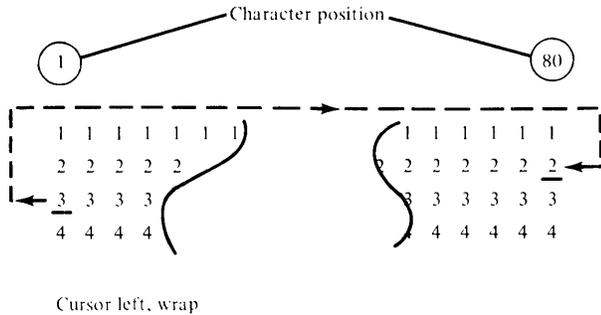
**Down Cursor.** This key moves the cursor downward (top to bottom) one line from its current position, but in the same character column. On reaching the bottom of the screen, it wraps to the top and continues downward again. This key is typamatic; it will continue cursor movement, line-by-line, until the key is released.

**Left Cursor.** This key moves the cursor right-to-left, one character position at a time, progressing across the screen. On reaching the left side of the screen it wraps to the right most position on the line above. On reaching the top left hand corner of the screen, it wraps to the bottom right hand position and continues moving as long as the key is held down. This is a typamatic key.



Cursor right, wrap

**Right Cursor.** This key moves the cursor left-to-right, one position at a time, progressing across the screen. On reaching the right side of the screen, it wraps to the left-most position on the line below. On reaching the bottom right hand corner of the screen, it wraps to the top left hand position and continues moving as long as the key is held down. This key is typamatic.



This completes the description of the cursor positioning keys (see Figure 2-15 for summary), but there are still more keys in the Local Function group.

**DUPC (Down Shift–Duplicate Character)**

This duplicate character key enters the character to the left of the cursor into the display position, marked by the cursor, provided that position is not a protected position. When the character is entered it is unprotected and the cursor is moved one position to the right. Because it's a typamatic key, this routine is repeated as long as the DUPC key is depressed or until a protected character is encountered. If a protected character is encountered, it remains unchanged and the cursor moves over one character position to the right. The end of the display line wraps to the next lower line and the end of the bottom display line wraps to the top line. This function operates only with the keyboard in the down shifted mode.

ATTN	(	)	*	%	\$	<	>	0					DUPL DUPL	ERASE EOL	EOS EOF
PF4 PF1	←	Q	W	E	R	T	Y	U	I	O	P	→	←	↵	DEL
PF5 PF2	⏏	A	S	D	F	G	H	J	K	L			↕	↕	
PF6 PF3	SHIFT	Z	X	C	V	B	N	M					SHIFT	←	→
	RESET												ENTER		

**DUPL (Upper Shift–Duplicate Line)**

The duplicate line key enters the character located one line step above the current cursor line, provided the current cursor position is not a protected character. When the character is entered it is unprotected and the cursor is moved one position to the right. Because this is a typamatic key, this routine is repeated as long as the DUPL key is pressed or until a protected character is encountered. If a protected character is encountered it remains unchanged and the cursor moves over one position to the

right. The end of the display line wraps to the next lower line and the end of the bottom display line wraps to the top line. This key operates only with the keyboard in upshifted mode.

**Erase EOL (Erase to End of Line)**

Pressing this key will enter a null data character (HEX 00) at the cursor position, provided it is an unprotected position, and at all unprotected positions to the right of the cursor through the end of the line. This will erase all unprotected data (protected data is skipped) from the cursor location to the end of the line (EOL). The cursor will be positioned at the first position on the next lower line (unless it is on the bottom of the screen, then it will wrap to the top leftmost position). When the cursor skips to a new line, this key will have to be pressed to do another Erase EOL operation.

ATTN	(	)	*	%	\$	<	>	0					DUPL DUPL	ERASE EOL	EOS EOF
PF4 PF1	←	Q	W	E	R	T	Y	U	I	O	P	→	←	↵	DEL
PF5 PF2	⏏	A	S	D	F	G	H	J	K	L			↕	↕	
PF6 PF3	SHIFT	Z	X	C	V	B	N	M					SHIFT	←	→
	RESET												ENTER		

**EOS/EOF Key**

**EOS (Upper Shift–Erase to End of Screen).** This key will enter a null character (Hex 00) at every unprotected position starting with the current cursor position and ending at the lower rightmost corner of the screen. This erases all unprotected data from the current cursor location to the end of the screen (EOS). The cursor will then wrap to the top leftmost position of the screen.

**EOF (Lower Shift–Erase to End of Field).** This key will enter a null data character in every unprotected position starting with the current cursor position and every position to the right of the cursor until a protected position is encountered or the end of the screen is detected. This will erase all unprotected data from the current cursor location to the end of the unprotected field (EOF). The cursor will then move to the first protected character or to the top leftmost position of the screen. If the EOF key is pressed when the cursor is at a protected character position, the cursor is advanced to the next unprotected field encountered, or until the end of the display screen is encountered. In this manner, the protected characters are “skipped over.”

ATTN	(	)	*	%	\$	<	>	0					DUPL DUPL	ERASE EOL	EOS EOF
PF4 PF1	←	Q	W	E	R	T	Y	U	I	O	P	→	←	↵	DEL
PF5 PF2	⏏	A	S	D	F	G	H	J	K	L			↕	↕	
PF6 PF3	SHIFT	Z	X	C	V	B	N	M					SHIFT	←	→
	RESET												ENTER		

### INS (Insert Mode)

Pressing this key places the keyboard in an Insert Mode of operation. All of the keys will perform normally while the keyboard is in this mode. This mode allows entry of a character or characters into the middle of a line or field without disturbing the information already displayed there. All of the entries will be inserted at the cursor location provided that position does not contain a protected character, and that one or more null characters exists at the end of the line or field. Upon inserting a character all of the unprotected characters on that line, as well as the cursor, will be shifted one position to the right of the cursor. This operation is halted if the operator tries to insert any characters into a protected location, or if any

non-null character would be shifted out at the end of a line or shifted into a protected location. The cursor may be advanced to the right with the cursor control keys, skipping over any protected locations, allowing the Insert Mode to be continued. The number of characters that can be inserted is limited to the number of blanks remaining in that field or line.

Insert Mode continues until a Reset or a Delete key is pressed, or until a Power-On-Reset, System Reset, Device Reset or Halt condition occurs. (Figure 2-15 shows an example of inserting data.)

*Note.* The Insert Mode condition is denoted by a flashing cursor on the screen.

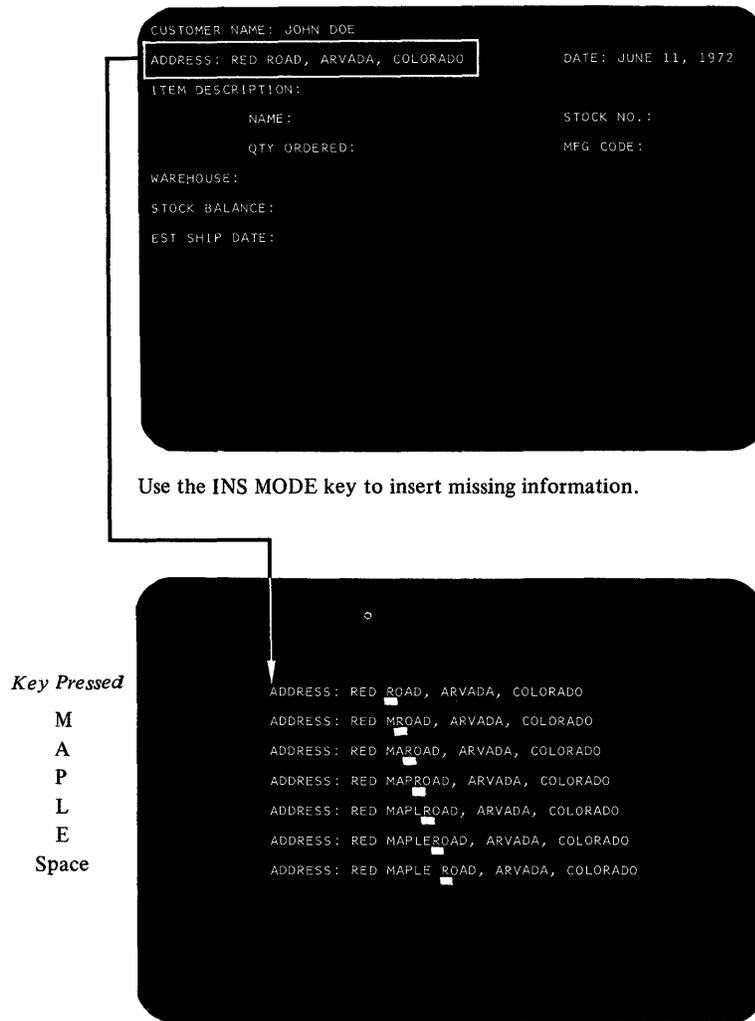


Figure 2-15. Inserting a word moves some displayed data to the right

## DEL (Delete Key)

Pressing the DELETE key when the cursor is located in an unprotected field deletes the character occupying the position underlined by the cursor. All characters to the right of the cursor in that line or field (on the same line) are shifted one position to the left. Null characters are inserted at the end of the line (or unprotected field).

If the delete is attempted at a protected field, the delete is not executed and the cursor is not moved. This key places the 4979 attachment into a non-insert mode.

ATTN																	DUPL	ERASE	EOS
																	DUPL	ERASE	EOS
PF4	PF1															INS	MODE	DEL	
PF5	PF2															↑	↓		
PF6	PF3	SHIFT	Z	X	C	V	B	N	M	7	8	9	SHIFT			ENTER			
		RESET																ENTER	

## RESET Key

Pressing the RESET key restores the keyboard to normal operation. The RESET key *will not* reset when the display is sending data to or receiving data from the processor. This key is used to deactivate the INS MODE key.

ATTN																	DUPL	ERASE	EOS
																	DUPL	ERASE	EOS
PF4	PF1															INS	MODE	DEL	
PF5	PF2															↑	↓		
PF6	PF3	SHIFT	Z	X	C	V	B	N	M	7	8	9	SHIFT			ENTER			
		RESET																ENTER	

## Interrupt Request Keys

The last group of keys are the Interrupt Request keys (Figure 2-16).

