Publication Number
GC31-2069-1

4700 Finance Communication System

Controller Programming Library

Volume 4
Loop and DCA Device Programming

IBM

4700 Finance Communication System

Controller Programming Library

Volume 4
Loop and DCA Device Programming

Publication Number
GC31-2069-1

File Number
S370/4300/8100/S34-30

Second Edition (January 1984)

This edition applies to Release 3 of the 4700 Finance Communication System and all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters (TNLs).

Changes occur often to the information herein; before using this publication in connection with the installation or operation of IBM equipment, consult the latest *IBM System/370 Bibliography of Industry Systems and Application Programs*, GC30-0370, for the editions that are applicable and current.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. Address comments concerning the contents of this manual to IBM Corporation, Information Development, Department 78C, 1001 W. T. Harris Blvd., Charlotte, NC 28257, USA. IBM may use or distribute any information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This is Volume Four of the *4700 Controller Programming Library*—one of a set of six volumes for the 4700 programmer. The table on page v summarizes the topics covered in the other volumes. All six volumes are available from your IBM representative or local IBM office under a single order number (GB0F-1387).

Who Should Read This Book

The manual is intended for those who must program terminal and device functions. The device information is also intended for those operations specialists who will create operation specifications for individual work stations and operators.

How This Book is Organized

This book has two parts. Part 1 is a general guide to device and terminal I/O programming; it contains chapters for general types of loop or DCA device programming, and ends with a chapter describing the 4700 assembler terminal and device instructions you must use. Part 2 comprises device-oriented chapters for each model device that attaches to the 4700 controllers. These chapters discuss the unique programming aspects of each device.

Following Part 2 are the appendixes:

Appendix A describes the machine instruction formats.

Appendix B defines the COPY instruction parameter lists for device-related operations.

Appendix C describes the program check codes.

Appendix D defines the terminal and device status codes and meanings.

Appendix E is a reference to the device statistical counter descriptions.

Appendix F describes the DATSM sample program and error codes.

The table at the end of this preface summarizes the topics covered in this and the other *Controller Programming Library* volumes. All six volumes are available from your IBM representative or branch office under the single order number GBOF-1387.

What Else to Read

Before using this book, you should be familiar with the following information:

1. OS/VS-DOS/VSE-VM/370 Assembler Language

2. IBM 4700 Finance Communication System, Controller Programming Library, Volume 1: General Controller Programming

3. IBM 4700 Finance Communication System, System Summary

VOLUME 1: GENERAL CONTROLLER PROGRAMMING (GC31-2066)
- Coding Considerations
- Instruction Categories
- Coding Rules
- Instruction Descriptions
- Machine Instruction Formats
- Copy Files
- Error Messages, Program Check Codes, and Status Codes
- Functions Retained for 3600 Compatibility
- Program Communication with the System Monitor

VOLUME 2: DISK AND DISKETTE PROGRAMMING (GC31-2067)
- Basic Disk and Diskette Programming
- Extended Disk and Diskette Access Method
- Instruction Descriptions
- Machine Instruction Formats
- Copy Files
- Status Codes

VOLUME 3: COMMUNICATION PROGRAMMING (GC31-2068)
- SNA/SDLC Communication Programming
- SNA/SDLC Communication Macros
- SNA/SDLC Communication Instructions (Reference)
- BSC3 Host Communication Programming
- BSC3 Communication Instructions (Reference)
- Communication Status Codes
- Communication Parameter List Reference
- Communication Machine Instruction Formats

VOLUME 4: LOOP AND DCA DEVICE PROGRAMMING (GC31-2069)
- General Protocols for Displays
- 4704 and 3604 Displays
- 3270-Compatible Displays and Devices
- 3606 and 3608 Financial Services Terminals
- General Protocols for Printers
- 4710 and 4720 Printers
- 3610, 3611, and 3612 Printers
- 3615 and 3616 Printers
- 3270-Compatible Printers
- 3624 Consumer Transaction Facilities
- Data Stream Mapping (DATSM) Protocols
- Device Status Codes
- Device Parameter List Reference

VOLUME 5: CRYPTOGRAPHIC PROGRAMMING (GC31-2070)
- Cryptographic Concepts and Facilities
- Enciphering and Deciphering Operations
- Generating and Exchanging Cryptographic Keys
- Authenticating Messages
- Validating and Translating PINs
- Using the Encrypting PIN Keypad
- Host Support Encryption Routines (BDKDPRS and BDKDES)
- Cryptographic Programming Instructions (Reference)
- System Cryptography
- Cryptographic Machine Instruction Formats
- Cryptographic Parameter List Reference
- Cryptographic Program Checks and Status Codes

VOLUME 6: CONTROL PROGRAM GENERATION (GC31-2071)
- Overall View of Control Program Generation (CPGEN)
- Sample CPGEN
- CPGEN Macro Statements (Reference)
- Using the Local Configuration Facility (LCF)
- CPGEN Messages
- LCF Messages

Figure 0-1. 4700 Controller Programming Library (GBOF-1387)

Summary of Amendments

| GC31-2069-1 (January, 1984)

This edition replaces GC31-2069-0. Significant changes and additions to this manual are marked with the same change bar you see at the left of this summary entry.

This edition supports the following device cluster adapter (DCA)-attached devices that are new for Release 3:

- IBM Magnetic Stripe Reader (Model 600) for 75- and 210-bpi data.
- IBM Magnetic Stripe Reader/Encoder (Model 200) improvements.
- 16-2/3 condensed printing for the IBM 4720 Printer.
- IBM 3178 Display Station
- IBM 5210 Printer, Models G01 and G02
- IBM Personal Computer and Personal Computer/XT
- IBM Displaywriter

This edition also describes the new 16-2/3 cpi printing capability on the B-loop–attached 4720 printer.

Besides the above changes, this edition also includes many editorial changes and corrections, particularly in the 3270 data stream processing information (Chapters 4 and 5, and Appendixes C and F).

Contents

Figures

Part I. Programming Guide

The chapters in this part of the book discuss general loop and DCA device programming. Each chapter except the last describes programming concepts for a general device type or function. The last chapter of Part I describes the 4700 Assembler instructions you use to perform loop and DCA control and input/output (I/O) operations.

Chapter 1. General Terminal and Device Programming

This chapter provides information about the instructions used to communicate with terminals and the controller facilities that support those terminals. The *3614 Programmer's Guide,* GC66-0002, and the *3624 Programmer's Guide,* GC66-0008, contain information about communicating with the 3614 and 3624. For information on data stream mapping for the 3262, 3278, 3279, 3287, and 4704-2/-3, refer to Volume 3.

Attaching Terminals and Devices to The Controller

Some 4700 terminals attach to the controller by a loop. Other terminals such as the 4704 Models 2 and 3, 3262, 3278, 3279, and 3287 attach to the controller by a device cluster adapter (DCA) and can operate through the 3270 Data Stream Mapping (DATSM) facility. The loop provides a data path both to and from the terminals or terminal components. These local loops can serve as remote loops when 3603 Terminal Attachment Units provide the connections to the telecommunication links. (You must specify the loop types—local or remote—and remote loop speeds—600[1] or 1200 bps—when you order the controller.)

The local loop speeds (600[1], 1200, 2400, and 4800 bps) are set by your service or installation people, as directed by the financial institution during controller installation.

Terminal Addressing

The controller application program uses data transmission input/output (I/O) instructions containing logical device addresses to send data to and receive data from the terminals without being concerned with the physical address of the terminal. However, the physical address is used by the controller to select a component within a device.

Physical Device Addresses

Each terminal component attached to a controller has a unique physical address comprising a loop number, a terminal address, and a component address. Devices attached to the Device Cluster Adapter (DCA) are selected using loop "A". A 4700 terminal may contain one or more addressable components. For example, a 4710 Document Printer is a single-component terminal, while a 4704 has a keyboard and a display, and may also include a magnetic stripe encoder.

The physical address of a terminal, terminal component, or loop device is in three parts:

- The number of the loop to which the terminal attaches (X'A' for terminals connected to the DCA).
- The address of the terminal, which is set in the terminal switches (0-15), or the port number (0-7) for terminals connected by the DCA.
- The address of the terminal component, which is fixed for each component type.

---

1 Not available in the United States.

The physical address of a DCA-attached terminal or component comprises the following:

- The direct-attach identifier is X'A'.
- The address of the port (0-7).
- The address of the terminal component, which is fixed for each component type.

You specify the variable parts of the physical addresses for each addressable unit in the system during configuration.

Setting Loop Addresses

Loop terminals have terminal address switches to set the terminal's base address on the loop. All terminals also have component address that is either fixed or can be set by using additional switches. The following terminals have component addresses as shown:

Component:	Component Address:	
4704-1/3604 keyboard	1	
4704-1/3604 display	2	
4704-1/3604 magnetic stripe encoder	3	
3610/3612 document printer	4	
3615 administrative terminal printer	4[2]	
4710/4720/3616 passbook/document printer	4[3]	
3611/3612 passbook printer	5	
3606/3608 keyboard/display and magnetic stripe reader	6	} Financial Services Terminals
3608 printer	7	
3614/3624 consumer transaction facility	8	

You indicate base address 16 by setting 0 in the switches.

The 3612 and 3616 Passbook and Document Printers have two addressable components: the passbook printer and the document printer. A 3608 Printing Financial Services Terminal has two addressable components: the keyboard-display-magnetic stripe reader and the printer. All keyboard/displays have either two components (the keyboard and the display) or three components with the magnetic stripe device. All other devices are single components.

The Financial Services Terminals are designed so that a collection of these terminals (a terminal group) attached to a loop all share a common terminal address on the loop. The common terminal address is set in each terminal through the terminal address switches mentioned above. In addition, the financial services terminals have a second set of switches that set a subaddress. This subaddress identifies which terminal within the terminal group is assigned to the common terminal address. More than one group can be on a loop.

---

[2] If address sharing is used, the component address may be any value ranging 1 through 15 to match the DEFADDR configuration macro.

[3] The component address is any value ranging 2 through 15 to match the DEFADDR configuration macro.

One 4704-1 Display and one 4710, 4720, 3610, 3611, or 3612 printer can share the same loop address, but each 36nn device must have a special address-sharing terminal loop adapter.

Device Cluster Adapter (DCA) Device Addresses

DCA terminals attach directly by coaxial cable and have component addresses similar to those for devices attached to the loops. The following directly-attached terminals have fixed component addresses (binary numbers):

Component:	Address:
4704-2,-3/3278/3279 keyboard	1
4704-2,-3/3278/3279 display	2
4704-2,-3 magnetic stripe encoder	3
3262 printer	4
3287 printer	4

A 4704 with the magnetic stripe encoder has three addressable components: the keyboard, the display, and the encoder. A 3278 or 3279 has two addressable components; the keyboard and the display.

Logical Device Addresses (LDA)

One or more terminal components may be assigned to a work station. When a component is assigned to a station, it is given a logical device address (LDA). The LDA allows your application program to refer to terminal components without concern for their physical addresses.

Each station has eight LDAs, numbered from 0 to 7. Any terminal component can be assigned to any LDA. This assignment can be done during the configuration (CPGEN) procedure, or during program execution using the ASSIGN or DPOOL instructions.

By following a convention of assigning a component to a given LDA (for example, 4704s to LDA 1), an application program can be shared by more than one station, even though the physical device addresses vary from station to station. The ASSIGN, DEVPARM, LCHECK, LREAD, LWRITE, SIGNAL, and WRTI instructions refer to the LDA rather than the physical device address.

For example, assume that each of three stations has a 4704 display, and each display component has been assigned to LDA 1. All three stations share an application program that uses LDA 1 to refer to the display. When a station executing the program refers to LDA 1, the controller converts the LDA reference to the physical address of the 4704 display component assigned to that station, and transmits or reads data accordingly.

The LDA concept allows you to write application programs that can be shared among controllers and stations without regard for physical component addressing, providing that you follow the LDA-to-component addressing convention on all controllers and stations.

There is a set of default LDA assignment conventions and mnemonic LDA codes supported by the configuration macros and the 4700 assembler instructions. These default assignments and their corresponding mnemonics are:

Component:	LDA:	Mnemonic:
3614/3624 consumer transaction facility	0	CT
4704/3604/3278/3279 keyboard	0	KB
4704/3604/3278/3279 display	1	DS
4704/3604 magnetic stripe reader/encoder	2	MS
4710/4720/3610/3612 document printer	3	JP
3615 administrative terminal printer	3	JP
3616 journal printer	3	JP
3611/3612 passbook printer	4	PB
3616 document/passbook printer	4	PB
3606/3608 keyboard/display magnetic stripe reader	0	(none)
3608 printer	1	(none)
3262/3287 printer	6	(none)

Note: When like components of a terminal group — the Financial Services Terminals — have the same LDA, the subaddress identifying the specific terminal component in the group is put in the work station's segment 1 (SMSTGU). When unlike terminal components — the 4704 and 4710, for example — share a common terminal address, they cannot have the same LDA.

The data transmission instructions DEVPARM, LCHECK, LREAD, LWRITE, SIGNAL, and WRTI all have an operand used to specify the LDA of a terminal or terminal component. The operand can be either a decimal number (0-7), the label of an EQUATE instruction that specifies a decimal number, or one of the mnemonics listed above. If the mnemonic is used, the applicable LDA is generated in the instruction. For example, if KB (for the 4704 keyboard) is specified, LDA 0 is generated in the instruction.