- ATTN (Attention)
- ENTER
- PF1 (Program Function 1 lower half)
- PF2 (Program Function 2 lower half)
- PF3 (Program Function 3 lower half)
- PF4 (Program Function 4 upper half used with SHIFT)
- PF5 (Program Function 5 upper half used with SHIFT)
- PF6 (Program Function 6 upper half used with SHIFT)

ATTN																	DUPL	ERASE	EOS
																	DUPL	ERASE	EOS
PF4	PF1															INS	MODE	DEL	
PF5	PF2															↑	↓		
PF6	PF3	SHIFT	Z	X	C	V	B	N	M	7	8	9	SHIFT			ENTER			
		RESET																ENTER	

Figure 2-16. Interrupt request keys

The Interrupt Request keys cause an attention interrupt request to the processor. The attachment causes a keyboard lockout condition, which prohibits any additional key entry, until the interrupt is honored by the processor. The system must be programmed to reset the lock-out condition with the appropriate start command. During the period the keyboard is in a lock-out condition, the display cursor is blanked out as an indication to the operator. The cursor is also blanked out when the program commands a lock-out via an IO instruction. None of the Interrupt Request keys have typamatic action.

## ATTN Key

Pressing this key normally readies the keyboard and display, ending any stopped state, however, the operation is defined by the application program.

ATTN																	DUPL	ERASE	EOS
																	DUPL	ERASE	EOS
PF4	PF1															INS	MODE	DEL	
PF5	PF2															↑	↓		
PF6	PF3	SHIFT	Z	X	C	V	B	N	M	7	8	9	SHIFT			ENTER			
		RESET																ENTER	

## ENTER Key

When the ENTER key is pressed, keyed-in data on the display normally enters the system and is available to be read by the program, however, the operation is defined by the application program. Before ENTER is pressed, any keyed-in data can still be changed by the operator.

ATTN																	DUPL	ERASE	EOS
																	DUPL	ERASE	EOS
PF4	PF1															INS	MODE	DEL	
PF5	PF2															↑	↓		
PF6	PF3	SHIFT	Z	X	C	V	B	N	M	7	8	9	SHIFT			ENTER			
		RESET																ENTER	



## Chapter 3. Programming Input/Output Operations

This chapter discusses the data flow to and from the 4979. Specific topics are commands, status information and I/O Instructions.

### Data Transfer Operations

Data is transferred between the processor and the attachment, in a parallel operation (16 data bits plus 2 parity bits). The number of data words transferred and the direction in which they move are determined by the I/O command. The I/O command also determines whether data is transferred to or from processor storage, under Direct Program Control (DPC) only, or under Direct Program Control and in Cycle Steal (CS) mode. The 4979 has a maximum data transfer of 1920 characters per Device Control Block (DCB). The processor storage buffer area may vary in size from 0 to 1920 bytes, the maximum number of character bytes contained in the display buffer of the attachment, with no beginning or ending boundary restrictions. There are no overrun conditions with the 4979. Data is sequentially transferred to or from the display buffer two bytes at a time. The beginning address of the processor data table may be an odd or even address to allow for program independent character alteration.

There is a one-to-one relationship between display screen positions and the display refresh buffer addresses. The data is transferred in EBCDIC (extended binary coded decimal interchange code). Entries from the keyboard are converted to EBCDIC within the attachment prior to being placed in the display buffer. See Appendix A for character to EBCDIC Conversion.

### Direct Program Control (DPC)

Under direct program control, only one word of data moves to or from processor storage at a time. After moving the data, the processor continues processing other instructions. Moving data under DPC does not cause interrupts.

### Cycle Steal

When data is moved to or from processor storage by stealing storage cycles (Cycle Steal mode), processing and I/O operations are overlapped. Overlapping allows the processor to execute other instructions while the display is performing I/O operations.

### Initiating a Display Operation

Every I/O instruction to the 4979 requires (in processor storage):

1. An Operate I/O instruction
2. An I/O command, device address, and an immediate data field

### Operate I/O Instruction

The following description is an overview of the Operate I/O Instruction (Figure 3-1). Refer to the *IBM Series/1 Model 5 4955 Processor and Processor Features Description*, GA34-0021 or the *IBM Series/1 Model 3 4953 Processor and Processor Features Description*, GA34-0022, for a more detailed description of the I/O Instruction.

All input/output operations from the processor to the display, are initiated by an I/O Instruction. An address field (bits 16–31) and the R2 field (bits 8–10) in the Operate I/O instruction point to a processor storage location containing an IDCB (Immediate Device Control Block). The IDCB is a two-word block of storage that contains device directed I/O commands. Before issuing the I/O instruction for an operation, the command field of the IDCB (bits 0–7) must be set, along with a device address (bits 8–15), and any field of immediate data required by the command in the IDCB (bits 16–31). The information specified in the immediate field depends on the command to be performed. The device address of the 4979 can be one of 128 (0–127) possible device address combinations. This address is determined by the Device Address field of the IDCB. Bit 8 of this field must be zero.

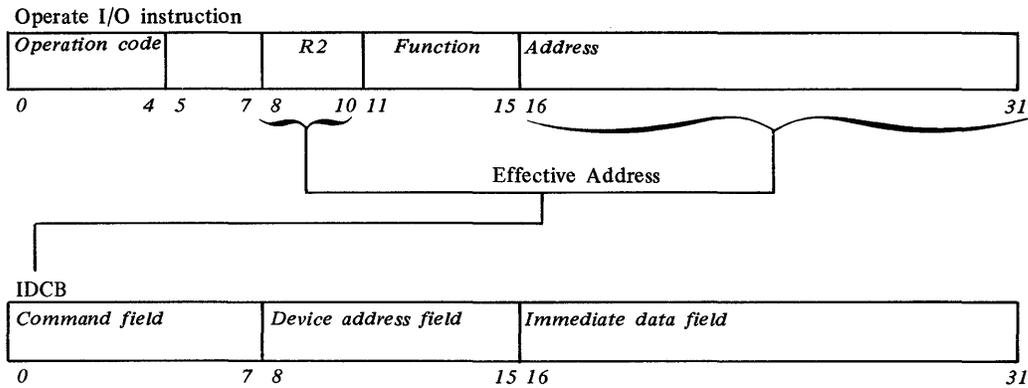


Figure 3-1. Operate I/O instruction

### Using the IDCB

An Immediate Device Control Block (IDCB) is required for every I/O command issued to the display. The format of the IDCB is shown in Figure 3-2. Before issuing an I/O instruction, an I/O command must be stored in the associated IDCB. The immediate data field of the IDCB should contain either a data word or a DCB address. I/O commands that execute under direct program control require a data word, while the commands that execute in cycle steal mode require a DCB address.

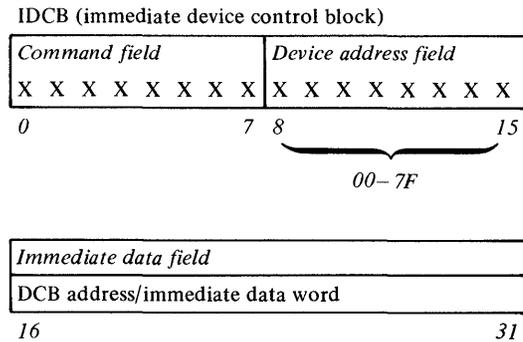


Figure 3-2. IDCB format

### Input/Output Commands and Display Operations

The I/O command, stored in the IDCB, determines whether a single word of data is transferred under direct program control only, or following a direct program control operation, additional words of data are to be transferred to the processor under cycle steal mode. The 4979 responds to the following I/O commands (defined in the command field (bits 0-7) of the IDCB):

#### Direct Program Control (DPC)

1. Prepare
2. Device Reset
3. Read Device ID

#### Cycle Steal Mode (CS)

1. Start
2. Start Cycle Steal Status

#### Command Execution Under DPC Mode

When the display executes a Prepare, Device Reset, or Read ID command, a single word of data is moved to or from the immediate data field of the IDCB in processor storage. These commands do not cause interrupts. After execution of the command, the 4979 reports a condition code that indicates whether the I/O operation succeeded or failed. See "Condition Codes" later in this chapter. Processing operations are halted while the I/O operation is in progress. See Figure 3-3.

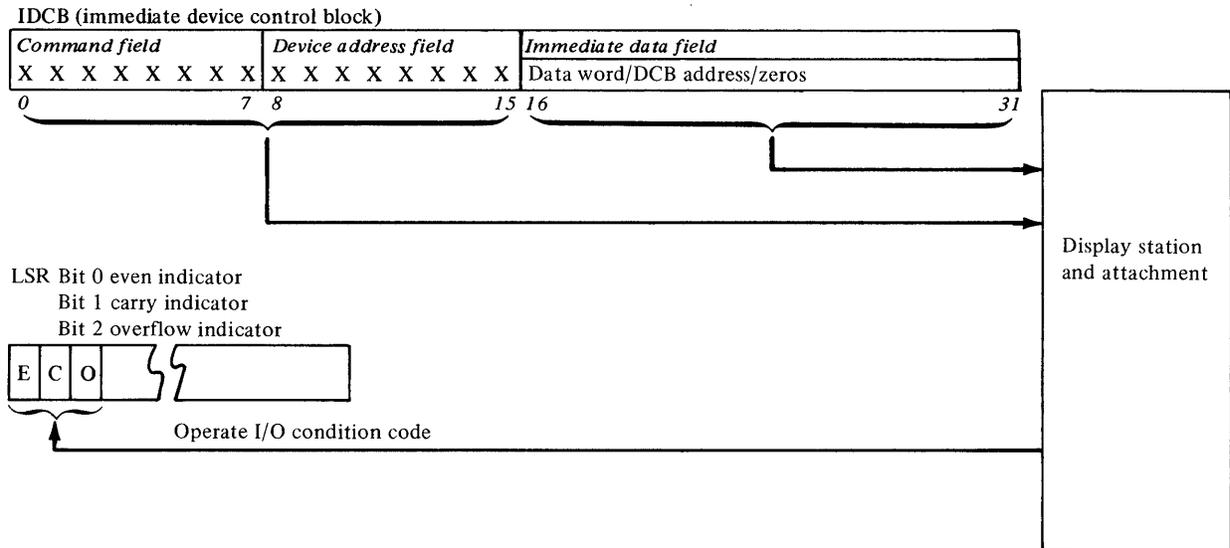
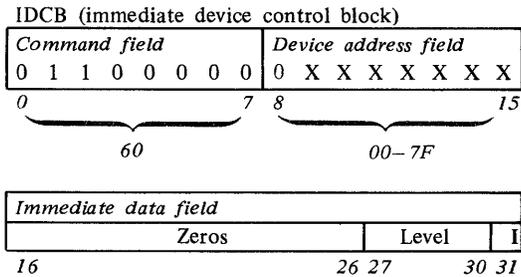


Figure 3-3. Command execution under DPC

**Prepare**

Before the display can execute interrupt causing commands, it needs interrupt parameters. These parameters, stored in the IDCB immediate field associated with a Prepare command, contain the level on which the attachment is to interrupt (bits 27–30), and an interrupt enable (bit 31), and are transferred to the attachment upon execution of the Prepare command. If the I bit (bit 31) equals 1, the display can interrupt. If the I bit equals 0, it cannot interrupt. The Prepare command operates under direct program control and does not cause an interrupt.

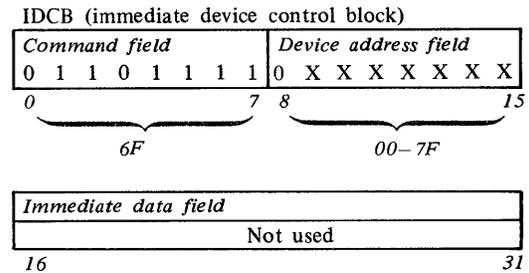


**Device Reset**

A Device Reset command will:

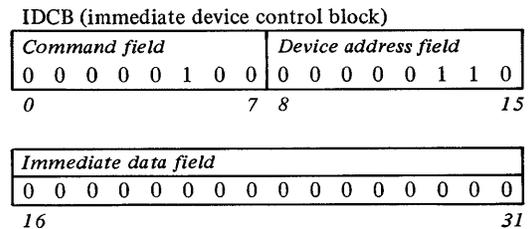
- Halt any Start Command or local operator function.
- Reset Control, Status and pending interrupts (excluding priority interrupts).
- Reset the screen to a non-insert mode.
- Enable the keyboard with the cursor displayed.
- Move the cursor to home position at the upper left-most character position on the screen.
- Allow no data characters to be altered or Prepare fields to be reset.

The command code and device address supply all needed information. The device reset command operates under direct program control and does not cause an interrupt. Although the IDCB immediate data field is not used or checked, the bits should be set to zero.



**Read Device ID**

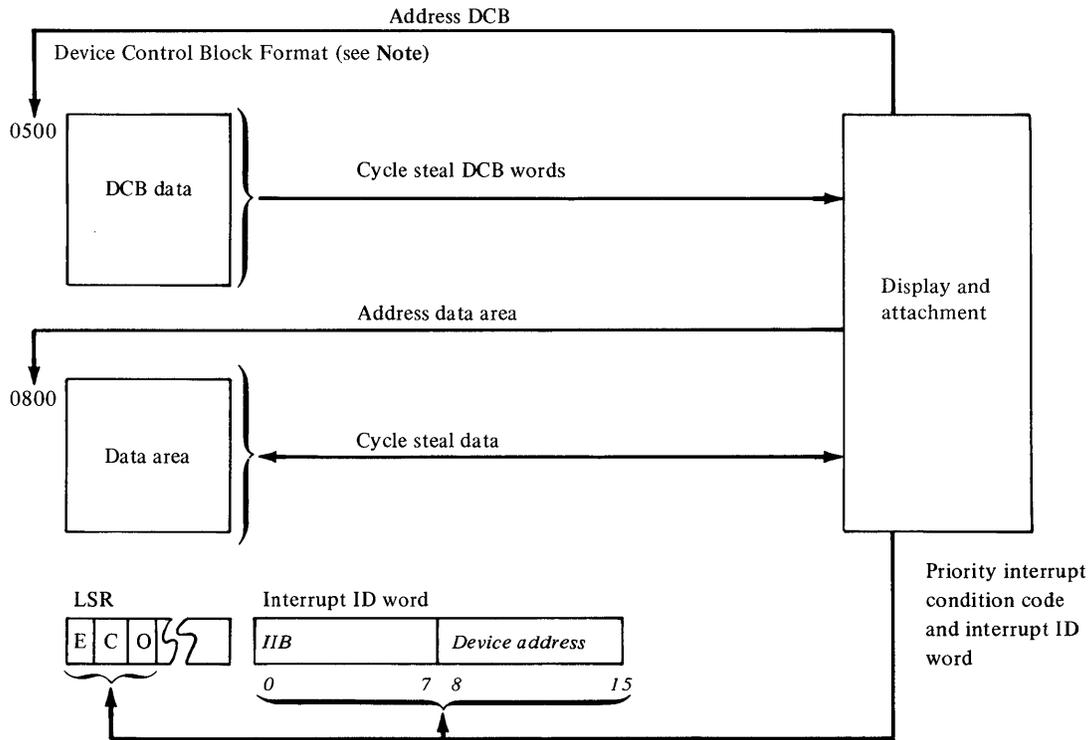
This command transfers the device ID word for the 4979 to the immediate field of the IDCB associated with that particular command. If the 4979 is busy or has an interrupt pending, condition code 1 is returned. The Read Device ID command operates under DPC and does not cause an interrupt. The device ID word for the display is:



### Command Execution in CS Mode

The Start and the Start Cycle Steal Status are interrupt-causing commands that move data in cycle steal mode. When the 4979 receives and accepts either of these commands, it reports a condition code to the processor and begins command execution. The processor continues with other operations while the attachment is 'busy' cycle stealing data for the I/O operation. When the I/O operation is completed, the attachment sends an interrupt request to the processor. At interrupt presentation time the attachment reports a condition code and transfers an interrupt ID word containing status information to the processor. See "Status Information" later in this chapter.

The immediate data field of an IDCB containing either a Start command or a Start Cycle Steal Status command must point to a Device Control Block (DCB). See Figure 3-4. The DCB must contain the control information and device parameters that are required to execute an I/O operation in cycle steal mode.



**Note.** DCB format is shown for a start command. The DCB format for a start cycle steal status command appears later in this chapter.

Figure 3-4. Command execution under CS mode

## Using the DCB

A Device Control Block (DCB), comprised of eight contiguous words in main storage, must be reserved for every I/O operation that moves data in cycle steal mode. All Start commands, Start Cycle Steal Status commands, and all display operations included in a DCB command chaining sequence require a separate DCB. Device parameters that define and control the I/O operation must be stored in each DCB.

## DCB Format

The following text will describe the bit significance of each of the eight contiguous data words in the DCB.

### Control Word—DCB Word 0

The control word is the first word of the DCB. It is a 16 bit word that explains the cycle stealing operation and contains two distinct bytes of control parameters. These parameters are used with the particular Start command to be performed. Figure 3-5 shows the DCB and Control Word 0.

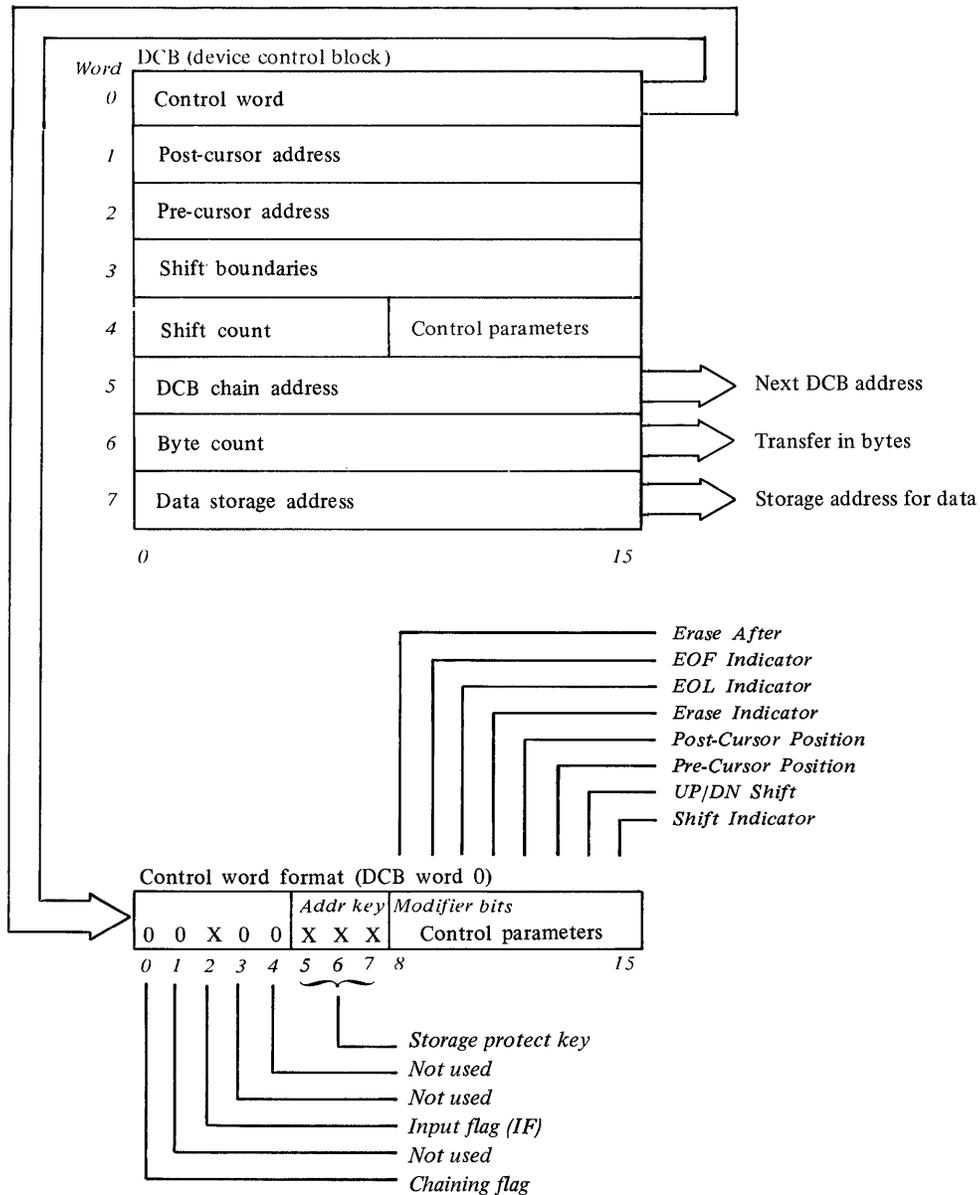


Figure 3-5. DCB and its control word

Bits 1, 3 and 4 of the control word are not supported and should be zero.