Assigning LDAs During Configuration

All terminals must be defined to the controller by DEFADDR/DEVnnnn or DCAPORT/DCAnnnn pairs of configuration macros. For loop devices, the configuration macros specify the loop number, the terminal base address portion of the physical address, and the subaddress of the terminal address for variably addressed terminals. The component addresses are not coded for terminals with fixed addresses. For devices attached via the device cluster adapter (DCA), the DCAPORT configuration macros specify the port number.

You use the STATION, DEFADDR, or DEV/DCAnnnn macros to relate LDAs to actual devices by first relating the LDA to a physical address, and then relating the physical address to a device. You can also assign the device to a device pool for a certain station or set of work stations by specifying a common device pool (DPOOL) name on those macros.

LDAs and the physical device addresses of a DCA terminal are related through the DA operand of the STATION macro. The label of a DCAPORT macro and the LDA to be assigned to the terminal component are specified in the DA operand. For terminals having more than one component (the 3278, for example), the DA operand allows assignment of any logical device address to any component.

Note: Financial Services terminal components of the same type sharing a terminal address are treated as one logical device, for example, all have the same LDA.

Device Pools

You can assign devices to groups, or pools, to make them available only to certain work stations. This is useful if you want to use a specific device type, but are not concerned with which physical device your program uses. Device pools also allow a work station to have exclusive use of a physical device (components of that device may be assigned only to that work station) or to share a physical device with other work stations having the same device pool.

You assign components using the component ID. This ID eliminates the need for knowing the actual physical device address. Individual components of a physical device can be assigned to different work stations, but each component can be assigned to only one work station.

To assign devices to a device pool, you must specify a common device pool name on the DEVnnnn or DCAnnnn macro for the restricted device as well as on the STATION macros for the station or stations allowed to use that device. The program you create for those work stations then acquires and releases devices by issuing the DPOOL instruction. DPOOL, which recognizes components and devices by their configuration-assigned component IDs, assigns devices to and releases them from the work station LDAs.

All devices and components not assigned to either a work station LDA or defined as part of a device pool are in the free pool. Your program can assign and release devices in the free pool using the ASSIGN instruction, described later in this section.

Assigning Terminals from The Free Pool

Controller-attached terminals do not have to be assigned to stations or to a device pool, nor must a station keep one set of terminals. Terminals not assigned to stations or a device pool are in the free pool. Your program issues the ASSIGN instruction to acquire terminals from the free pool, place a terminal in the free pool, or transfer a free pool terminal from one work station to another. The controller rejects an ASSIGN instruction if data is being sent or received from the related device, when status is pending for the device, when the device is shared and ownership has been established, when the assignment is to an LDA already being used, or if the device is in a device pool.

The ASSIGN instruction refers to a parameter list that contains the loop number, terminal address, component address, shared indicator for shared devices, number of the station to receive the terminal (0 for the terminal pool), and LDA to which the component is to be assigned. When the reassignment is completed, the controller replaces the station number and LDA in the parameter list with those assigned to the terminal before it was reassigned. The same parameter list can then be used to return the terminal to its original assignment.

Refer to the appropriate macro descriptions in Volume 6 of the *4700 Controller Programming Library* for information about assigning logical device addresses.

Device Sharing

Any component with the ability to distinguish between operators (Operator A and Operator B keys, for example) can be assigned to two different stations simultaneously. This ability, which must be ordered as a feature on some devices, allows both stations to share the component without having to assign and reassign it. Refer to Chapter 3, "Programming for Printers."

The two stations that share the component are known as the A and B stations. When a shared component is assigned to a station, either during controller configuration or by means of the ASSIGN instruction, the A or B designation for the station must be specified.

During operation, each station regards the shared component as its own. In the case of a shared printer, for example, the operator for station A inserts a form or passbook and then presses the Operator A (or equivalent) key, signaling the appropriate program to continue. Work station B, recognizing only it's own key, remains idle. This prevents the transactions from becoming confused. In this way, the two stations can share the same device and loop resources concurrently.

To define a component as being shared, the DEVnnnn configuration macro describing that component must specify the SHARED=Y operand. Volume 6 contains detailed information about the device descriptor macros (DEVnnnn and DCAnnnn).

Sharing Loop Addresses

Loop devices can, with some restrictions, share the same transmission frame slots. In this case, each sharing terminal has the same loop address, but a different subaddress. Address sharing, also known as slot sharing, allows more terminals to connect to a loop. This group of terminals, called the *slot group*, must meet the following requirements:

1. Address sharing terminals must either have implied address sharing capability, or be defined as address sharing devices during configuration. The DEVnnnn configuration macros for 3600 series devices must specify MG1586 for the OPTIONS keyword if they are other than 3606, 3608, or 3616. These and the non-DCA 4700 series terminals have implied address sharing capability.

2. All address sharing terminals without implied address sharing ability must contain address sharing function RPQ MG1586 or 8K0610. Before attempting to share such a terminal, contact your marketing representative for guidance.

Configuring Address-Sharing Devices

The DEVnnnn macro describes terminals, and the DEFADDR macro assigns terminals to loop addresses, or loop slots. For address-sharing devices, the DEFADDR macro for the shared address must assign multiple terminals to the same address. Multiple component terminals require an even/odd pair of subaddresses.

The STATION configuration macro's TERM= operand either lists the terminals for that station, or points to the DEFADDR macro to associate all terminals assigned to that address with the station. The TERM= operand also assigns up to eight logical device addresses (LDAs) to the terminal components. Any components with unassigned LDAs receive default LDAs beginning with the last LDA, plus one, assigned by TERM=.

In the following example, several terminals for work station 2 share the slot 3 address on loop 1:

```
STA2      STATION   ID=2,SS=1,APBNM=MYPROG,CPU=N,STARTUP=Y,
                    DELSET=X'FF',INSTR=50000,TERM=(SHARE(0,1,6,7,3))
.
.
.
SHARE     DEFADDR   T3604S,(T3608,6),T3610,ADDR=(L1,3,8)
.
.
.
T3604S    DEV3604   OPTIONS=MG1586,...
.
.
.
T3610     DEV3610   OPTIONS=MG1586,...
.
.
.
T3608     DEV3608   ...
```

Programs operating on work station 2 refer to the assigned terminals as follows:

3604 keyboard: LDA 0

3604 display: LDA 1

3610 document print station: LDA 3

3608 keyboard/display: LDA 6

3608 printer: LDA 7

If TERM= points to LDA configuration macros, multiple address-sharing terminals can be assigned to the same or to different work stations. The LDA macro operands can point to a single component. The LDA operands are:

- The label of a DEFADDR macro.
- The label of a DEVnnnn macro referenced by the DEFADDR macro.
- The position of the DEVnnnn component within the component string defined by the DEFADDR macro. If no position is specified, all components are assigned. The same DEVnnnn macro can appear more than once in the DEFADDR macro.
- The LDA for the component. If omitted, LDA assigns the default LDA.
- The shared device (operator A/B) designation.

In the next example, two 3616 printers share an address at loop 1, slot 2. Work station 2 is assigned the 3616 having subaddresses 2 and 3; work station 3 is assigned the 3616 with subaddresses 4 and 5:

```
STA2     STATION     ID=2,TERM=(...,JP2,DP2)
STA3     STATION     ID=3,TERM=(...,JP3,DP3)
.
.
.
JP2      LDA         DFA,T3616,1
DP2      LDA         DFA,T3616,2
.
.
.
JP3      LDA         DFA,T3616,3
DP3      LDA         DFA,T3616,4
.
.
.
DFA      DEFADDR     (T3616,2),(T3616,4),ADDR=(L1,2,4)
.
.
.
T3616    DEV3616     ...
```

Work station 2 and 3 each refer to their assigned components as:

Journal print station: LDA 3

Document print station: LDA 4

Note that both stations use the default LDAs defined by the LDA macro.

Points To Consider When Address Sharing

Terminals sharing one address can be in either a single work station or in different work stations. In addition, programs written for terminals with separate addresses can operate on those terminals when they share a loop address. However, loop performance can be degraded if you do not consider the following configuration and programming points:

Configuring a Shared Terminal for Address Sharing: Sharing a terminal with "operator A/B" capability is not allowed. On terminals where address sharing is a feature, the combination of the shared terminal feature and address sharing functions cannot be ordered on terminals requiring the address sharing function RPQ. Sharing a terminal having implied address sharing and the shared terminal ("operator A/B") ability can cause operator contention when one operator (B, for example) attempts to continue with a printing operation but the printer, which shares addresses with operator A's keyboard display, cannot operate because that keyboard display is active.

Switching Power On and Off: Switching power off and then on again for one terminal in a slot group only causes incorrect device status to occur in SMSDST. To ensure correct restarting, all terminals in the slot group must be switched off for at least 30 seconds before being switched on.

If power is left off for all terminals in the group, the controller steps statistical counter 2 for the first (lowest numbered) component in the group because the slot does not acknowledge Leave Pass Mode commands. The controller issues Leave Pass Mode periodically when no activity occurs on a slot, allowing the terminal to present attention.

I/O Conflicts on Address-Sharing Devices: I/O that occurs simultaneously on separately addressed terminals is done sequentially on address-sharing terminals. As a result, keying at a keyboard while the program writes to a shared printer causes undisplayed, or "blind", keying. Data overrun can also occur if keying is too fast.

A write operation issued to the printer after keying has begun is delayed until either keying stops for three seconds or the controller detects an end-of-message (EOM) character in the input data. However, a keying delay of more than three seconds can allow another printer operation to begin. When keying then resumes, a keyboard overrun can occur.

Controlling Device Operating Characteristics

Each terminal has operating characteristics that must be set by the financial institution. These characteristics include such things as the size of the forms or passbook, and how the passbook should be aligned. Most of the initial terminal characteristics are specified in the DEVnnnn or DCAnnnn configuration macro instructions described in Volume 6. Some characteristics such as printing modes and fonts are effective during some or all of a transaction, and are therefore controlled by the application program using the DEVPARM instruction. Others depend on the data being sent to the device; these controls must be imbedded within the data itself.

For more information on the device characteristics controlled by the DEVnnnn and DCAnnnn configuration macros, refer to the individual device chapters in Part II and the configuration macro descriptions in Volume 6.

Device- and Forms-Dependent Control

Such factors of terminal operation such as forms size, margins, print character and line density depend on the type of form to be printed and the varying limits of the devices themselves. These and other device-dependent parameters are read and set by issuing the DEVPARM instruction in your application program before I/O actually begins.

By issuing a DEVPARM instruction with the SETRET=RETURN operand, your program can read the operating characteristics of most terminals into a parameter list. This parameter list, also defined by DEVPARM, varies depending on the device addressed and the options specified. The ability to read device parameters is particularly useful if, for example, you need to know the cursor position of a keyboard/display before you write to the display. A DEVPARM specifying SETRET=RETURN and EXP=Y returns an expanded parameter list containing cursor position as well as the line and character density of the display.

DEVPARM also allows you to control device characteristics that cannot be specified or predicted during configuration, such as the centerfold spacing for varying forms or passbooks, or the form size itself.

Data-Dependent Terminal Control

You control the data-dependent aspects of terminal operation such as character and line spacing, carriage returns, and finding a new print position. You do this in your program by inserting hexadecimal control values in the output data stream. These terminal control characters are hexadecimal values ranging X'00' through X'3F'; however, not all of those values are valid control characters, and all control characters are not valid on every terminal or device.

Where possible, control characters are device-independent, or cause device-dependent but similar operations. For example, a new-page control character causes a 4704-1 to erase the screen and then position the cursor at line 1, column 1; the same control character causes a 4710 printer to release a cut form or skip to a new page when using a journal roll and causes the 3611 or 3612 to release the passbook.

Refer to the device chapter in Part II that describes a given device for descriptions of the control characters the device accepts.

Terminal -- Controller Data Flow

The controller contains an input buffer that is approximately 80 bytes long for each attached keyboard display, and a shared input buffer for each terminal group of financial services terminals. Data passes from the terminal to the buffer and then to the logical work station when the station becomes active and issues an LREAD instruction.

The system translates the data when it moves the data from the input buffer to the logical work station's storage. It reads data directly from a 3624 into the logical work station's storage. This data transfer does not take place, however, until the applicable station issues the read.

The controller contains one buffer for each output terminal or terminal group, except the 3614, 3616, 3624, and 4710. Data goes directly to these devices from the station's storage. The buffer size for the other devices is approximately equal to the line length for the particular device.

Processing Asynchronous Input Data

Data received at the controller from a terminal or terminal group causes the controller to set a bit in the LDA's attention summary field (SMSSAM) of the receiving work station. The bit that is set (0-7) corresponds to the logical device address of the terminal transmitting the data. The bit is set whether the station is active or idle, and whether the LDA is able to cause asynchronous interruptions or not.

An active station can test the SMSSAM field periodically and read from the LDAs if their bits are set. When dispatched, an idle station can test the field and issue read instructions to the appropriate LDAs.

The LDA's attention summary field determines when terminals assigned to LDAs that cannot cause asynchronous interruptions have data to transmit to the work station. When the controller performs the read instruction, it turns off the attention request bit for that LDA and transfers the data from its own buffer to the station's segment storage.