#### Bit 0—Chaining Flag

This bit indicates a command chaining operation. After completing the current DCB operation, the attachment will not interrupt but will fetch the next DCB pointed to by the Chain Address in DCB word 5. Command Chaining is valid only for a Start I/O command. It is ignored and not checked for the other commands.

#### Bit 2—I/F (Input Flag)

This bit indicates the direction of the cycle steal operation:  
0 = out of processor storage, 1 = into storage.

#### Bits 5, 6 and 7—Address Key

This is a three bit key presented by the attachment during data transfers to ascertain storage access authorization. An incorrect Address key will cause an exception interrupt.

Bits 8 through 15 (the second byte of the control word) are device dependent. The format for these control indicator bits is explained individually.

#### Bit 8—Erase After

If the Erase After bit = 1, the display buffer is erased from the current cursor location to the End of Line, End of Field, or End of Screen.

If the Erase After bit = 0, no erase operation will be performed. During an Output (bit 2 = 0) operation, if the Erase After bit is ON and: bits 9 (EOF) or 10 (EOL) are ON, then an erase after output is completed (after the Byte Count reaches zero) using the EOF/EOL bits as a termination control. If the EOF/EOL bits = 0, the erase routine will default by erasing to the EOS (End of Screen).

The Erase After bit is also used with bit 1 (Protect, Unprotect or P/U) of Device Parameter 4 to define which data characters should be erased.

*Note.* On an Input operation characters read from the refresh buffer are erased.

During an Input or Output operation:

If the Erase After bit = 1 and the P/U bit = 1, all characters (protected and unprotected) are erased.

If the P/U bit = 0, only unprotected characters are erased.

Either the EOF/EOL bits or the EOS condition are used with the Erase After function during an Input operation prior to the byte count reaching zero.

*Note.* Erased is defined as substituting unprotected NULL characters (Hex 00) into the attachment refresh buffer, instead of graphic characters. These NULL characters will appear as blank spaces on the display screen.

#### Bit 9—End of Field (EOF)

A field is a boundary where data characters change from unprotected to protected, and is used with the erase or other operations to define the termination of an operation.

If the EOF bit = 0, the end of field condition is not used during Input or Output operations.

If the EOF bit = 1 during an output operation, this bit is used as a terminator with bit 8 (Erase After) or bit 11 (Erase) during an Erase function.

If the EOF bit = 1 during an input operation, it is used as a boundary marker causing an early end to an Input operation, before the byte count reaches zero, and gives an exception interrupt with bit 0 (Status Available) ON in the Interrupt Status Byte (ISB).

#### Bit 10—End of Line (EOL)

This bit indicates the data character at the right most character position of a display line marked by the cursor location.

During an Input or Output operation:

If the EOL bit = 0, EOL condition is not used as a terminating condition.

During an Output operation:

If the EOL bit = 1, it is used as a terminator during an Erase or Erase After operation.

During an Input operation:

If the EOL bit = 1, it causes an early end to an input function, prior to the Byte Count reaching zero, and gives an Exception Interrupt with bit 0 (Status Available) ON in the ISB.

#### Bit 11—Erase

During an Output operation:

If the Erase bit = 1, this is a pure erase operation using the Byte Count, EOL, EOF, or EOS, as a controlling terminator. No data characters are transferred to or from processor storage.

The P/U bit is used with bit 11 to define which characters are erased.

During an Output operation:

If the Erase bit = 1 and the P/U bit = 1, all characters are erased. If the P/U bit = 0, only unprotected characters are erased. In this operation the Byte Count is decremented as each character is scanned, erased or not.

If either EOF, or EOL bits = 1, and either an EOF or EOL condition is reached before the Byte Count reaches zero, the Erase condition is halted and an exception interrupt occurs with bit 0 (Status Available) ON in the ISB. If both EOF/EOL bits = 0, and if an EOS is encountered before the byte count reaches zero, the Erase is halted and an exception interrupt occurs with bit 0 ON in the ISB.

*Note.* The erase bit cannot be ON during an Input operation. An erase operation may be coupled to a shift operation. The Shift routine will occur prior to the Erase.

#### Bit 12—Post-Cursor Position

During an Input or Output operation:

If the Post-Cursor bit = 0, the cursor is repositioned to the last recorded position prior to the current operation. This is done at the completion of the current operation.

If the Post-Cursor bit = 1, the cursor is repositioned at the cursor address given in DCB Word 1.

#### Bit 13—Pre-Cursor Position

During an Input or Output operation:

If the Pre-Cursor bit = 0, the current operation begins at the current cursor location.

If the Pre-Cursor bit = 1, this indicates that the present operation begins at the cursor address given in DCB Word 2.

#### Bit 14—UP/DN Shift

If the UP/DN bit = 0, the character lines will shift downward during the shift routine.

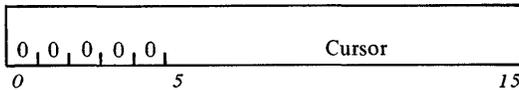
If the UP/DN bit = 1, the character lines will shift upward during the shift routine. The UP/DN shift (bit 14) is used when bit 15 = 1.

### Bit 15—Shift

During an Input or Output operation:

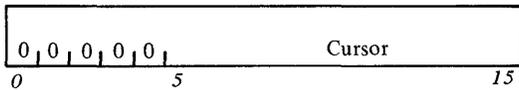
If the shift bit = 0, no line shift occurs. If the shift bit = 1, a line shift occurs prior to the current operation to be executed. Actual line shifting occurs only within the high and low boundaries as defined in Device Parameter 3. The number of shift operations is determined by the shift count defined in DCB Word 4. The upward or downward shift direction is defined by the UP/DN indicator (bit 14) of the DCB Control Word.

### Post-Cursor Address—DCB Word 1



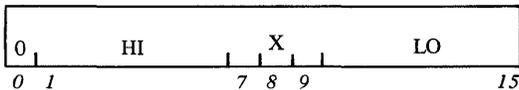
This word contains the post-cursor address. This is the address of the cursor at the completion of an I/O command, if the Post-Cursor bit of the DCB Control Word (control parameters) is ON. An address greater than 077F (Hex) will cause an Exception Interrupt with bit 3 (DCB Specification Check) ON in the ISB.

### Pre-Cursor Address—DCB Word 2



This parameter contains the beginning address of the display or pre-cursor address and will cause the cursor to move to the pre-cursor location on the display screen for the beginning of an operation (output, input erase etc). A cursor address of 0000 (Hex) = the top left-most character position on the display screen and an address of 077F (Hex) = the lower rightmost character position on the display screen. A pre-cursor address greater than 077F (Hex) will cause an Exception Interrupt with bit 3 (DCB Specification Check) ON in the ISB.

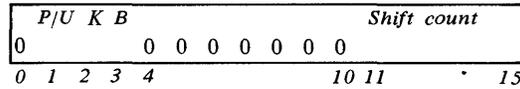
### Shift Boundaries—DCB Word 3



This word contains the parameters required during a shift operation. Lines are shifted upward or downward between the HI and LO boundary limits by duplicating the appropriate lines. Lines moved beyond these limits are lost because higher or lower lines are superimposed on them. The Shift Operation may be coupled with an Output (Write or Erase) or Input (Read) operation. The repeated lines, or any other lines may be operated on by the coupled Output/Input operation, together with any Erase After termination operation necessary by programming. The Shift Operation is always executed prior to any coupled I/O

operation. The HI boundary address must have a boundary value less than the LO boundary address, and the HI and LO addresses must contain exactly the seven most significant bits of the beginning address of the respective HI and LO boundary lines, otherwise, an Exception Interrupt will occur with bit 3 (DCB Specification Check) ON in the ISB.

### Shift Count and Control Parameters—DCB Word 4



DCB Word 4 contains additional control parameters to be used with the START I/O command.

Bit Explanation:

Bit 0 and bits 4 through 10 are not used and should always be zero to avoid future code obsolescence.

#### Bit 1—Display Protect (Protect/Unprotect)

During an Output operation:

If the P/U bit = 0, the data characters are marked unprotected in the display buffer, or only unprotected characters can be erased during an ERASE or ERASE AFTER operation.

If the P/U bit = 1, this indicates either data characters are protected in the display buffer, or all data characters are to be erased during an ERASE or ERASE AFTER operation.

During an Input operation:

If the P/U bit = 0, only unprotected data characters are read or read and erased.

If the P/U bit = 1, all data characters are read or read and erased.

#### Bit 2—KBD Lockout

This bit is used to enable (bit 2 = 0) or disable (bit 2 = 1) a keyboard electronic lockout condition at the completion of each DCB operation. This bit is set locally when any special function key is pressed causing an Attention Interrupt Request, provided the device has been prepared (I bit = 1). If the device is not in a prepared state (I bit = 0) and any special function key is pressed, no Interrupt Request to the processor will occur.

This condition is disabled (reset) by a: Power On Reset, Device Reset, System Reset, or an appropriate Start command.

#### Bit 3—Blank

This bit is used to blank (bit 3 = 1) or unblank (bit 3 = 0) all data characters including the cursor at the completion of each DCB operation. This does not imply that SPACE or NULL characters are substituted for data characters in the display refresh buffer, during an operation which utilizes the Blank bit.

#### Bits 11—15

These bits define the shift count used during the shift operation. This count is a binary value that defines the total number of lines to be shifted. Bit 15 is the least significant bit of the parameter. A value of 00000 indicates NO shift and a value of 10111 indicates a maximum shift of 23 lines.

A shift count with a value equal to or greater than the total number of lines contained within the shift window boundaries, or a shift count of zero, or a shift count greater than 23 will terminate the operation and cause an Exception Interrupt, with bit 3 (DCB Specification Check) ON in the ISB.

### Chain Address—DCB Word 5

This word contains the address of the next DCB to be used (chained to) when the current byte count is zero and the Chain Flag (bit 0) is ON in the DCB Control word. Abnormal ending sequences which cause Exception Interrupts disrupt the normal chaining operation. An Odd Chain Address will cause an Exception Interrupt with bit 3 (DCB Specification Check) ON in the ISB, and terminate the operation. The chaining address is not checked unless the chaining flag is ON in a valid control word.

### Byte Count—DCB Word 6

Byte Count contains the count of the number of bytes to be transmitted to or received from processor storage, using the operation specified in the DCB Control Word, or the number of sequential control operations used for an Erase command. A byte count greater than 1920 (HEX0780) will cause an Exception Interrupt with bit 3 (Specification Check) ON in the ISB. The byte count may be odd or even.

### Data Address—DCB Word 7

This word contains the address in processor storage where the data is to be fetched or stored for a transmit or receive operation.

For examples of Start I/O DCBs detailing the use of the device dependent bits contained in the control word (DCB Word 0) the DCB Word 4 (see Appendix B).

### Programming Considerations When Using the DCB

1. Only those words required for the cycle stealing operation are fetched. The contents of the words must be specified correctly.
2. The DCB address, chain address, and the status address must be even.

### DCB Command Chaining

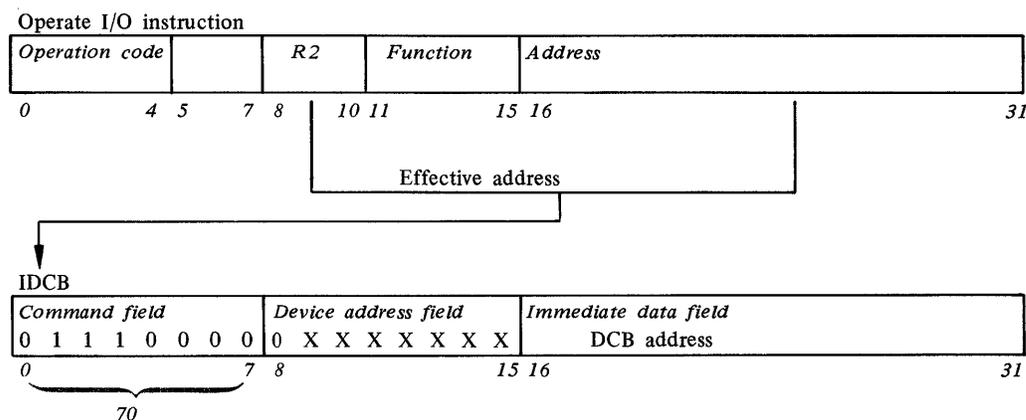
DCB command chaining is executed when the current DCB comes to a normal completion and a new DCB is fetched without issuing a new Operate I/O instruction. The DCBs belonging to such a sequence are said to be chained.

The first DCB in the chain contains the address of the next DCB. As each operation in the sequence is completed, the 4979 uses the chain address stored in the current DCB to select the next DCB. The chained-in DCB is examined to determine which operation is next in the sequence and whether the associated device parameters are valid. DCB command chaining operations continue until a DCB is fetched having the chaining bit in the control word (DCB word 0) set to zero. This indicates the last operation in the chain.

If an error or operation check occurs, chaining to succeeding DCBs is automatically suspended, and the attachment sends an interrupt request to the processor. The attachment does not request an interrupt until the last DCB operation in the chain is completed. By using command chaining, the processing time required to execute I/O operations is reduced. As stated previously, the 4979 responds to the Start and Start Cycle Steal Status commands in cycle steal mode.

### Start I/O Command

The Start command initiates all display operations that transfer data to or from processor storage under direct program control and then in cycle steal mode. When the Operate I/O instruction is issued, the Start command is transferred under direct program control from the IDCB to the display. While the display is 'busy' executing the I/O operation, the processor continues with other operations. Beginning at the DCB address specified in the IDCB, the eight words in the DCB are transferred to the display from processor storage. The data is transferred in cycle steal mode one word at a time. The DCB information is decoded and the display begins executing the operation called for in DCB control word (DCB word 0). Refer to "Control Word—DCB Word 0." When the operation (or operations when chaining) ends, an interrupt request is sent to the processor. At interrupt presentation time, a condition code and an interrupt ID word containing status information are presented to the processor. The format of the IDCB for a Start command is:



*Notes.*

1. A Start I/O command issued to the display while the device is Not-Ready, will cause an interrupt with condition code 2 (Exception) and bit 0 (Device Status Available) set in the ISB. The Start command will not be executed.
2. The 4979 uses coupled commands, which allow several operations to occur under any one Start I/O command.

*Example.* Shift, Read, and Erase. The coupled commands are executed via control indicators located in the DCB Control Word and Device Parameter 4. The order of execution of coupled commands within one Start command is:

**Output**

1. Shift followed by Erase
2. Shift followed by Write
3. Shift followed by Write followed by Erase After

**Input**

1. Shift followed by Read
2. Shift followed by Read followed by Erase After

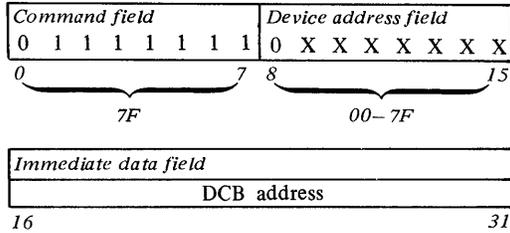
The Shift function is optional, and may be omitted for any Start I/O command. No other Start I/O command combination is available.

**Start Cycle Steal Status Command**

The Start Cycle Steal Status command initiates the transfer of up to 6 bytes of status information from the attachment to processor storage. This status information is used to determine why a given command did not execute properly. The processor storage address is specified in word 7 of the applicable DCB. This command operates under direct program control and then in cycle steal mode and causes the attachment to present an interrupt request when execution is complete.

The Start Cycle Steal Status command requires an Operate I/O instruction with the address of an IDCB, an IDCB with the address of the DCB, and a DCB. Figure 3-6 shows the formats of the IDCB, DCB and 6 bytes of status information.

**IDCB (immediate device control block)**



**DCB (device control block)**

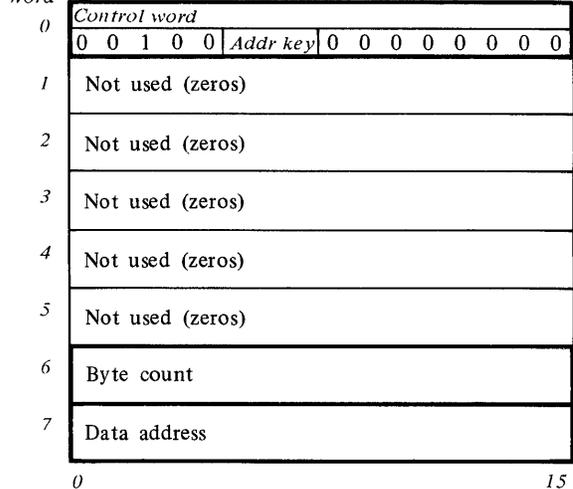


Figure 3-6. Start cycle steal status

**DCB Word Description**

- |                        |  |
|------------------------|--|
| Control Word           | No device dependent control bits are used. The I/O bit must be 1. Chaining is not supported for the Start Cycle Steal Status command. The cycle steal address key bits (5-7) are used. |
| DCB Words 1-5          | are not used and must be zero.   |
| Word 6<br>Byte Count   | The Byte Count must be 6 (0006) Hex, otherwise an Exception Interrupt will occur, with bit 3 (DCB Specification Check) ON in the ISB.  |
| Word 7<br>Data Address | The data storage location must be an even address, otherwise an Exception Interrupt will occur (CC-2), with bit 3 (DCB Specification Check) ON in the ISB.                             |

## Status Information

After execution of a given command under either DPC alone or under DPC and in cycle steal mode, status information will be reported to the processor for analysis of that command's execution. The three types of data that make up this status information are:

- Condition codes
- Interrupt ID word
- Cycle Steal Status

### Condition Codes

A condition code is reported to the processor for (1) every Operate I/O instruction and (2) upon presentation of a priority interruption. The condition code is available in the even, carry and overflow bit positions of the Level Status Register (LSR) in the processor. Refer to "Prerequisite Publications" in the Preface of this manual for order numbers of the IBM Series/1 Processor Unit Description manuals. For commands that do not cause interrupts, the condition code reported after the instruction is executed is the only status information required or available.

Condition codes reported at the completion of an Operate I/O instruction are:

Condition Code	Meaning
0	Device not attached
1	Busy
2	Busy after reset
3	Command reject
4	Not used
5	Interface data check
6	Not used
7	Satisfactory

#### Condition Code 0—Device Not Attached

Reported by the channel when the addressed device is not attached to the system.

#### Condition Code 1—Busy

Reported by the device when it is unable to execute a command because it is in the busy state. The device enters the busy state upon acceptance of a command that requires an interrupt for termination. It exits the busy state when the processor accepts the interrupt.

#### Condition Code 2—Busy After Reset

Reported by the display when it is unable to execute a command because of a reset and the device has not had sufficient time to return to the quiescent state. No interrupt occurs to indicate termination of this condition.