If the terminal signaling attention is a 3614 or 3624, the controller receives an indication that the terminal is ready to transmit. The controller allows the terminal to transmit after the station has been dispatched and the read has been issued.

Programming for Asynchronous Interruptions

You can specify an asynchronous terminal entry point in the application program. When an idle work station receives data from a terminal with an LDA capable of asynchronous interruptions, the controller places the data in a buffer and dispatches the station at the next opportunity.

You specify the LDAs that can cause asynchronous interruptions in the STATION configuration macro instruction describing the logical work station. When the controller dispatches that work station, the program starts executing at the label defined by the BEGIN instruction's ATD operand.

Translation Tables

The bytes of data that appear on a loop as a result of pressing a keyboard key are referred to as *scan codes*. The bytes of data sent to a display are referred to as *character positions*; the bytes of data sent to a printer are referred to as *print wheel positions* or *character matrix positions*.

Translation tables, either standard tables supplied by IBM or tables specified by the programmer, are used by the controller to translate all data passing between terminals and work stations.

Translation tables are specified during controller configuration. A translation table for each type of keyboard is defined using the INTRTBL and TRTBHDR configuration macros. A translation table for each type of output device is defined using the OUTRTBL or OUTSPEC configuration macros. A translation table for data entered via the magnetic stripe readers or written to the magnetic stripe encoder is defined using the MSTRTBL configuration macro. The translation table for a terminal is selected by the DEVnnnn or DCAnnnn configuration macro describing the terminal. See Volume 6.

Translating Output Data

The controller begins creating an output translation table when an OUTRTBL or OUTSPEC macro occurs in the configuration. The controller then creates a table that contains one position for each possible byte that can be equated to a print position. Each table position is first set to X'FF', and then modified to include the standard character set specified in the macro. The next step is to further modify the table to correspond to the additional positions specified in the macro by your program.

If, during execution, the application program tries to print a translation table position that has not been modified (that is, that still contains hex FF), a translation check occurs. You can avoid translation checks by specifying the DEF parameter. The table generator changes all undefined (X'FF') table positions to the output position specified with the DEF parameter. Specifying DEF ensures that the table can contain no undefined positions and that no translation check can occur. During operation, the translation table converts any data stream characters not defined by the translation table to the character at the DEF position.

The 4710, 4720, and 3616 passbook/document printers use the standard EBCDIC character set. In this case, no translation occurs; the controller sends the data directly to the device. Besides the standard 3616 and 4710 character set, you can also select either a 13-character optional character set from one of the IBM Data Processing National Use Graphics tables or a 16-character user-defined translate table. This user-defined table, which is also possible on the 4720, is created during controller configuration. The DCA-attached 3262 and 3287 printers and the 3614 and 3624 do not use configuration- generated translation tables; They also use the standard EBCDIC character set. Refer to the *3614 Programmer's Guide* and the *3624 Programmer's Guide* for additional information about 3614/3624 character sets and their specification.

Translating Input Data

Input translation tables are generated in the following manner. As a result of specifying a keyboard type, a 2-byte entry is created for each scan position from 00 to the highest scan code on the keyboard. This table is then modified to include the character set indicated by the CASE operand. The macro then further modifies the table to include your specifications for single-character, multicharacter functions, and EOM/EOF keys. The controller adds one byte for each character in a multicharacter specification. When all tables are generated, the CPGEN assembles the TRTBHDR or TRTLIST macros containing the tables for each shift case. Refer to Chapter 2, "Programming for Displays and Keyboards" for a description of the universal translation table.

Printing without Translation

Your program can stop translation of normally translated output data by including a *transparent write* control character in the data stream. The data specified with transparent write contains the actual print positions to be printed or display positions to be displayed. For example, to display position 94 (a rising arrow) on the 3604 without translation, you send the following control and data character sequence:

```
X'35015E'
```

Terminal Status Codes

Status is returned after execution of LREAD, LWRITE, LCHECK, WRTI, DEVPARM, or SIGNAL instruction to indicate a failure or exceptional condition. Status is checked by coding a branch instruction that checks for a condition code of hex 02. The status is returned in segment 1 in the 2-byte field named SMSDST. Refer to Appendix B for a detailed explanation of the status bits and suggested actions.

Signaling Attention

If a logical work station has a keyboard assigned to LDA 0, and if the station is not idle, the operator can send an attention signal to that station by pressing the reset key twice.

This action sets the attention bit in the station's SMSIND field unless a read operation for the keyboard is pending when the attention signal is sent (in this case, the X'0800' cancel status code is returned for the read operation, and SMSIND is not set). In addition, it may break the station out of a conditional wait state, and may result in the interruption or inhibition of a deferred data transmission instruction issued to a shared device.

Your program can test the attention bit in SMSIND. If the bit is on, the program should reset it before issuing any additional I/O instructions. The attention bit is always reset by an LEXIT instruction.

Conditional Wait

A conditional wait state is indefinite; ending such a state requires some action outside of the controller (for example, operator intervention).

A station enters a conditional wait state if it issues an LWRITE or WRTI after receiving intervention required status (X'8000') from any printer except the 3608. The station will not enter the conditional wait if LCHECK for the device is issued after receiving the intervention required and before issuing the write.

The first write instruction completes with condition code X'02' and intervention required status set in SMSDST. The second write places the station in the conditional wait state. This condition ends and the write executes if the operator makes the device ready. The wait condition also ends and the write operation completes with condition code X'02' and attention status in SMSDST, if the operator signals attention from the keyboard.

A station enters a conditional wait state if it issues an LREAD instruction for the host, or a 3614/3624, when no data is pending.

Deferred Instructions

A logical work station enters a deferred state if it attempts to refer to a shared device that is in use by another station. *Deferred* means that the instruction is repetitively retried until the device is free for use by this station. Each retry is followed by a controller dispatching cycle to give other stations a chance to process. At each retry, the controller tests the attention bit in SMSIND. If the attention bit is on (even if set by the application program), the controller completes the instruction rather than deferring it, sets condition code B'02', and stores status in SMSDST.

Terminal I/O Operations

The LWRITE, WRTI, LREAD, and LCHECK instructions are used to transfer data between the controller application program and the terminals.

The LWRITE and WRTI instructions can be used to send data, control characters, or a combination of data and control characters to a terminal. Hexadecimal values from X'00' to X'3F' are reserved for control characters, X'40' to X'FF' for data; thus data and control characters may be intermixed and transmitted as a unit. WRTI is used primarily for sending control characters since it is limited to transmitting either one or two bytes of information.

The LREAD instruction reads data from the keyboards and magnetic stripe readers.

The LCHECK instruction checks for completion of an I/O operation and puts the logical work station in a wait state if the operation is not complete. When the operation is complete, LCHECK sets the condition code, sets the status (if any) in SMSDST, and resumes program execution at the instruction following LCHECK. LCHECK has a test operand, TIO, that prevents the logical work station from entering a wait state, even though the pending I/O operation is incomplete.

Synchronizing I/O Operations

Synchronization (serialization or overlap) of I/O operations with application program processing is accomplished by use of the WAIT/NOWAIT operands in the I/O instructions, the LCHECK instruction, and by issuing two successive I/O requests to the same device with the NOWAIT operand.

The WAIT Operand

Use of the WAIT operand in the LREAD, LWRITE, and WRTI instructions (WAIT is the default operand for LREAD) causes application program execution to halt while the controller processes the I/O function. The logical work station is again dispatched on completion of the I/O operation and posting of the condition code and status. Program execution resumes at the next sequential instruction. LWRITE and WRTI with WAIT may be used where processing cannot continue until the write operation is completed. The need for an LCHECK instruction to serialize program flow is eliminated.

The NOWAIT Operand

By specifying NOWAIT on the LREAD, LWRITE, and WRTI instructions (NOWAIT is the default for LWRITE and WRTI) you allow your program to continue execution (following the normal dispatch cycle taken after each I/O instruction) while the controller processes the I/O function.

Note: When your program issues multiple I/O NOWAIT commands to different devices, *do not* reuse input or output segment areas until the instructions to which they were assigned have been completed and— in the case of LREAD—until the application program has processed the input.

Issuing two successive I/O instructions with NOWAIT to the same device (without an intervening LCHECK) causes execution of the second instruction to be deferred until the first instruction completes. Only immediate status (command reject, cancel, conditional wait) is posted on the first instruction when issued. Other status (if any) is posted when the second instruction is issued. If the first instruction failed, execution of the second instruction is aborted immediately, and the prior operation bit is set in the status field (SMSDST) along with the other status information.

Note: If you issue LREAD NOWAIT to a 3604, 4704, 3278, or 3279; the data transfer is not complete; and you then issue LEXIT; the controller does not allow the work station to exit until the data transfer ends.

If LWRITE data output is still operating and an LEXIT is executed, the station waits at the LEXIT instruction until the LWRITE completes. If the LWRITE fails and the station does not issue LCHECK, no status is set until the next time the station issues another data I/O instruction to that component. If the component is shared, status returns to the station that issues the next I/O instruction to that component, even if the station did not issue the failing write. Hence, a program should always clear an outstanding write with LCHECK before executing LEXIT.

NOWAIT following WAIT

When an I/O instruction with NOWAIT is followed by one with the WAIT operand, both operations can be considered successful and complete when a condition code of X'01' is returned from the second operation. If a failure occurred, the prior operation bit in the status field will indicate which instruction failed (not on = 2nd failed, on = 1st failed) and, if the first instruction failed, the second instruction is not executed.

NOWAIT and LCHECK with TIO

The combination of the NOWAIT operand in I/O instructions and the LCHECK instruction with the TIO operand may be used to completely avoid any waits for I/O completion when handling terminals. The application program must, however, provide synchronization of I/O with processing where needed.

Programmable Indicators

All loop and DCA devices have program-controlled indicators and lights. Indicators on the 3278/9, 3604, 3610, 3612, 3616, 4704, 4710, and 4720 and attached magnetic stripe and PIN devices are controlled by the SIGNAL instruction; indicators on the 3606, 3608, and 3615 terminals are controlled by embedding program operator control sequences in the data sent to the terminal.

During configuration, one of the 3604 or 4704 programmable indicators can be designated as the magnetic stripe device indicator. This indicator then comes on automatically when a read or write operation to the device is issued. Terminals having programmable indicators also control the PIN keypad feature by issuing SIGNAL to turn on indicator 3. One of the indicators on the control operator's 4704 or 3604 may also be specified during configuration as the unsolicited-message light; this indicator comes on when a message having a second byte of X'F1' is written to the controller log.

Loop Control

Two instructions control the operation of failing loops; STPLPS and STRLPS. STPLPS is used to stop one or all loops that are currently in error recovery. STRLPS is used to start one or all stopped loops.

Chapter 2. Programming for Displays and Keyboards

This chapter provides general programming information for reading data from and writing data to 4700 keyboard displays. Refer to Part II for specific information about each device.

Read Operations

The controller application program reads data from the keyboard by issuing LREAD instructions. The controller reads the incoming data and places it into the specified segment starting at the primary field pointer (PFP). Unless the NOTRACK operand is specified, the controller also displays the incoming data on the display component. When you press function keys (such as backspace, shift, or reset) or EOM keys, the appropriate control actions are performed by the controller. Unless the NOWAIT operand is used, the application program waits until the read operation is completed and status is stored. (TRACK and WAIT are the default values.)

The LREAD instruction is ended by the controller when one of the following events occurs:

1. You press an EOM key and the resulting EOM character is encountered during translation.
2. The end of the input field is reached (if the FLI is nonzero and less than, or equal to, the length between the PFP and the end of the segment).
3. The end of the segment is passed (if the FLI is zero).
4. The input segment provided for the operation is full and ERTLS=Y was specified on the device configuration (DEVnnnn or DCAnnnn) macro instruction.
5. The reset key is pressed twice in succession to signal an attention (if the keyboard component is assigned to LDA 0).
6. A loop error, terminal address card error, or DCA error is detected.

If the end of a segment is passed (if FLI is greater than the length between PFP and the end of segment), a program check occurs immediately and LREAD does not start. After the read operation has been completed, you can test the condition code using a conditional branch instruction, and the message length can be found in SMSIML. If the read operation was not completed normally, status bits are set in SMSDST (if an attention occurred, an attention bit is also set in SMSIND.)

If a scan code results in a multicharacter string during translation and there is insufficient room in the input segment for the entire string, the LREAD ends with incorrect length status, and the input character is saved for the next LREAD. If, on the subsequent LREAD, there is sufficient space for the character string, processing continues normally. If there is insufficient space for the character string again, LREAD ends with incorrect length and the input byte is discarded to prevent looping on the same byte. The controller application program should check the condition code and (if the code is X'02') the status bits before processing the message from the read operation. Status bits and suggested actions are listed in Appendix D, "Terminal and Device Status Codes" on page D-1.

Translation

Each key on a keyboard component (except the 4704-1 ALT key) generates a translatable scan code. The controller translates the scan codes into hexadecimal values representing graphics (alphameric characters or other symbols) according to a header and one or more input translation tables generated for the keyboard component by the TRTBHDR and INTRTBL configuration macro instructions. Refer to the individual device chapters for the specific scan codes and the keys they represent.

Translating Between Scan and Character Codes

You must specify at least one translation table for each type of keyboard component in the system (that is, for each different keyboard feature number in the system). Up to four translation tables, each associated with a case and activated by a shift key, can be defined for each individual keyboard. The same translation table used for a larger keyboard could be used for a smaller keyboard that is a subset of the larger, but adhesive labels would normally have to be placed over engraving on one of the keyboards. The standard translation tables for each device are shown later in the chapters for individual devices.

If you choose, you can redefine the meanings of keys by specifying their scan codes and new equivalent values when the INTRTBL macro creates the translation tables during configuration. However, the scan codes of keys defined as numeric (0 through 9) and as plus (+) and minus (-) keys must translate into the equivalent EBCDIC characters (X'F0' through X'F9', X'4E' and X'60') to allow instructions such as LDFLDC, LDSEGC, and VERIFY to execute properly.

Any key on most keyboards can be defined so that its scan code is translated into any one or string of from two to seven characters. Translation into multicharacter strings can support special keys (such as deposit and double or triple zero) and convert characters (such as Kanji to Katakana).

Keys that cannot be translated into multiple character strings are those that have fixed functions when engaged with the ALT key, and all keys on the 4704-2 or 4704-3 when operating in local tracking mode. Refer to "DATSM Programming Considerations" on page 4-27 for a description of local tracking mode.

Defining Accented Characters

Characters requiring accents (acute, grave, circumflex, cedilla, diaeresis, and tilde) can be entered from any terminal having an accent hold table defined for it. The accent hold table defines the accent character scan codes, their hexadecimal values, and the scan codes and hexadecimal values for the associated letters allowed for those accents. When the operator enters a valid accent character, the controller holds the cursor in that position. The operator must then enter one of the allowed alphameric characters for that accent before the cursor moves to the next position.

Your program can use a default accent hold table, or you can define a table of your own. To use the default table, the INTRTBL macro for the input case containing the desired accent must define the accent key scan code, its hexadecimal value, and the operand, 'DKA'. The "DKA=" (dead key accent) operand selects the default accent hold translation table; each time an operator enters that accent, only the characters allowed by the default table can be entered.

To define your own accent hold table, you must also code a DKATBL configuration macro to define the table content and then select it by name with the DKATBL= operand of the TRTBHDR macro. Refer to Volume 6 of the *4700 Controller Programming Library* for more detailed information on coding these macros and defining the accent hold table.

Note: When you have defined the accent key and the accent hold table, you must change the output translation table to display the composite character (character and accent) in the OUTRTBL configuration macro.

Tracking

If the TRACK operand is specified in an LREAD instruction, the controller performs tracking in the following manner:

1. The scan codes sent from the keyboard are compared with the scan codes in the input translation table.

2. The character or characters equated to each scan code are placed in the input buffer.

3. Each character in the input buffer is compared with the characters in the output translation table. If the character in the input buffer is not defined in the output translation table or is a device control character (X'00' through X'3F'), the controller substitutes a blank for that character position.

4. The position codes found for each character in the output translation table are then sent to the display where they appear as indicated by the cursor. The cursor automatically moves to the first position of the next line when a line fills.

Tracking does not take place until the application program issues an LREAD. Any characters entered before tracking is requested are held and displayed after the LREAD instruction is issued. Tracking begins at the current output position for the display component; the controller application program can select a character position before issuing the LREAD instruction. A new line operation occurs automatically when the last character of a line has been tracked. Any additional characters are then tracked. The application program can also begin a new line by sending the new line control character. Tracking does not take place while the keyboard display is in purge mode.

Purge Mode

Purge mode is the keyboard state during which characters from keys other than the Reset key are discarded. When the keyboard display is in purge mode, the Check indicator is on. Purge mode means that the data entered previously either could not be handled soon enough by the controller or contained an error according to the application program. Pressing the Reset key takes the keyboard display out of purge mode and returns it to its normal state.

Purge mode results from the following conditions:

1. The keyboard buffer was overrun because the controller could not receive the input data fast enough. This may occur if data is written to the display at the same time that data is entered at the keyboard.

2. The input buffer for the keyboard in the controller was overrun because more than 47 characters were entered before the controller application program issued an LREAD instruction to read from the keyboard.

3. An error was detected during a read from the magnetic stripe reader. Pressing the Reset key once takes the keyboard display out of purge mode but does not end the read operation (which can be ended by pressing the Reset key twice to cause an attention).

4. The controller application program detected an error and placed the keyboard display in purge mode by the SIGNAL instruction. Purge mode forces the operator to acknowledge the error by pressing the reset key, thereby clearing any characters entered since the last EOM or EOF key was pressed. The next LREAD instruction issued by the application program reads data entered after the Reset key was pressed.

The application program can issue a keyboard purge request with an outstanding LREAD NOWAIT. This ends the outstanding read request and allows the program to continue without waiting for the operator to complete a message. The program can then write to the display, exit, or perform any other operation.

EOM/EOF Keys

An end-of-message (EOM) key is any key that translates into an EOM character. You must define at least one EOM key for each keyboard. When the controller encounters an EOM character for a keyboard, it ends the physical read operation from the keyboard, and the LREAD instruction, allowing the application program to read the data. An end-of-field (EOF) key is any key that translates into an EOF character. EOF ends the LREAD but does not end the physical read from the keyboard. This saves loop overhead involving starting and stopping physical terminal read operations when reading data as a series of fields. If no additional input occurs within approximately three seconds of receipt of an EOF value, the read is ended, allowing the resumption of faster display output.

Each EOM or EOF key has associated with it a mask that is used to check whether the key currently represents EOM or EOF for the keyboard. If any bit in the mask for the key matches a corresponding bit in another mask called the EOM/EOF set selection mask, the key represents a valid EOM or EOF. Define EOM and EOF keys and their masks with the INTRTBL macro. Specify the EOM/EOF set selection mask by the EOMSET operand of the device configuration macro instruction, and modify it with the DEVPARM instruction.

An EOM or EOF key can also translate into a character or character string, as defined by the INTRTBL macro. The character or character string is placed into the application program's input segment, and the number of characters placed into the segment is stored in SMSECT. The character or character string can be used by the application program to distinguish among EOMs and EOFs.

When you specify the EID option for EOM/EOF in the INTRTBL macro, a one-byte character of the EOM/EOF (only one byte of a multiple character EOM/EOF) is returned to SMSEID, but is not translated to the input segment. The EOM length (SMSECT) is zero. A value of X'00' is not allowed for EID. When SMSEID is not zero, it contains the EID character; when SMSEID is zero, no EID was specified and EOM is returned with its length in SMSECT.

Two EOM keys, defined as EOMA and EOMB by the INTRTBL macro, can also indicate which operator is using a shared station. The first time that the program reads from the keyboard after controller load or an LEXIT command, these keys indicate to the controller which segment zero (application registers) to use for the station. If the program senses neither EOMA or EOMB at message end, the controller uses the segment zero last selected for the station.

When the station translates the EOMA or EOMB key, it places a X'FA' or X'FB' into the application program's input segment to indicate which operator entered the message. The EOMA or EOMB key can translate into an additional character or character string, which is placed into the segment ahead of the X'FA' or X'FB'.

The INTRTBL macro also allows specification of EOF, EOFA, and EOFB keys in place of the EOM designations.

Function Keys

Most of the 4700 keyboard/displays have the function keys described in this section. The transient shift, downshift, and repeat-action keys are not available on keyboards attached to the secondary port of a 4704-2/-3.

Backspace Key

A Backspace key moves the cursor and the location in the application program's segment backward one character at a time, nondestructively, as far as the beginning of the current message. Attempting to backspace beyond the beginning of the current message results in no operation. A Backspace key is defined as BKSP in the INTRTBL macro. Its definition is optional.

Double Backspace

This key moves the cursor, and the location in the program's segment, backwards two spaces at a time, nondestructively, as far as the beginning of the current message. If you try to backspace past the beginning, the request is ignored. Define the Double Backspace as BKSP2 in the INTRTBL macro instruction; it is optional.

Advance Key

An Advance key moves the cursor and the location in the application program's segment forward one character at a time, nondestructively, as far as the end of the application program's segment. Attempting to advance beyond the end of the segment results in a segment overflow error condition. An Advance key is defined as ADVAN in the INTRTBL macro; its definition is optional.

Double Advance

This key advances the cursor and the location in the program's segment two spaces at a time, nondestructively, as far as the end (or one character from the end) of the program's segment. If you try to advance past the end, a segment overflow results. Define a Double Advance key as ADVAN2 in the INTRTBL macro instruction; it is optional.

Shift Keys

You can define up to four input translation tables to convert keyboard data, and the keyboard keys to select those tables. Figure 2-1 shows the types of case shift keys you can define and what they do.

SHIFT KEY TYPE:	FUNCTION:	HOW DEFINED:
Locking	Selected case remains active until another shift key is pressed.	CASn parameters of INTRTBL macro
Transient Shift	Selected case is active only while key is pressed.	CASn parameters of INTRTBL macro and SHFT parameter of TRTBHDR macro.
Downshift	Pressing and releasing the key once returns to previous case. Pressing and releasing the key twice or more makes the first case (CAS1) active.	SHFT parameter of TRTBHDR macro only.

You can define from one to four locking shift and transient shift keys for each translation table (case) and one or more downshift keys for almost any keyboard. However, you cannot define a total of more than 16 downshift and repeat-action (typematic) keys for any keyboard. In addition, you cannot define either transient or downshift keys for a keyboard attached to the secondary port of a 4704-2/3.

There are restrictions for using a downshift key on a 3278 or 3279. Also, you cannot define downshift or transient shift keys for keyboards attached to the secondary port of a 4704. For more information on these restrictions, refer to the appropriate device chapter in Part II of this manual and to the descriptions of the macro instructions in Volume 6 of this programming library.

ALT (Alternative Function) Key

This key, available on the 4704 model series, provides alternative functions for certain other keys on the keyboard such as cursor type and display selection, and unit self-test functions. The 4704-1 sends a translatable scan code of X‘3F’ to the controller, as do the 4704-2 and 4704-3 when operating in controller tracking mode; however, translating the ALT key code on these 4704 models can cause the operator to lose certain display functions and may disconnect the display from the controller. Translating the X‘3F’ scan code is therefore not recommended.

Repeat-Action Key

A repeat-action (typematic) key repeats its function continuously until it is released. Repeat-action keys can be specified by the TRTBHDR macro; their specification is optional. Once defined, a repeat-action key remains in effect regardless of the case being used. You can specify no more than 16 repeat-action and downshift keys, total. You also cannot define a repeat-action key for a keyboard attached to the secondary port of a 4704-2/3.

Repeat-action keys are a function of the 3278 and 3279; they may not be defined by TRTBHDR for those devices. In addition, you cannot define repeat-action keys for a keyboard attached to the secondary port of a 4704.

Reset Key

A Reset key must be defined for each keyboard by the TRTBHDR macro; it is the same for all cases of the keyboard. The Reset key can cause the controller to carry out any of the following actions:

Resetting Error Conditions: When the Check or System Check indicator is on to indicate purge mode, pressing the Reset key once resets the error condition and allows normal input to resume from the point of error.

Cancelling Input: When the Check indicator is not on, pressing the Reset key once resets the current message by discarding all characters entered since the last EOM or EOF character; the characters are changed to blanks (a X'40') in the application program's segment, and any tracking of the characters (described below) is erased from the display component.

Sending Attention: Pressing the Reset key twice for an active station assigned to LDA 0 sends attention to the controller.

The application program detects the attention by testing the attention indicator (SMSIND). The controller then takes appropriate action. It does not end the transaction because unpredictable results can occur; however, attention does cause the controller to cancel some wait conditions if data transfer has not yet started because the length of these conditions is unlimited.

The controller ends a keyboard, 3614/3624, or host READ WAIT operation and indicates the abnormal read operation ending in the condition and status codes. Attention causes the controller to also cancel a write wait (intervention required) condition for a printer or a check wait condition for the magnetic stripe reader/encoder (although the reader/encoder remains ready to encode a stripe unless reset by the application program).

If the controller detects the attention after data transfer begins, the wait operation is not cancelled since it is now a limited (conditional) wait. This removes the need for unnecessary application program processing.

Logging On the System Monitor: Pressing the Reset key three successive times causes the idle station to try logging on the system monitor facility. If the system monitor facility is unavailable, the Check light comes on; otherwise, a logon message appears on the display to indicate that the monitor is operating.

Shifting Case: Depending on the configuration of the terminal, pressing the Reset key can cause the terminal to return to case 1 from another case. If the terminal's configuration macro specifies the *NSCC* option parameter, the terminal remains in the current case when the Reset key is pressed.