#### Condition Code 3—Command Reject

Reported by the display when:

1. A command is issued that is outside the device command set.
2. The device is in an improper state to execute the command.
3. The IDCB contains an incorrect parameter. For example: an odd byte DCB address, or an incorrect function/modifier combination.

When a cycle-steal device reports command reject, it does not fetch the DCB.

#### Condition Code 5—Interface Data Check

Reported by the display or the channel when a parity error is detected on the I/O data bus during a data transfer.

#### Condition Code 7—Satisfactory

Reported by the display when it accepts the command.

These condition codes are mutually exclusive and have a priority sequence. That is: beginning with CC=7, each successive condition code through CC=0 takes precedence over the previous code. For example, if the display cannot accept a command because it is busy, it reports CC=1, irrespective of error conditions encountered.

Condition codes reported during priority interruptions are:

Condition Code	Meaning
0	Not used
1	Not used
2	Exception
3	Device end
4	Attention
5	Not used
6	Not used
7	Not used

#### Condition Code 2—Exception

This code is reported whenever the boundary conditions EOF, EOL or EOS are reached prior to the byte count reaching zero, or the DCB parameters are incorrectly specified.

This interrupt is also used when a Delayed Command Reject, DCB Specification Check, Storage Data Check, Device Not Ready, Invalid Storage Address, Protect Check or Interface Data Check is detected during a cycle steal operation. The Interrupt Status Byte (ISB) further describes this condition.

#### Condition Code 3—Device End

This code is reported when no error exception or attention conditions occur during the I/O operation. A normal termination of the operation has occurred.

### Condition Code 4—Attention

This code is reported when a keyboard entry or device going Ready is detected.

Along with the interrupt condition code, the attachment also transfers an interrupt ID word which provides additional information on interrupting conditions.

### Interrupt Identification Word

Acceptance of an I/O interrupt causes the attachment to place an ID word in Register 7 (R7) of the interrupted level. For condition code 2, the Interrupt ID word consists of the Interrupt Status Byte (ISB) and the address of the interrupting device. For condition code 4, the Interrupt ID word consists of the Interrupt Information Byte (IIB) and the address of the interrupting device. The first byte will be zero for all other condition codes.

Interrupt ID word

IIB	Device address
X X X X X X X X	0 X X X X X X X
0	7 8 15

### Interrupt Information Byte

The 4979 uses an Interrupt Information Byte (ISB). This byte is associated with an Attention Interrupt (CC-4), caused by certain keyboard keys, or whenever the transition from a device Not-Ready to a device Ready status is detected.

### IIB Keyboard Format

- 0–3 Not used—returned as zero
- 4 Ready Status detected
- 5–7 Keyboard Code Bits

0 through 3 of the keyboard bits are not used but are always returned as 0. Bits 5, 6 and 7 are the code bits for the keyboard, entered by the 4979 during a keyboard interrupt. These bits represent the interrupt key that was activated by the operator, either ENTER, ATTN or one of the PROGRAM FUNCTION keys.

Bits 5 6 7

- 0 0 0 Enter
- 0 0 1 Attn
- 0 1 0 PF1 Lower
- 0 1 1 PF4 Upper
- 1 0 0 PF2 Lower
- 1 0 1 PF5 Upper
- 1 1 0 PF3 Lower
- 1 1 1 PF6 Upper

### Bit 4 Ready Status Detected

This bit is set ON whenever a Not-Ready to Ready transition condition occurs.

As a result of an Attention Interrupt, the user's program can examine the IIB bits (0–7) to determine exactly which interrupt key was entered by the local operator, or to detect the 4979 going to a Ready condition.

### Interrupt Status Byte

The ISB stores accumulated status information.

The format of the ISB is:

- 0—Device Dependent Status Available
- 1—Delayed Command Reject
- 2—(Not Used)
- 3—DCB Specification Check
- 4—Storage Data Check
- 5—Invalid Storage Address
- 6—Protect Check
- 7—Interface Data Check

Bit 2 of the ISB is not supported and should always be returned as zero.

### Bit 0—Device Dependent Status Available

Set ON when:

the display reaches a boundary condition (EOF, EOL, EOS) prior to the byte count reaching zero, and causes an early termination to the operation, or a Start command is issued when the 4979 is in a Not-Ready state or a Not-Ready is detected during a Start operation.

It is also set on when the DCB parameters of a Start command are incorrectly specified.

The status (residual address cursor address, status bits, DCB and operation checks) can be examined by issuing a Start Cycle Steal Status command.

### Bit 1—Delayed Command Reject

Set ON when:

the display cannot execute a command it has received. Included are all Write commands and any Start commands with modifiers other than 0001, 1101 or 1111.

### Bit 3—DCB Specification Check

Set ON when:

the display receives any incorrect DCB parameters necessary to perform the desired operation. The operation is terminated.

The cause of the DCB check can be determined by a Start Cycle Steal Status command, provided the DCB check was not the result of a Start Cycle Steal Status command.

### Bit 4—Storage Data Check

Set ON during:

a cycle steal output operation to indicate an incorrect parity from accessed storage. The operation is terminated and the parity error in processor storage is not corrected.

### Bit 5—Invalid Storage Address

Set ON during:

cycle steal I/O operations whenever the processor storage address presented by the attachment for data or DCB access exceeds the storage size on the system. The attachment records the status, and the operation is terminated. Condition Code 2 is reported at interrupt accept time.

**Bit 6—Protect Check**

Set ON when:  
the attachment attempts to access a processor storage location without the correct address key. Refer to the *IBM Series/1 Model 5 4955 Processor and Processor Features Description, GA34-0021*, or the *IBM Series/1 Model 3 4953 Processor and Processor Features Description, GA34-0022*. The operation is terminated. Condition Code 2 is reported at interrupt accept time.

**Bit 7—Interface Data Check**

Set ON when:  
a parity error is detected on a cycle steal data transfer. This condition can be detected by the attachment or by the channel. In either case the operation is terminated and an interrupt is reported to the processor.

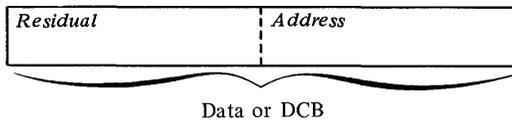
Condition Code 2 is presented at interrupt accept time (see Figure 3-1 for condition code and status information).

**Cycle Steal Status Words**

Six bytes of status information are transferred to the processor upon execution of the Start Cycle Steal command. Bits 0–15 from the data word position of the IDCB are transferred into the address register of the attachment and define the processor storage location of the DCB. The attachment will fetch the DCB from processor storage and cycle steal six data bytes into processor storage, beginning at the location specified by the DCB data address word.

The format of the six bytes returned to processor storage is described below:

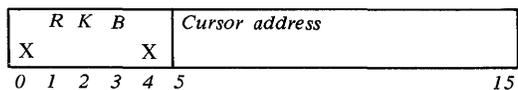
**Bytes 1 and 2**



Bytes one and two contain the Residual Address of the last previous attempted cycle steal, excluding a Start Cycle Steal Status command.

*Note.* The display is a byte oriented device, although it may cycle steal two bytes at a time (one word). In the particular case where the last previous cycle steal was attempted at an even byte address, and only one byte was transferred, the posted Residual Address will be even. For all other combinations of address and byte count the Residual Address will be odd.

**Word 1 (Bytes 3 and 4)**



Bytes three and four contain the address of the display cursor, and the display control status bits. No 4979 state changes or cursor alteration can occur as a result of executing this command.

Bit 0—Not used, should be zero.

**Bit 1—Ready Status**

If this bit is ON, it indicates that the Display/Keyboard is OFF, and that the unit is not in an operable state.

**Bit 2—Keyboard Status**

If this bit is ON, it indicates the keyboard is in a locked-out condition. The cursor is blanked as an indication to the operator of a keyboard lock. This condition may have resulted from a previous Start I/O command, the operator keying an interrupt request key, or the device going to a Ready condition.

During the interval when the keyboard is in a locked-out condition (bit 2 ON), the cursor address, keyboard status (bit 2) and display blank status (bit 3) will remain static until altered by an appropriate Start command. The Ready status (bit 4) will follow the condition of the Display/Keyboard.

**Bit 3—Display Blank Status**

If this bit is ON, it indicates the display screen is blanked, including the cursor.

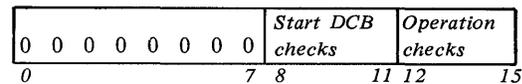
**Bit 4—Not used and must be zero.**

**Bits 5–15—Cursor Address**

The cursor address is an eleven bit binary value which represents the screen address of the cursor, with bit 15 as the least significant bit.

X Bits The X bits are not used but must be zero.

**Word 2 (Bytes 5 and 6)**



Byte 5 (bits 0–7) is not used and must always be zero. Byte 6 (bits 8–15) contains error recovery check bits associated with either DCB execution errors or operational checks.

DCB check bits 8–11 is a Hex coded number which delineates the particular type of error associated with an Exception Interrupt (CC-2) with a (DCB Specification Check) bit 3 in the ISB. The incorrectly specified DCB parameter checks are listed below:

- Bits 8–11 Start DCB Error
- Hex 0 No DCB Checks
- 1 Reserved
- 2 The Post-Cursor Address or the Pre-Cursor Address is greater than 1919 (77F Hex).
- 3 The LO or HI shift window boundary address is greater than 24 (73F Hex).
- 4 The LO or HI shift window boundary address is not an exact beginning address of a display line.
- 5 The HI shift window boundary address is greater than the LO shift window boundary address.
- 6 The HI shift window boundary address equals the LO shift window boundary address.
- 7 The shift count (DCB Device Parameter 4) equals zero (00 Hex).
- 8 The shift count is greater than 23 (17 Hex).
- 9 The shift count is greater than the total number of character lines contained in the shift parameter window range, LO Address and HI Address.
- A The DCB Chain Address is odd.
- B The byte count is greater than the maximum of 1920 allowed for a Start command.
- C–F Not defined.

Operational Checks (bits 12–15) is a Hex coded number which delineates the particular type of operational check associated with an Exception Interrupt (CC-2) and only bit 0 (Device Dependent Status Available) ON in the ISB.

Bits 12–15	Operational check
Hex 0	No operational check
1	The end of a field was detected prior to the byte count reaching zero during the operation with control word EOF indicator (bit 9) on.
2	The EOL was detected prior to the byte count reaching zero during the operation with control word EOL indicator (bit 10) on.
3	The EOS was detected prior to the byte count reaching zero during the operation.
4–F	Not defined.

### Status After Power and Resets

During Power On or Reset, the following actions occur:

- ISB register is reset
- Interrupt pending and busy status are cleared
- Cycle steal operations terminate
- Keyboard is unlocked and the Keyboard Disable status indicator is reset (Keyboard enabled with cursor displayed)

- Display screen is unblanked and the Blank status indicator is reset (Unblanked)
- Cursor is moved to home position at the upper left-most position on the screen
- Prepare information is reset
- All alphanumeric data in the refresh buffer is altered to a NULL code.
- Registers associated with cycle steal operations become invalid and must be reestablished prior to the next Start instruction. The Residual Address register is reset to all zeros.
- The display is reset to a Non-Insert Mode.
- An internal check is performed on the attachment.

During System Reset the following actions occur:

- The cursor is moved to home position at the upper left-most position on the screen.
- The display is set to a Non-Insert Mode.
- The keyboard status is enabled (no lock-out) with the cursor displayed.
- No data characters in the display refresh buffer are altered and any local operator function or Start command is immediately halted.
- The display screen status is unblanked.



## Appendix A. Characters—EBCDIC Hexadecimal Equivalents

<i>Character</i>	<i>EBCDIC Code (Hex)</i>	<i>Character</i>	<i>EBCDIC Code (Hex)</i>
A	C1	6	F6
B	C2	7	F7
C	C3	8	F8
D	C4	9	F9
E	C5	Space	40
F	C6	¢ Cent Sign	4A
G	C7	. Period	4B
H	C8	< Less than	4C
I	C9	( Left Parenthesis	4D
J	D1	+ Plus	4E
K	D2	Logical OR	4F
L	D3	& Ampersand	50
M	D4	! Exclamation Point	5A
N	D5	\$ Dollar Sign	5B
O	D6	* Asterisk	5C
P	D7	) Right Parenthesis	5D
Q	D8	; Semicolon	5E
R	D9	⌋ Logical NOT	5F
S	E2	- Minus-Hyphen	60
T	E3	/ Slash	61
U	E4	\ Reverse Slash	E0
V	E5	, Comma	6B
W	E6	% Percent	6C
X	E7	_ Underscore	6D
Y	E8	> Greater than	6E
Z	E9	? Question mark	6F
0	F0	: Colon	7A
1	F1	# Number	7B
2	F2	@ At	7C
3	F3	' Prime Apostrophe	7D
4	F4	= Equal	7E
5	F5	" Quotation mark	7F



.

.



.

.



## Appendix B. Start I/O Examples

### DCB Start I/O Examples

Following are six examples detailing the use of the device dependent bits contained in the DCB Control Word and Device Parameter 4.

- Example 1*    START (Read From Buffer) Input
- Example 2*    START (Set Cursor and Device States) Output
- Example 3*    START (Write to Buffer/Protected) Output
- Example 4*    START (Write/Erase—After to End of Line) Output
- Example 5*    START (Shift Lines/Write Protected/Erase After) Output
- Example 6*    START (Erase Unprotected) Output

#### START (Read From Buffer) Input

Device Parameter 4			Control Word		
Bits 0	N/A	Off	Bits 8	Erase-After	Off
1	P/U	Off	9	EOF	Off
2	Kyb-Lo	Off	10	EOL	Off
3	Blank	Off	11	Erase	Off
4–10	N/A	Off	12	Post-Cursor	On
11–15	Shift Count	Off	13	Pre-Cursor	On
			14	U/D	Off
			15	Shift	Off

Byte Count = XXXX (Hex) as required

This command moves the cursor to the position designated by Device Parameter 2 on the display screen and reads the data character from the display refresh buffer designated for that position. If the character is a NULL character a Hex 00 will be sent to processor storage. If the data character is not a NULL character, the EBCDIC character, if unprotected, will be sent to processor storage. Only unprotected characters are stored in contiguous processor storage; protected characters are skipped over and not transferred to processor storage. The Byte Count is decremented for each unprotected character (byte count is not decremented when protected characters are skipped over), including NULLs. Subsequently, the cursor is incremented one character position to the right and this routine is repeated until the Byte Count reaches zero.

At the completion of the data transfer, the cursor is moved to the position designated by Device Parameter 1 on the display screen. The 80th data character position of each line wraps to the first character position of the next lower line during the operation.

If an EOS condition is detected prior to the Byte Count reaching zero, an Exception Interrupt occurs. The keyboard is unlocked (enabled), and the display screen is unblanked at the completion of the data transfer.

#### START (Set Cursor and Device Status) Output

Device Parameter 4			Control Word		
Bits 0	N/A	Off	Bits 8	Erase-After	Off
1	P/U	Off	9	EOF	Off
2	Kyb-Lo	On	10	EOL	Off
3	Blank	On	11	Erase	Off
4–10	N/A	Off	12	Post-Cursor	On
11–15	Shift	Off	13	Pre-Cursor	Off
			14	U/D	Off
			15	Shift	Off

Byte Count = 0000 (Hex)

This command cycle steals only the DCB from processor storage. The cursor is moved to the post-cursor position as designated by Device Parameter 2, the keyboard is locked out (disabled), and the display screen blanked at the completion of the command.

#### START (Write to Buffer/Protected) Output

Device Parameter 4			Control Word		
Bits 0	N/A	Off	Bits 8	Erase-After	Off
1	P/U	On	9	EOF	Off
2	Kyb-Lo	Off	10	EOL	Off
3	Blank	Off	11	Erase	Off
4–10	N/A	Off	12	Post-Cursor	Off
11–15	Shift Count	Off	13	Pre-Cursor	On
			14	U/D	Off
			15	Shift	Off

Byte Count = XXX (Hex) as required

This command moves the cursor to the pre-cursor position on the screen, and reads data characters from processor storage. All of the EBCDIC characters are marked protected, except NULL (Hex 00) characters, and are subsequently written into the display refresh buffer designated for that cursor position. A protected character cannot be altered by the local user via the keyboard.

If the data is a NULL character, an unprotected NULL character is written at that position, and the cursor is incremented one position to the right until the Byte Count reaches zero. At the end of the previous operation, the cursor is repositioned at the last previous cursor position prior to the output operation.

*Note.* Incorrect EBCDIC characters are translated to NULL codes and entered into the display buffer as Hex–00.

The 80th character position of each line wraps to the first character position of the next lowest line. If an EOS condition is detected prior to the Byte Count reaching zero, an Exception Interrupt occurs. The keyboard is unlocked and the display screen is unblanked at the completion of the command.

**START (Write/Erase-After to End of Line) Output**

Device Parameter 4			Control Word		
Bits 0	N/A	Off	Bits 8	Erase-After	On
1	P/U	On	9	EOF	Off
2	Kyb-Lo	Off	10	EOL	On
3	Blank	Off	11	Erase	Off
4-10	N/A	Off	12	Post-Cursor	On
11-15	Shift	Off	13	Pre-Cursor	On
			14	U/D	Off
			15	Shift	Off

Byte Count = 0001 (Hex)

This command cycle steals only one data byte from processor storage, which contains one EBCDIC data character. The cursor is moved to the pre-cursor position as designated by Device Parameter 2, and the one protected data character is placed into buffer storage at the cursor position. All positions to the right of the cursor and to the end of the current line are filled with unprotected NULL characters. A NULL character has the effect of erasing, since it is equivalent to a SPACE character on the screen.

At the completion of the command, the cursor is positioned at the address located in Device Parameter 1, the keyboard is unlocked, and the display screen is unblanked.

**START (Shift Lines/Write Protected/Erase After) Output**

Device Parameter 4			Control Word		
Bits 0	N/A	Off	Bits 8	Erase-After	On
1	P/U	On	9	EOF	Off
2	Kyb-Lo	Off	10	EOL	On
3	Blank	Off	11	Erase	Off
4-10	N/A	Off	12	Post-Cursor	On
11-15	Shift Count 01 (Hex)		13	Pre-Cursor	On
			14	U/D	On/Off
			15	Shift	On

Byte Count = 2

This command cycle steals the DCB and causes the display lines to shift upward or downward, as directed by the U/D bit, but within the Hi (top line) and Lo (bottom line) boundaries contained in Device Parameter 3 of the DCB. One line is shifted as indicated in DCB Device Parameter 4. Lines shifted out of the boundary are lost, while the line shifting in from the opposite end remains intact to be

operated on (if so desired) by the current coupled output operation. At the completion of the shift operation, the 4979 cycle steals data words from processor storage according to the remaining parameter words in the DCB. In this case, two bytes.

If the Byte Count (Count) were zero, no data characters would be transferred from processor storage.

The data characters are sequentially transferred and inserted into the display buffer, beginning at the pre-cursor coordinate, indicated in Device Parameter 2, and continuing until the Byte Count reaches zero. The data, except NULL characters, is stored as protected characters. At the completion of the data transfer and store operation, the current line is filled with NULL characters to the end of the line, and the cursor is repositioned at the post-cursor coordinate indicated in Device Parameter 1. The keyboard is unlocked and the display screen unblanked at the completion of the command.

**START (Erase/Unprotected) Output**

Device Parameter 4			Control Word		
Bits 0	N/A	Off	Bits 8	Erase-After	Off
1	P/U	Off	9	EOF	Off
2	Kyb-Lo	Off	10	EOL	Off
3	Blank	Off	11	Erase	On
4-10	N/A	Off	12	Post-Cursor	On
11-15	Shift	Off	13	Pre-Cursor	On
			14	U/D	Off
			15	Shift	Off

This command moves the display cursor to the pre-cursor position designated by Device Parameter 2 on the display screen and reads the data character from that position. If the data character is marked protected, the cursor is moved to the next character position and the operation is repeated. If the data character is marked unprotected, a NULL character (Hex 0) is inserted for that position, the cursor is indexed to the next adjacent position and the routine is repeated until the Byte Count reaches zero. The Byte Count is decremented for each character position during the operation, regardless of the character being erased or not. The end of each line wraps to the beginning of the next lower line. If the End of Screen condition is detected prior to the Byte Count reaching zero, an Exception Interrupt occurs.