Examples of Reading From The Keyboard

Example 1. EOM Keys and EOM/EOF Set Selection Mask

The keys on the 30-key numeric 3604 keyboard (see Figure 2-1) could be defined by specifying the TRTBHDR and INTRTBL macros as follows:

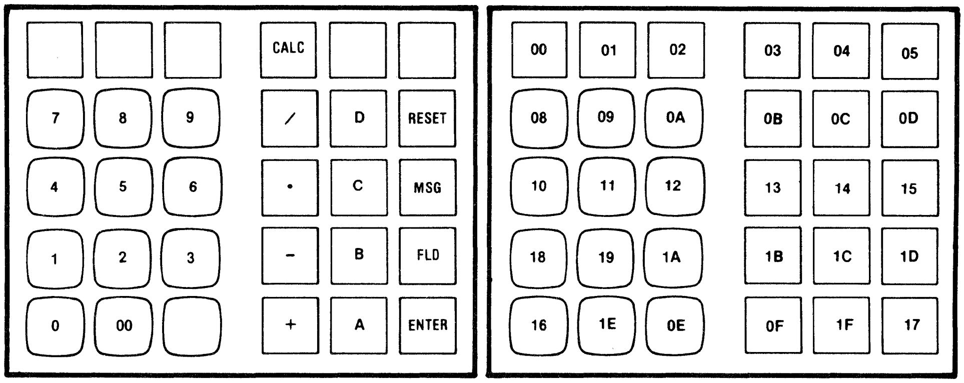


Figure 2-1. 3604 Numeric Keyboard

```
KYBD     TRTBHDR   CANCEL=0D,INTRTBL=CASE
CASE     INTRTBL   (1E,C'00'),(0E,C'.'),(0F,C'+',EOF,X'04'),
                   (1B,C'-',EOF,X'04'),(13,C'*',EOF,X'04'),
                   (0B,C'/',EOF,X'04'),(03,C'&',EOM,X'03'),
                   (1F,C'A',EOF,X'01'),(1C'C'B',EOF,X'01'),
                   (14,C'C',EOF,X'01'),(0C,C'D',EOF,X'01'),
                   (1D,C'F',EOM,X'07'),(15,C'M',EOM,X'07'),
                   KYBD=4661,CASE=1
```

Keys 0 through 9 are defined as character keys, and ENTER is defined as an EOM key in the default translation table for this type of keyboard. RESET, with scan code 0D, is defined as the Reset key. The double zero and decimal point keys, with scan codes 1E and 0E, are defined as character keys. The +, -, *, and / keys, with scan codes 0F, 1B, 13, and 0B, are character keys and, depending on the mask, also EOF keys. CALC, scan code 03, is a character key with the value C'&' and, depending on the mask, also an EOM key. The A, B, C, and D keys, scan codes 1F, 1C, 14, and 0C, are character keys and, depending on the mask, also EOF keys. FLD and MSG, scan codes 1D and 15, are character keys with the values C'F' and C'M' and also EOM keys.

Also, the EOM/EOF set selection mask could be set initially to X'01' by specifying EOMSET=X'01' in the device configuration macro. When the EOM/EOF set selection mask is X'01', the A, B, C, and D keys are EOF keys. When the mask is X'04', the +, -, *, and / keys are EOF keys.

A transaction involving a credit of 50 and a debit of 100 to a balance of 500 in account 1234 could be handled in the following different ways:

- *Transaction processing by field with EOF field delimiting.* could be pressed to indicate processing by field to the controller application program, which could set the EOM/EOF set selection mask to X'01'. The application program expects each field to be followed by an EOM or EOF. The group of fields making up a transaction is followed by an ENTER. The following sequence of keys could be pressed:

  ```
  1 2 3 4 A 5 0 0 B 5 0 C 1 0 0 D ENTER
  ```

  The application program would receive the fields as the following separate messages:

  C'1234A'

  C'500B'

  C'50C'

  C'100D'

  The application program would probably read from the keyboard without tracking, format each field, and write the formatted field to the display component. (ENTER could be detected as a message with zero message length and no EOM characters.)

- *Transaction processing by field with FLI field delimiting (used only with fixed-length fields).* FLD could indicate processing by field and cause the EOM/EOF set selection mask to be set to X'01'. The controller application program knows the field lengths and uses the FLI rather than an EOM or EOF to read each field. The group of fields making up a transaction is followed by an ENTER. The following sequence of keys could be pressed:

  ```
  1 2 3 4 0 5 0 0 + 0 0 5 0 + 0 1 0 0 - ENTER
  ```

  By proper manipulation of the FLI, the application program could receive the fields as the following separate messages:

  C'1234'

  C'0500+'

  C'0050+'

  C'0100-'

  The FLI would be set to 4 for the first read operation, which ends when the fifth key is pressed; the FLI would be 5 for the other read operations, which end when the sixth key since the last read is pressed. The application program would anticipate a wrong-length indication after the read operations and would read the next field from the keyboard. ENTER could be detected as a message with zero message length and no EOM characters.

- *Transaction processing by message with program-defined field delimiting.* MSG could be pressed to indicate processing by message to the controller application program, which could set the EOM/EOF set selection mask to X'02'. The application program expects each group of fields making up a transaction to be followed by an ENTER.

```
1 2 3 4 A 5 0 0 B 5 0 C 1 0 0 D ENTER
```

  The application program would receive the following message containing all the transaction fields:

```
C'1234A500B50C100D'
```

  The application program would probably read from the keyboard with tracking. The application program would use its own delimiters A, B, C, and D to recognize the fields within the message. (Because ENTER is EOM for the whole message, it would not have to be detected by the application program.)

- *Adding machine processing.* CALC could be pressed to indicate adding machine processing to the controller application program, which could set the EOM/EOF set selection mask to X'04'. The following sequence of keys could be pressed:

```
5 0 0 + 5 0 + 1 0 0 -
```

  The application program would receive the fields as the following separate messages:

  C'500+'

  C'50+'

  C'100-'

  As each field is received, the application program could perform the appropriate computation and write it to the display component. FLD or MSG could be pressed to return to transaction processing.

Example 2. Space, Backspace, Advance, and Repeat-Action Keys

Repeat-action, space, backspace, and advance keys can be defined for all keyboards by specifying the TRTBHDR and INTRTBL macros as follows:

```
KYBD    TRTBHDR    CANCEL=31,INTRTBL=CASE,TYPA=(40,0E,04)
CASE    INTRTBL    (04,ADVAN),KYBD=47US62,CASE=1
```

The spacebar, with scan code 40, is defined as X'40' and the backspace key, with scan code 0E, is defined as BKSP in the default translation table for this type of keyboard. The key with scan code 04 is defined to be an advance key. All three keys are defined to be repeat-action keys.

The advance and backspace keys change the position for the next character in the application program's segment and (if tracking is done) on the screen. When EOM is pressed, any characters at or past the current character position are not included in the message read by the controller application program.

If the application program's segment contains the following:

```
X'F9F9F9F9F9F9F9'
```

and keys were pressed in the following sequence:

```
1 3 2 ADVAN A B C
```

the segment would then contain:

```
X'F1F3F2F9C1C2C3'
```

and the screen would display:

```
1329ABC
```

If keys were then pressed in the following sequence:

```
BKSP BKSP BKSP BKSP BKSP BKSP 2 3
```

the segment would contain:

```
X'F1F2F3F9C1C2C3'
```

and the screen would display:

```
1239ABC
```

If an EOM key is then pressed, the application program receives the following message in the segment:

```
X'F1F2F3'
```

Example 3. Shift and Downshift Keys

A translation table with two shift cases could be defined for an alphameric 4704 keyboard by specifying the TRTBHDR and INTRTBL macros as follows:

```
KYBD    TRTBHDR   CANCEL=31,INTRTBL=(CASE1,CASE2),SHFT=(30,3D)
CASE1   INTRTBL   (30,CAS2),(3D,CAS2),(20,CAS2),...,
                  KYBD=47US112,CASE=1
CASE2   INTRTBL   (20,CAS1),KYBD=47US112,CASE=2
```

The keys with scan codes 30 and 3D are transient shift keys in CASE1 and downshift keys in CASE2; they are defined in both the TRTBHDR and INTRTBL macro instructions for CASE1. The key with scan code 20 is defined as a locking shift key for CASE2.

Initially, CASE1 is in effect for the keyboard (that is, scan codes are translated using the translation table associated with CASE1). While key 30 or 3D is pressed, CASE2 is in effect; when the key is released, CASE1 is back in effect. When key 20 is pressed, CASE2 is in effect and remains in effect until key 30 or 3D is pressed and released. In the CASE2 translation table, keys 30 and 3D are defined as shift keys to CASE2; pressing the key causes no operation, but releasing the key causes a downshift. For CASE2, these keys could have been defined as a NOOP or data key with the same effect. The shift and downshift keys in this example allow the keyboard to be shifted in the same way a normal typewriter is shifted; key 20 is the locking shift key and keys 30 and 3D are the transient shift keys.

Write Operations

The controller application program writes data to the display component by issuing LWRITE or WRTI instructions. LWRITE is used for data characters or control characters or both; WRTI is used primarily for control characters, since only one or two bytes can be written. The outgoing data is translated and sent to the display component by the controller, starting at the secondary field pointer (SFP) and continuing up to (but not including) the primary field point (PFP). The controller application program manages the screen (indicates character positions and writes multiple lines) by preceding or embedding control characters that cause the controller to perform the appropriate control actions.

The outgoing data must not be modified until the write operation is complete. An LCHECK instruction or a second LWRITE or WRTI instruction (using another buffer) should be issued to the same display component, so that the controller application program waits until the write operation has been completed and status has been stored. The WAIT operand could also be used, immediately causing a similar wait with no further processing.

The controller application program should check the condition code and (if the code is X'02') the status bits before processing further. (Appendix A lists the status bits with suggested actions.) The condition code (in SMSCCD) can be tested by a conditional branch instruction. If the write operation did not complete normally, status bits in SMSDST indicate the error (if bit 3 of SMSDST is 1, the status bits are for a prior write operation, and the current operation was not initiated).

Write operations to the display component and read operations from the keyboard may affect one another, because of their interaction on the loop. Data written back to the display by an LREAD TRACK issued to the keyboard displays more slowly than display data written by a write operation; data is also displayed at a slower speed if keys are being pressed during a write operation.

Character Translation

The EBCDIC characters that make up the message written by your application program represent data to be displayed, or are control commands that cause appropriate actions to be taken.

EBCDIC characters with a value in the range X'00' to X'3F', are control characters; characters in the range X'40' to X'FF' are data characters. Except during a transparent write operation, the controller translates data characters using the translation table created for the display component by the OUTRTBL configuration macro instruction. If the data character is not defined in the translation table or the control character is invalid for that particular display, a translation error occurs. During a transparent write operation, the controller sends that data to the display without translation.

The standard translation tables for the specific devices are shown in the device chapters. These tables define values for all of the standard EBCDIC characters. You can change table entries by specifying the hexadecimal value of the EBCDIC character and the decimal table position of the character you want the hexadecimal value to represent; this is done on the OUTRTBL macro before the table is generated during configuration. You can define different or additional EBCDIC values for any of the display characters.

For those output characters requiring accents, your program can either use a default accent hold table or define its own. Refer to "Translation" on page 2-2 for a description of the processing and requirements for accented characters.

Screen Management

All screen management for the display component is done by the controller application program using control characters to establish the initial line and column position for a write or read with tracking operation, to update the position for subsequent operations, and to erase the screen.

A write or read using tracking changes only the part of the screen involved in the operation; the rest of the screen remains unchanged. The application program must explicitly clear any parts of the screen that are to be cleared.

Immediately after a read operation, the message on the screen matches the message in the application program's segment; subsequently changing the contents of the segment does not change the contents of the screen. The application program must rewrite any messages that are to reflect changes made to them in the segment. For example, a message may be rewritten before having the operator make corrections to it. The controller follows each write operation with an automatic new line control action, unless the message written ends with a control character. This automatic new line can be suppressed by ending the message with a skip control character (X'0400').

When data is displayed in the last position of a line, the current line position is not advanced until a subsequent data character is displayed or the position is explicitly changed by control characters from the controller application program.

The upper left corner of the screen corresponds to line 1, column 1 of an x-line by y-column matrix. The screen wraps from the bottom line to line 1 and from the last position to position 1. Absolute screen references are based on line 1 (vertical), and column 1 (horizontal); relative screen references are based on the current cursor position. A character, line, or column request changes the character, line, or column relative to the base (the current position).

A character position request greater than the highest advances the position p character positions relative to the base, where p is the modulus of total positions on the screen.

A line position request greater than the maximum number of lines advances the position q lines relative to the base, where q is the modulus of total lines available. The horizontal position remains unchanged.

For the 4704, a column request beyond the end of the screen results in the next position being at the end of the current line.

For the 3604, a column position request greater than the maximum columns on a line advances the position r lines and s columns relative to the base, where r is the number of times the maximum column count can be divided into the request and s is the remainder of the division.

For a write or read with tracking operation, the first character, or the cursor, is displayed at the current output position for the display component.

A parameter of the device configuration macro allows specification of cursor operations: always present, not used, or present only during a read operation.

Display Performance

Data written on the display can be positioned using control bytes. Consider these control bytes when the application program is designed. The control bytes are sent out in series with the data bytes and can affect the response and elapsed time of a transaction.

When control characters do not appear at the end of data in the application program's segment, a "new line" control action will be assumed. Two bytes will be added to the data sent over the loop to the terminal. This control action can be suppressed by adding a skip control character (X'0400') to the segment data.

To determine the actual number of bytes to be sent over the loop, you must know both the current screen position and the new screen position desired. Vertical and horizontal movement is accomplished by separate commands to the display screen. These commands are of the following types:

1. Vertical or horizontal movement--two bytes for each.

2. Erase--two bytes for every 255 bytes or fraction thereof. Erase advances the cursor position, and no explicit position need be sent to the display.

3. Erase and restore cursor--four or six bytes, a combination of 1 and 2 above. Erase advances the cursor, and a "position cursor" restores it to the original position.