This operation has the effect of erasing all data characters on the screen or any portion that had previously been declared unprotected by the program, in one operation, with no data transfer to or from processor storage. The keyboard is unlocked and the display screen is unblanked at the completion of the command.

- address field 3-1
- address key 3-6
- address locations, buffer 2-1
- alphabetic characters 2-4
- attachment feature functional description 1-1
- attention, condition code 4 3-11
- ATTN 2-9
  
- basic components 1-2
  - display screen 1-2
  - keyboard 1-2
    - graphic alphanumeric keys 1-2
    - interrupt request keys 1-2
    - local function keys 1-2
    - shift/lock keys 1-2
- blank bit 3-7
- blanked cursor 2-2
- brightness control 1-3
- buffer 2-1
- buffer address locations 2-1
- busy, condition code 1 3-10
- busy after reset, condition code 2 3-10
  
- cathode ray tube 1-2
- chaining flag 3-6
- characters, alphabetic 2-4
- characters—EBCDIC hexadecimal equivalents A-1
- characters, graphic 2-4
- characters, numeric 2-4
- command, device reset 3-3
- command, prepare 3-3
- command, read device ID 3-3
- command, start cycle steal status 3-9
- command, start I/O 3-8
- command chaining, DCB 3-8
- command execution in CS mode 3-4
  - start cycle steal status command 3-9
    - status words, cycle steal 3-12
  - start I/O command 3-8
    - erase/unprotected output B-2
    - read from buffer input B-1
    - set cursor and device status output B-1
    - shift lines/write protected/erase after output B-2
    - write/erase-after to end of line output B-2
    - write to buffer/protected output B-1
- using the DCB 3-5
  - DCB command chaining 3-8
  - programming considerations when using the DCB 3-8
- command execution under DPC mode 3-2
  - device reset command 3-3
  - prepare command 3-3
  - read device ID command 3-3
- command reject, condition code 3 3-10
  
- commands, coupled 3-9
- commands, interrupt-causing 3-4
- commands, non-interrupt causing 3-3
- commands and display operations, input/output 3-2
- components, basic 1-2
- condition codes 3-10
- contrast control 1-3
- control word—DCB word 0 3-5
- controls 1-3
- coupled commands 3-9
- CRT 1-2
- CS 3-2
- CS mode, command execution in 3-4
- cursor 2-2
  - cursor positioning keys 2-5
    - down cursor 2-6
    - left cursor 2-6
    - new line cursor 2-6
    - right cursor 2-7
    - tab backward 2-6
    - tab forward 2-5
    - up cursor 2-6
- cursor address 3-12
- cursor wrap 2-6
  
- data field, protected 2-1
- data field, unprotected 2-1
- data fields 2-1
- data representation 2-2
- data transfer 3-1
- DCB 3-5
- DCB, programming considerations when using the 3-8
- DCB, using the 3-5
- DCB format 3-5
  - byte count—DCB word 6 3-8
  - chain address—DCB word 5 3-8
  - data address—DCB word 7 3-8
  - post-cursor address—DCB word 1 3-7
  - pre-cursor address—DCB word 2 3-7
  - shift boundaries—DCB word 3 3-7
  - shift count and control parameters—DCB word 4 3-7
- DCB start I/O examples B-1
- DCB words 3-5
- DEL 2-9
- DELETE key 2-9
- device end, condition code 3 3-10
- device not attached, condition code 0 3-10
- device reset 3-3
- direct program control 3-2
- display images, unformatted and formatted 2-1
- display operations 2-1
- display protect bit 3-7
- display station functional description 1-1
- dot matrix 1-2

down cursor 2-6  
downshift 2-4  
DPC 3-1, 3-2  
DPC mode, command execution under 3-2  
DUPC 2-7  
DUPL 2-7  
duplicate character key 2-7  
duplicate line key 2-7

EBCDIC 3-1  
end of field 3-6  
end of line 3-6  
ENTER key 2-9  
EOF 2-7  
EOL 2-7  
EOS 2-7  
exception, condition code 2 3-10  
expansion unit, 4959 input/output 1-1

features, standard 1-2  
flashing cursor 2-2  
format, DCB 3-5  
formatted images, unformatted and 2-1  
functional description, attachment feature 1-1  
functional description, display station 1-1

graphic alphanumeric keys 2-3

HI and LO boundary 3-12

I/O instructions 3-1  
IDCB 3-2  
IDCB, using the 3-2  
identification word, interrupt 3-11  
IIB 3-11  
immediate device control block 3-1  
immediate data field 3-2  
input flag 3-6  
input/output 3-2  
INS 2-8  
interface data check, condition code 5 3-10  
interrupt-causing commands 3-4  
interrupt information byte 3-11  
introduction 1-1

KBD lockout bit 3-7  
keyboard 2-3  
    graphic alphanumeric keys 2-3  
        alphabetic characters 2-4  
        graphic characters 2-4  
        numeric characters 2-4  
        space bar 2-5  
    interrupt request keys 2-9  
        ATTN key 2-9  
        ENTER key 2-9  
        program function keys 2-10

keyboard (continued)  
    local function keys 2-5  
        delete key 2-9  
        duplicate character key 2-7  
        duplicate line key 2-7  
        erase to end of field key 2-7  
        erase to end of line key 2-7  
        erase to end of screen key 2-7  
        insert mode key 2-8  
        reset key 2-9  
    shift/lock keys 2-3  
keyboard character set 1-2  
keyboard operations 2-2

left cursor 2-6  
level status register 3-10  
local function keys 2-5  
lock key 2-3  
lock out feature, electronic 1-2  
LSR 3-10

mode, formatted 2-1  
mode, unformatted 2-1  
model 3 4953 processor 1-1  
model 5 4955 processor 1-1

new line cursor 2-6  
non-interrupt causing commands 3-3  
NULL character 2-2  
numeric characters 2-4

on-off knob 1-3  
operation and instruction 2-1

PF 2-10  
positioning keys, cursor 2-5  
post-cursor position 3-6  
power and resets, status after 3-12  
pre-cursor position 3-6  
preface v  
prepare 3-3  
processor, model 3 4953 1-1  
processor, model 5 4955 1-1  
program function keys 2-10  
programming input/output operations 3-1  
    data transfer operations 3-1  
        cycle steal 3-1  
        direct program control 3-1  
    initiating a display operation 3-1  
    input/output commands and display operations 3-2  
    operate I/O instruction 3-1  
        using the IDCB 3-2  
protect check 3-12  
protected data field 2-1

read device ID 3-3  
refresh rate 1-2  
repeat keys (*see* typamatic keys)  
reset, device 3-3  
RESET key 2-9  
right cursor 2-7

satisfactory, condition code 7 3-10  
shift bit 3-7  
shift count 3-7  
space 2-5  
standard features 1-2  
    character set 1-2  
    cycle steal 1-2  
    data protect feature 1-2  
    electronic lock out feature 1-2  
start cycle steal status 3-9  
start I/O examples B-1  
start I/O examples, DCB B-1  
status after power and resets 3-12  
status information 3-10  
    condition codes 3-10  
        condition code 0—device not attached 3-10  
        condition code 1—busy 3-10  
        condition code 2—busy after reset 3-10  
        condition code 2—exception 3-10  
        condition code 3—command reject 3-10  
        condition code 3—device end 3-10  
        condition code 4—attention 3-11  
        condition code 5—interface data check 3-10  
        condition code 7—satisfactory 3-10  
        IIB keyboard format 3-11  
    cycle steal status words 3-12  
    interrupt identification word 3-11  
    interrupt information byte 3-11  
    interrupt status byte 3-11  
    status after power and resets 3-12

tab (backward) 2-5  
tab (forward) 2-6  
tube, cathode ray 1-2  
typamatic 2-3  
typamatic action 2-3

underscore 2-2  
unformatted and formatted display station 2-1  
    formatted mode 2-1  
        protected data field 2-1  
        unprotected data field 2-1  
    unformatted mode 2-1  
unprotected data field 2-1  
up cursor 2-6  
UP/DN shift 3-6  
uppercase characters 2-4

wide arrow (*see* lock key)

4959 input/output expansion unit 1-1



IBM Series/1 4979 Display  
Station Description  
GA34-0026-1

READER'S  
COMMENT  
FORM

**YOUR COMMENTS, PLEASE . . .**

Your comments assist us in improving the usefulness of our publications; they are an important part of the input used in preparing updates to the publications. All comments and suggestions become the property of IBM.

Please do not use this form for technical questions about the system or for requests for additional publications; this only delays the response. Instead, direct your inquiries or requests to your IBM representative or to the IBM branch office serving your locality.

Corrections or clarifications needed:

Page	Comment
------	---------

Cut or Fold Along Line

What is your occupation? \_\_\_\_\_

Number of latest Technical Newsletter (if any) concerning this publication: \_\_\_\_\_

Please indicate your name and address in the space below if you wish a reply.

---

---

---

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A.  
(Elsewhere, an IBM office or representative will be happy to forward your comments.)

**Your comments, please . . .**

This manual is part of a library that serves as a reference source for IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Cut Along Line

Fold

Fold

**Business Reply Mail**  
No postage stamp necessary if mailed in the U.S.A.

First Class  
Permit 40  
Armonk  
New York



IBM Corporation  
Systems Publications, Dept 27T  
P.O. Box 1328  
Boca Raton, Florida 33432

IBM Series/1 4979 Display Station Description (S1-03) Printed in U.S.A. GA34-0026-1

Fold

Fold



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



International Business Machines Corporation

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

IBM Series/1 4979 Display Station Description (S1-03) Printed in U.S.A. GA34-0026-1

GA34-0026-1

THE COMPUTER MUSEUM HISTORY CENTER



1 026 2097 4