The actual number of commands required to accomplish a position change is found by computing the difference between the current position and the new position, and then eliminating those commands not causing a position movement. For example, no horizontal position command is required if the current and new column positions are the same.

There are two modes of transferring data to a display on the loop: echo mode and redundant mode. Echo mode transfers data at the rate assigned to the terminal (at two data bytes per slot). Redundant mode speed is one half the assigned speed (each data byte is duplicated and goes to the terminal in one slot).

Redundant mode is automatically selected if the keyboard is being read at the time the application program issues an LWRITE to the display. This allows input while output is in progress.

Redundant mode is also used to transfer data to the display as the result of an LREAD with tracking.

If field processing (multiple LREADs for a single keyboard message) is used, and LWRITEs to the display are issued between fields, it is possible to overrun the keyboard buffer. An EOM character causes the keyboard to leave the read selected state. An EOF (end-of-field) character causes the data to be passed to the application program's segment but leaves the keyboard read selected. The terminal remains in this state for 3 seconds or until an EOM character is sent from the terminal.

Examples of Display Control

Example 1. Character Translation

The display characters [and] could have the values X'70' and X'71' defined for them (in addition to their standard values X'C0' and X'D0') in the output translation table for the display component by specifying the OUTRTBL macro as follows:

```
DISP   OUTRTBL   (91,'70'),(93,'71'),DEVICE=3604,CHARSET=175
```

The characters [[3604 KD]] could then be displayed beginning at line 1, column 1 by writing the following message to the display component:

```
X'0C 70 C0 F3 F6 F0 F4 40 D2 C4 71 D0'
```

X'0C' is the new page control character. X'70' and X'C0' are both defined as the [display character, and X'71' and X'D0' are both the] character.

The display character $ could have the value X'5B' defined for it by specifying the OUTRTBL macro as follows:

```
DISP   OUTRTBL   (36,X'5B'),DEVICE=3604,CHARSET=175
```

The characters ($ 3604 KD $) could then be displayed beginning at line 1, column 1 by writing the following message:

```
X'0C4D5BF3F6F0F440D2C45B5D'
```

The same characters could be displayed without defining a value for the $ character by making use of the transparent write in writing the following message:

```
X'0C4D350124F3F6F0F440D2C435015E4D'
```

X'35' is the transparent write control character. X'01' indicates that the length of the transparent write is 1 byte. X'24' is the hexadecimal representation of the position number of the $ display character.

Example 2. Screen Management

The character K could be displayed at line 4, column 10 and the character D could be displayed at line 5, column 10 by writing the following message to the display component:

```
X'0C34040434000AD2340C013408EFC4'
```

X'0C' is the new page control character; X'34' is the position control character. The characters X'0404' are the flag and positioning bytes for absolute vertical positioning to line 4; X'000A', the bytes for absolute horizontal positioning to column 10; X'0C01', the flag and positioning bytes for relative vertical positioning from line 4 to line 5; X'08EF', the bytes for relative horizontal positioning from line 5, column 11 to line 5, column 10.

The characters [3604 KD] could be written on one line by making use of new line suppression in writing the following two messages:

```
X'0CC0F3F6F0F4400400'
X'D2C4D0'
```

X'0400' is the skip control character sequence, which suppresses the new line control action that would otherwise be performed by the controller.

Magnetic Stripe Reader and Reader/Encoder

The magnetic stripe reader and reader/encoder are special features of the keyboard display. The magnetic stripe reader is read only; however, data is both read and written to the reader/encoder. The application program and the controller operate the magnetic stripe reader as an extension of the keyboard; the magnetic stripe encoder is handled separately. See the specific keyboard display chapters in Part II for detailed information about attaching magnetic devices.

The magnetic stripe reader/encoder has two operating states: read state and encode state. Read state, when the reader is ready to read a magnetic stripe, is the normal state. Encode state exists after the application program has issued a successful LWRITE instruction and the encoder is ready to encode a magnetic stripe. While the reader/encoder is in encode state, an indicator is on to notify the operator. (When configuring a 3604, specify this indicator in the DEV3604 device configuration macro.) The MSTRTBL configuration macro defines the input and output translation tables for the magnetic stripe reader/encoder.

When a document with a magnetic stripe is passed through the reader/encoder, the reader reads the stripe data if the device is in read state, and the encoder writes encoded data on the stripe in encode state. The reader/encoder returns from the encode state to read state after a document has been successfully written. After an unsuccessful encoding operation, the reader/encoder remains ready to encode until one of the following occurs:

An encode operation is successful.

The user presses the ALT and Reset keys on the associated keyboard.

The application program writes a message that has a reset control character (X'0C') as the first byte (any other message characters are ignored).

You can find more information about the 4704 magnetic devices in Chapter 1, "General Terminal and Device Programming."

Description of The Magnetic Stripe

The IBM standard magnetic stripe used with the magnetic stripe reader/encoder is 12.7 mm (0.5 in.) wide. The IBM Model 100 magnetic stripe reader reads magnetic stripe data encoded at a density of 75 bits-per-inch (75-bpi) only; this is the American Banking Association (ABA) standard. The IBM Model 600 reader and Model 200 reader/encoder read data encoded at either 75 or 210 bpi. The IBM Model 200 reader/encoder encodes data at 210 bits per inch. You can use either the Model 600 reader or Model 200 reader/encoder to read data encoded at 210 bpi. You can use any of the 4700-attached magnetic stripe devices to read data encoded by another 75-bpi encoder on the ABA track of an American National Standard stripe.

The Model 200 encoder writes the same data twice on the magnetic stripe. If the encoder fails to write both times successfully, status is returned to the controller application program and the operation ends. If, during read operation, the reader reads the data correctly at least once, the operation is considered successful. Figure 2-2 shows the magnetic stripe used with the reader/encoder and the logical placement of data on the stripe.

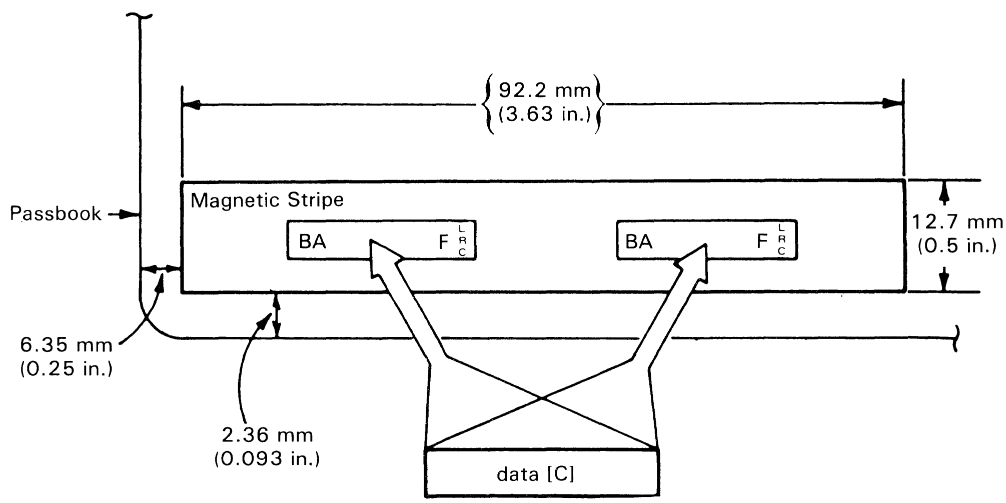


Figure 2-2. The Reader/Encoder Magnetic Stripe

Magnetic stripes on adhesive backing are available from IBM. You can use either these stripes (IBM PN 428650) or their equivalent. In either case, attach the stripes according to Figure 2-2, or the instructions provided with the stripes.

A "Mr" size (CR 80) plastic card with an American National Standard stripe can be used by tellers or branch supervisors to identify themselves to the controller application program. If these cards are encoded by the magnetic stripe encoder, they are limited to 36 bytes of data including the X'BA' that precedes the optional data and the X'F' or X'C' and LRC characters that follow the optional data.

Reading Magnetic Stripe Data

The controller application program reads data from the magnetic stripe reader/encoder by issuing LREAD instructions to the associated keyboard component. Data from the magnetic stripe and from the keyboard can be in the same message, or data from the magnetic stripe can be in a separate message.

The data on a magnetic stripe is in the 4-bit American Banking Association (ABA) standard code or the 4-bit IBM Specifications code, and has the following format:

B A 1-36 characters (0-9, D, or E) F or C LRC

Stripes always start with B which is the start-of-stripe (SOS) character. An A is always included as the second character of magnetic stripes encoded in the IBM code, but is not included in the ABA code. Either C or F may be used as the end-of-stripe (EOS) character. The longitudinal redundancy check (LRC) tests the integrity of the stripe data. The controller translates each 4-bit code, between EOS and SOS inclusively, and places the translated values in the application programs segment. Default magnetic stripe data input translation tables are generated, and optionally modified, by the MSTRTBL configuration macro.

In the default MSTRTBL translation table, (Figure 2-3) B, A, and F have no assigned EBCDIC value, so no corresponding value is placed in the application program's segment. Note the difference between the two EOS characters. C by default is an end-of-message (EOM) character that, when encountered during an LREAD operation, ends the operation the same way that pressing the EOM key does when reading from the keyboard. F by default is not an EOM; therefore, additional data can be entered from the keyboard or magnetic stripe before LREAD ends. C also has a default output value (X'7C') which appears in the application program's segment. This value can be used by the application program to indicate that the data came from the reader/encoder instead of the keyboard.

EID values may also be defined. They act as EOM indicators, but also return the character value in SMSEID. The EOM length (SMSECT) will be 0. Use the MSTRTBL macro to define EID.

Character:	EBCDIC:	ABA Code:	Meaning:
0	X'F0'	0	Numeric
1	X'F1'	1	Numeric
2	X'F2'	2	Numeric
3	X'F3'	3	Numeric
4	X'F4'	4	Numeric
5	X'F5'	5	Numeric
6	X'F6'	6	Numeric
7	X'F7'	7	Numeric
8	X'F8'	8	Numeric
9	X'F9'	9	Numeric
		A	First four bits after SOS
		B	Start of stripe (SOS)
	X'7C'	C	End of stripe and end of message
	X'7D'	D	Field separator
=	X'7E'	E	(Undefined)
		F	End of stripe (EOS)

Figure 2-3. Corresponding EBCDIC and ABA Codes for the Reader/Encoder Default Translation Tables

Encoding Operations

Encoding a magnetic stripe involves two steps. A message is written to the buffer of the magnetic stripe reader/encoder, placing the reader/encoder in encode state and lighting an indicator. Then the magnetic stripe on a document is encoded by passing the stripe through the reader/encoder, returning the reader/encoder to read state.

The controller application program writes data to the magnetic stripe reader/encoder by issuing LWRITE instructions to the reader/encoder. One message can contain from one to 36 EBCDIC characters. The controller translates the EBCDIC characters into the four-bit ABA codes and places the ABA codes BA in front and the 4-bit LRC in back (that is, following the EOS code). The longest translated message is therefore 20 bytes long. Use the MSTRTBL macro to define and modify the translation table.

If a reset control character (X'0C') is the first EBCDIC character encountered while translating the message, the controller ignores the rest of the message and simply sends a command that causes the reader/encoder to be returned to read state and the indicator to be turned off. While translating, the first EBCDIC character encountered that translates into an end of stripe code (C or F) causes the controller to put the code (C or F) into the translated message and stop translation. If no EBCDIC character is encountered that translates into an EOS code, the controller puts F into the translated message for the EOS code.

After an LWRITE instruction has been issued, the controller application program should issue an LCHECK instruction to the reader/encoder to wait for completion of the encoding operation. (The application program should not issue an LEXIT instruction before the encoding operation is completed because the station waits indefinitely until the reader/encoder is reset to read state by having a document passed through it.) After completion, the application program should check the condition code by a conditional branch instruction and, if an error occurred (indicated by a code of X'02'), check the status bits. If the operator pressed the Reset key twice to signal an attention (indicated by an attention bit in SMSIND), the application program should reset the reader/encoder to read state (otherwise, the reader/encoder remains in encode state, and a magnetic stripe might be encoded inadvertently or incorrectly).

Note: To avoid conflicts in the device, the reset (X'0C') should be issued only when the operator has indicated that the encode operation should be ended.

PIN Keypad

The PIN (personal identification number) keypad is a small keyboard that the customer uses to enter a PIN for identification during a transaction.

The application program enables the PIN keypad by switching on programmable indicator 3 using the SIGNAL instruction. The terminal places a X'7F' in the data stream, indicating an unencrypted keypad (a X'7E' is inserted for an encrypted keypad). As the customer enters data, each scan code is sent to the controller. If the customer makes an error, the field can be restarted using the Erase key on the PIN keypad. When this occurs, the terminal inserts a X'0B' in the data stream to cause the controller to ignore the preceding data. When the customer finishes entering the PIN, the terminal adds a X'7F' signifying the end of the PIN data. The PIN keypad is set off, allowing no further entry, when:

- The customer presses the End key.
- The operator presses the Reset key once, ending the PIN prematurely.
- The application program turns the PIN keypad off with SIGNAL.

An encrypted PIN keypad sends no data to the controller until the customer presses the End key, sending all PIN data from X'7E' to X'7F'. If the customer presses Reset, only the X'7E' and X'7F' are sent; the PIN keypad read operation completes successfully, but without recording PIN data.

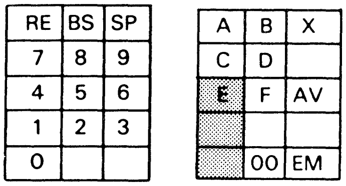
Note that while the PIN keypad is enabled on a 4704, all other keyboard keys except Reset are disabled. Figure 2-4 shows the default scan codes for all PIN pads.

Key:	EBCDIC:	Meaning:
0	X'F0'	Numeric
1	X'F1'	Numeric
2	X'F2'	Numeric
3	X'F3'	Numeric
4	X'F4'	Numeric
5	X'F5'	Numeric
6	X'F6'	Numeric
7	X'F7'	Numeric
8	X'F8'	Numeric
9	X'F9'	Numeric
	X'7E'	Beginning of encrypted PIN
	X'7F'	Beginning of clear PIN
	X'7F'	End of PIN
	X'5C'	Fill character for tracking

Figure 2-4. Default PIN Pad Translation Table

Universal Translation Table

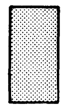
When you order a keyboard display, you can choose among many keyboards and you can assign meanings to the keys through the translation tables. However, if you do not select a translation table, or when you are using the system monitor, a standard *universal translation table* is used. The following pages illustrate the various keyboards for the 4704, 3604, 3278, and 3279 keyboard displays; they show the way the universal translation table handles the various keys. Note that keys A through F are included so you can enter hexadecimal data.



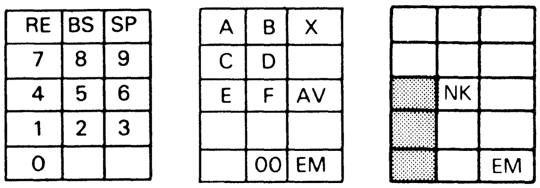
30-Key Keyboard

Legend:

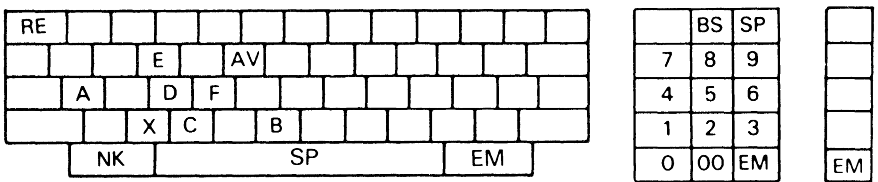
BC	=	blink cursor**	SP	=	space
BS	=	backspace	AV	=	advance/clear key*
FR	=	free key*	AC	=	alternate cursor**
RE	=	reset	CK	=	clicker**
EM	=	end of message			
NK	=	return to normal keyboard = 045			

 = motor bar available

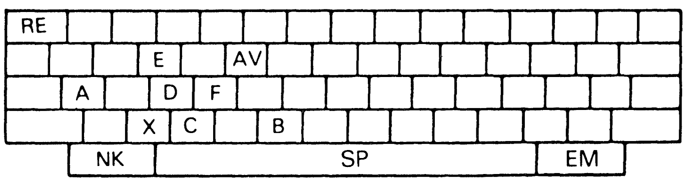
*available only with disk file facilities diskette
**available only with DCA 3278 devices



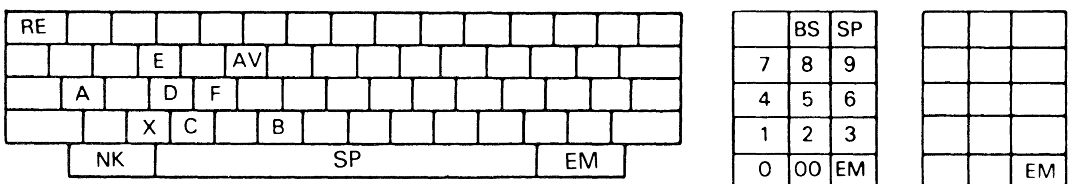
45-Key Keyboard



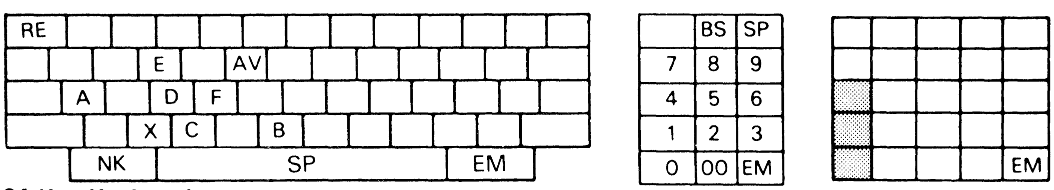
74-Key Keyboard



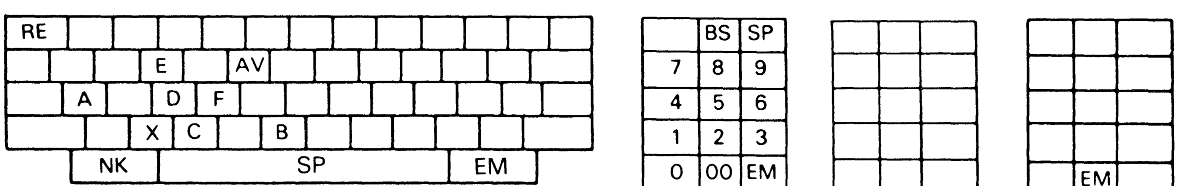
77-Key Keyboard



92-Key Keyboard



94-Key Keyboard



99-Key Keyboard

Figure 2-5. 3604 Keyboards with Universal Translation Table

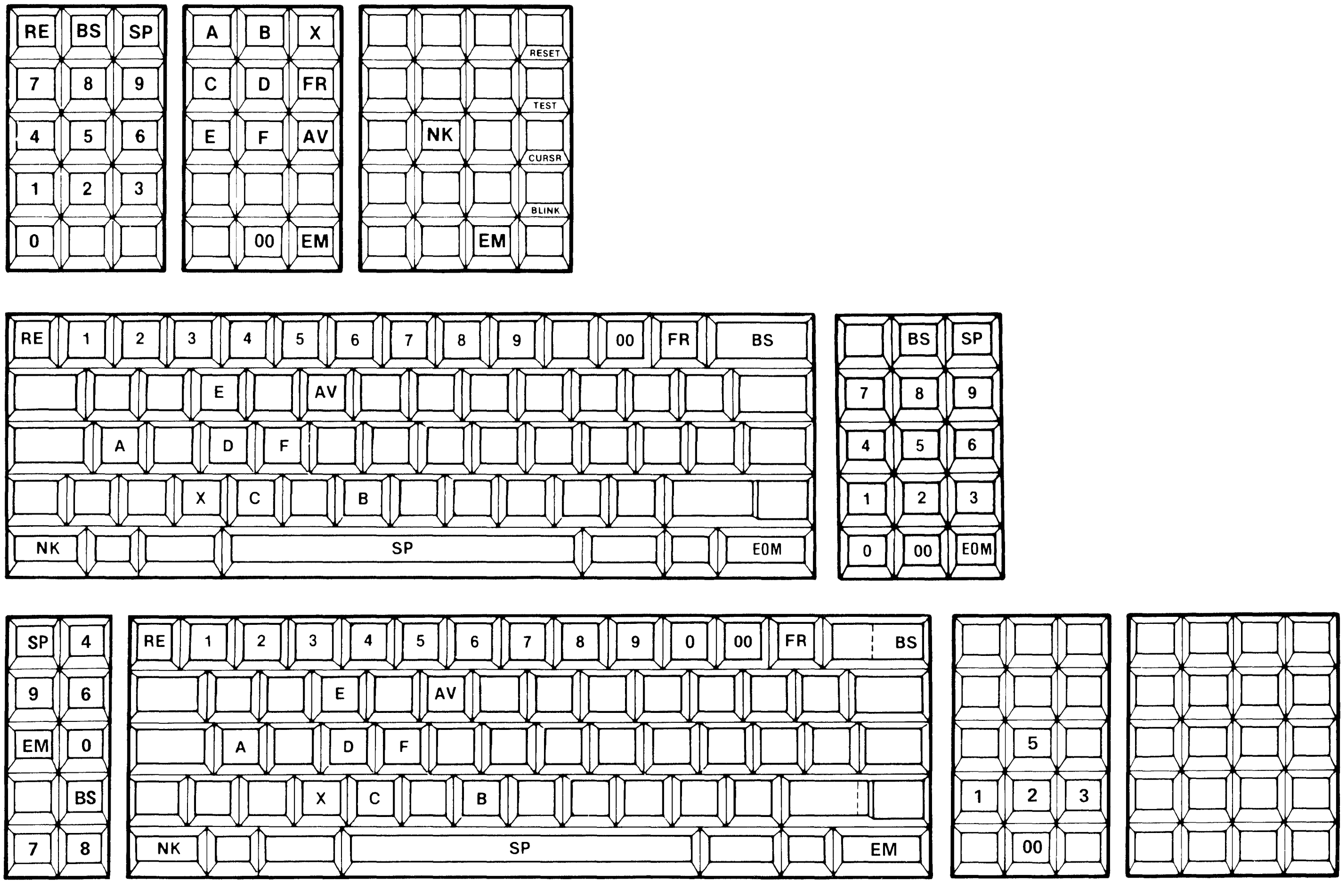


Figure 2-6. 4704 Keyboards with Universal Translation Table

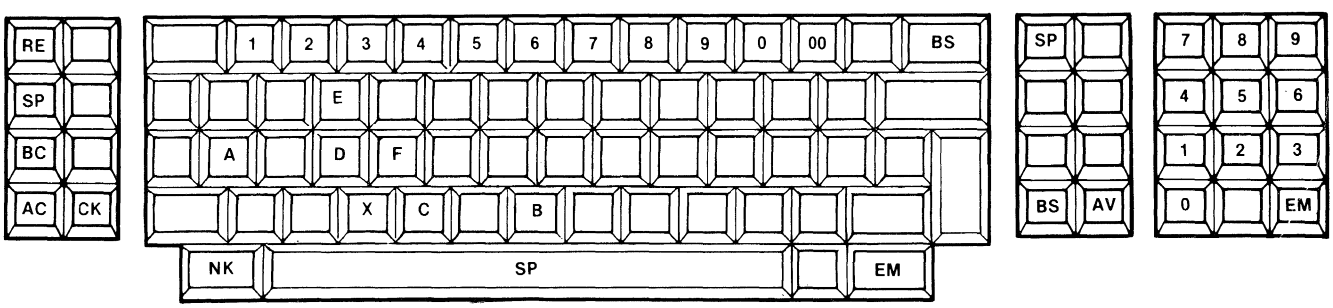


Figure 2-7. 3278/3279 Keyboards with Universal Translation Table

Chapter 3. Programming for Printers

This chapter describes the general programming information you need to control and print data on the printers attached to the 4700 controller. For information about a particular printer model, refer to the device chapter in Part II.

Translating Data and Handling Control Characters

The application program sends data and control characters to the printer by issuing LWRITE or WRTI instructions. The output stream specified by these instructions can consist of data only, control characters only, or data and control characters in any combination.

In response to a write instruction, each byte of the output stream is inspected to determine if it is data or a control character. A data byte is any byte having a value of X'40' or greater. A control character is any byte having a value less than X'40'. Within these two categories, not all hexadecimal values are valid.

Data bytes in the output stream are used to find the character representation in the printer. Control characters are processed according to the rules defined for each printer.

A data check occurs if an invalid character, data or control, is detected. Translation is suspended. Any valid data preceding the untranslatable character is printed, with no carriage return or indexing. Status bits indicating a data check are set for the application program. By using the residual count in SMSIML, the application program can find the invalid character in the output segment.

Printing Operations

When an LWRITE or WRTI instruction is issued to a printer in the ready state, the instruction is completed by setting the condition code (SMSCCD) to X'01' and setting SMSDST to 0's. If the motors in the printer are already running, or the printer does not require start motors, data transmission begins immediately. If the motors are not running, the motors are started before data transmission begins.

When an LWRITE or WRTI instruction is issued to a printer in a not-ready state, or to a printer that appears not ready to the station for which the write instruction is issued, the instruction is completed by setting the condition code to X'02' and setting intervention-required status in SMSDST. Refer to Appendix D, "Terminal and Device Status Codes."

If the instruction is completed with a condition code of X'01', the outgoing data must not be modified until the operation is known to be completed. Issue an LCHECK to determine if the operation is complete, or issue another LWRITE instruction using a different output area.

Sharing of Printers

A printer, or a component of a dual-component printer, can be assigned simultaneously to two different logical work stations so that both stations can share the component without the need for assigning and reassigning it. Either component of a 3612 or 3616 can be shared between two stations. The 3610, 3615, 4710, and 3616 journal print station can be shared in the same way as the document portion of the 3612. The 4720, 3611, and 3616 document print station can be shared in the same way as the passbook portion of the 3612. The two stations that share the component are known as the A station and the B station. A terminal can be shared in this manner if the SHARED=Y operand is coded on the configuration macro that defines the terminal.

In operation, the first station to refer to a shared component by means of a DEVPARM, SIGNAL, LWRITE, WRTI, or LCHECK instruction acquires temporary ownership of that component. When ownership is established in this manner, if the other station attempts to refer to the same component, the instruction is deferred until the owning station relinquishes ownership; that is, the instruction issued to the component by the second station is repetitively retried, with each retry followed by an implied PAUSE instruction to give other stations a chance to process. If the attention bit is set in SMSIND when the instruction is tried, retrying of the instruction is inhibited or interrupted, and the instruction is completed by setting a condition code of X'02' and setting attention status in SMSDST.

Note: If a passbook component or a document component in cut-forms mode is to be shared, the terminal must be ordered with the shared terminal feature. This feature consists of separate START PRINT A and START PRINT B keys. Pressing one of these keys makes the component ready for its corresponding station, but makes it appear not ready to the other station.

The 3262 and 3287 cannot be shared between stations.

Types of Sharing

The shared operation can be either concurrent or nonconcurrent. The type of sharing desired for a device component can be specified during the controller configuration by the DEVxxxx configuration macro. It can also be changed during execution of the controller application program by the DEVPARM instruction. Only printers in continuous-forms mode can be shared concurrently. The effect of concurrent sharing is that output from the two stations can be interleaved on the printer. The messages are printed in the sequence in which the write operations from both stations occur.

The difference between concurrent and nonconcurrent sharing is the way in which ownership is relinquished. With concurrent sharing, ownership is implicitly relinquished at the successful completion of the operation initiated by the owning station. (Successful completion means that the operation was completed with zero status.) If the operation results in any error status, ownership is retained until the owning station receives that status through another instruction referring to the component, or until the owning station explicitly relinquishes ownership by issuing an ASSIGN or an LEXIT instruction. If an owning station explicitly relinquishes ownership without receiving its error status, that status is returned to the next station that refers to the failing component.

With nonconcurrent sharing, ownership must be explicitly relinquished by the owning station by the ASSIGN or LEXIT instruction. Operating with nonconcurrent sharing enables a station to have the exclusive use of the component for the duration of a transaction.

Programming Considerations

When a printer is shared between two stations, a "race" condition can develop if both stations require the use of the printer at the same time. There are two ways to resolve this condition. The application program can resolve this race, or it can enable the operator to determine the "winner" by using the START PRINT keys.

The application program can acquire ownership and indicate the owning station by using the SIGNAL instruction to turn on one programmable indicator light for the *A* station and the other light for the *B* station. By issuing an ASSIGN instruction with the device address specified as X'FFFF', the application program can determine whether the *A* or *B* side of the printer is assigned to the logical work station.

The first station to issue an I/O instruction such as SIGNAL acquires ownership of the printer, and the light notifies the operator to insert the correct form. Following the SIGNAL instruction, the application program issues its first write instruction. If intervention-required status results, the write instruction can be issued again to put the station into a wait state until the operator inserts the document and presses the appropriate START PRINT key. Before issuing an LEXIT instruction, the station should issue another SIGNAL instruction to turn the light off. The second station, attempting to turn its light on, defers until the first station finishes with and releases the printer.

To enable the operator to determine the owning station with the START PRINT keys, the application program issues its first write instruction to the printer. If intervention-required status results, the program should issue LCHECK to inhibit the sharing station from entering a wait state on its next write instruction. The program then issues an ASSIGN instruction to release ownership, followed by a PAUSE instruction to give the other station a chance to process. You should then reattempt the LWRITE instruction.

This programmed loop should include instructions to zero the instruction threshold count (SMSLTC) to avoid a program check. It can also include a switch to display a message to the operator notifying him the first time the intervention-required condition occurs, and can check SMSIND for an attention in case the operator at a keyboard display pressed Reset to cancel the operation. When the operator presses one of the START PRINT keys, the program for that station succeeds in its write instruction attempt and does not reenter the loop. The other station continues to loop until the operator presses its START PRINT key.

Address Sharing

Some terminals and components can use the same slots in the loop transmission frame; for example, a 4710 printer can use the same loop slots as the keyboard display for the same or another work station. For a description of address, or "slot", sharing refer to "Sharing Loop Addresses" on page 1-6.

Printing Chained Data from The Host

Data chaining is a technique for grouping related host link messages together in a series. Chaining allows the receiving station to stay attached to the sending program until all related messages are received.

You can write an application program that passes chained data through to a DCA–attached printer with minimum interruption, allowing the printer to run at the fastest possible rate. To print a data chain, your program must issue DEVPARM to indicate the first–in–chain message, and follow it with an LWRITE to write that message to the printer. Following the first–in–chain message, issue another DEVPARM to indicate the first middle–in–chain message, then issue an LWRITE for each of the following middle–in–chain messages. Issue one final DEVPARM indicating the last–in–chain message before issuing LWRITE to write that message to the printer.

If you choose not to perform data chaining to the printer, you can first issue a DEVPARM indicating only–in–chain. All following LWRITEs are regarded as one–block chains.

Printing Structured DCA Fields

You can print structured DCA level 2 (text) data streams on a DCA printer if you first specify the correct SNA FM header type with DEVPARM. The printer supports only one FM header, and rejects all other types. Refer to the printer's Component Description manual for a definition of the allowable FM header type.

Chapter 4. Processing 3270 Data Streams

The 4700/3270 Data Stream Mapping (DATSM) facility is an option for the controller of a 4700 Finance Communication System. This option allows the use of existing Systems Network Architecture (SNA) Secondary Logical Unit Type 2 (SLU-2) 3270 host application programs—without change—to interact with terminals attached to a 4700 system.

To do this, DATSM provides a set of high-level instructions that translate a stream of output data for a 3270 to an output data stream that devices attached as part of a 4700 system can process. DATSM can also translate input data from a 4700 terminal to input data from a 3270 terminal.

DATSM therefore permits a 4700 application program to cause a 4700 display to appear to VTAM or TCAM as, for example, a 3277 or 3278 terminal. Specifically, the application program uses the SNA and synchronous data link control (SDLC) link interfaces for SLU-2 communication with the host system. It is this type of communication that lets the 4700 display appear as an SLU-2 device (such as a 3277 or 3278). Appendix F, "DATSM Sample Program and Error Codes" shows a sample data stream application program. Refer to this appendix when the text of this chapter refers to the sample program.

Basic Data Stream Mapping Concepts

The process of data stream mapping consists of the following steps for converting data from one format to another:

1. Initializing the DATSM control areas (PUTDMS).
2. Obtaining the display-bound data stream.
3. Passing the data stream to DATSM for conversion (PUTFLD).
4. Retrieving the converted data stream from DATSM (GETFLD).
5. Checking status information to determine the next action and normally, sending the converted data stream on to the display.
6. Retrieving the current field and placing it into the buffer into which the keyboard data will be read (GETFLD).
7. Reading the keyboard associated with the device and returning the previously retrieved field with the keyboard changes to DATSM (step 6 or step 8 next) (PUTFLD).
8. Retrieving the converted data stream and sending it to a host program or processing it in the 4700 controller (GETFLD).

If the 4700 system configuration includes data stream mapping, an image buffer is generated during startup for each station using the facility. The size of the image buffer is defined in the system configuration macros and can be any size, although for a 3270 data stream application, the buffer typically consists of 1920 positions arranged in 24 rows by 80 columns. The image buffers are allocated from the upper end of user storage. If the configuration includes extended user storage, the first image buffers are allocated in the extended storage. Additionally, a table called the field control table (FCT) is generated in user storage (not accessible by the controller application program).

After initialization, the application program reads display-bound data from the communication line (alternatively, this data can be created by the 4700 application program), and passes the data to the data stream mapping facility (DATSM) using a 4700 application instruction (PUTFLD). The instruction causes the data to be placed in the DATSM image buffer in the same arrangement as it would have taken on the originally intended display screen, with attribute positions identified by a start field code, X'1D'. The instruction also causes the attributes related to the data to be recorded in the FCT.

After the application program has presented all display-bound data to DATSM, it can retrieve the data using another instruction (GETFLD). Execution of this instruction causes the SNA character string (SCS) data stream for the 4700-attached display to be placed in a user segment. The application program should then use the appropriate I/O instructions to write this data to the display.

The user's I/O buffer can be shorter than the SCS data stream returned by the instruction. In this case, the user should write the data in the buffer to the display and reissue the instruction. This continues until all data is written to the display.

For maximum response time, the user's I/O buffer should be large enough to contain the longest data stream that will be processed. Hence, the user must weigh storage requirements against requirements of display interaction.

In the same way that display-bound data is placed in the image buffer, processor-bound data (that is, data read from the 4700 display) is placed in the image buffer. Subsequently, an application program can remove selected data from the image buffer for processing in a controller application program or for transmission to a host computer.

An additional function, "windows", allows a display-bound data stream intended for a specific display screen size to be mapped to the 4700-attached display screen of a different size.

The DATSM Instructions

The DATSM instructions are:

PUTDMS initializes the field control table.

GETDMS retrieves information from the field control table.

PUTFLD passes data streams to DATSM for mapping into the image buffer.

GETFLD reads data streams from the DATSM image buffer.

These instructions perform all communication between the application program and DATSM. A detailed description of each instruction is in Chapter 5, "4700 Loop and DCA Assembler Instructions."

You use the PUTDMS instruction in the 4700 application program to move mapping information from the DMS into the FCT header. You use the GETDMS instruction to reconstruct the DMS from the appropriate FCT fields at a location you define.

You use the PUTFLD and GETFLD instructions to move the data streams to and from DATSM, as described in the rest of this chapter.

DATSM Terminology

Other terms that are essential to an understanding of DATSM follow:

Buffered operation

Transferring data between the controller and the display by allowing the image buffer, when full, to steal a write cycle from normal controller I/O operation. Unbuffered operation is the same as normal 3270 operation.

DMS Header

The first 16 bytes of the DMS. The content can change with each use of a DATSM instruction.

Keytracking

The 4704-2/-3, when operating in local tracking mode, processes input data within the terminal. This is called *local tracking*. When operating in unbuffered or 3270 compatibility mode, input data is rewritten to the display through the controller. This is called *controller tracking*.

Window

A rectangular area in the image buffer defined to be the size of the data area to be displayed. The window must be smaller than or equal to the image buffer; otherwise, status is returned. The dimensions of the window must also be smaller than or equal to the corresponding dimensions of the actual display; otherwise, the displayed image appears distorted.

FLI

Field length indicator for a segment.

PFP

Primary field pointer for a segment.

SFP

Secondary field pointer for a segment.

Original Display and Actual Display

The original display is the display for which the display-bound data stream was originally made by the host application (or a 4700 application program that generates display-bound data streams). The actual display refers to the terminal with which the controller application program is actually communicating. If the program is receiving a data stream from a 3270 oriented host application program and sending the converted data stream to a 4704 terminal, for example, the 3270 is the original display and the 4704 is the actual display.

Actual Display Coordinates

The content of a field that shows the location of the top left character of the "window" area on the actual device screen.

Image Buffer Window Coordinates

The content of a field that shows the location of the top left character of the "window" area in the image buffer.

Presentation Position Field (PPF)

The row-column address in the image buffer that corresponds to the cursor position. This character is the target location for PUTFLD operations and the source location for GETFLD operations. When the controller application program issues a PUTFLD instruction, DATSM moves the data in the data buffer into the image buffer beginning at the row and column position indicated by the presentation position field. When the controller application program issues a GETFLD instruction to get the current field, DATSM moves the image buffer field designated by the PPF into the user's data buffer.

DATSM Control Areas

Three separate control areas are required for each 4700 work station using the DATSM instructions:

1. DATSM machine segment (DMS)
2. DATSM image buffer
3. Field control table (FCT)

The DATSM machine segment (DMS) is a communication area used by both the 4700 application program and the DATSM instructions to pass mapping, control, and status information. There is one image buffer for each logical work station that uses DATSM. The image buffer resides in user storage and is not directly accessible to the controller application program. The PUTFLD instruction places the data stream (from the A1, B1, and C2 flows -- see Figure 4-1 on page 4-6) into the image buffer in the same arrangement as it would have taken on the originally intended display screen with the attribute positions identified by a start field character, X'1D'. With the GETFLD instruction, you can retrieve fields of data one at a time, all at once, or all of a given class.

The image buffer is an area where DATSM lays out, or "maps" data stream data before displaying it at or after receiving it from the terminal. You must define the necessary storage for one or more image buffers. You can define a single image buffer for all stations, separate image buffers for each station, or both. You *must* define a separate buffer for each station that processes 3270 data streams in controller tracking mode. Refer to "Using DATSM" on page 4-15 for more information on specifying the image buffers.

The field control table (FCT) is a control block used by the DATSM instructions to control data stream mapping and conversion. The controller generates the FCT for a work station during CPGEN if the STATION configuration macro instruction specifies the DATSM parameter. Your program initializes the FCT by issuing the DATSM instruction, PUTDMS. The FCT itself is an internal control table for DATSM; you cannot control its contents from your program. The DATSM instructions control the FCT contents during program operation as it processes the data from the data streams and the image buffer.

DATSM Data Streams

A data stream, as input to or output from DATSM, is either a mixture of commands, orders, and data that produces an image on a display screen, or is a mixture of addresses and data that is returned to the application program from a display device. Each time a 4700 application program requests a data stream mapping by DATSM, it must specify whether the data stream is a 3270 or SCS data stream. It must also specify whether the destination of the data stream is a display or a processor.

DATSM Operation Modes

DATSM operates in several modes. They are:

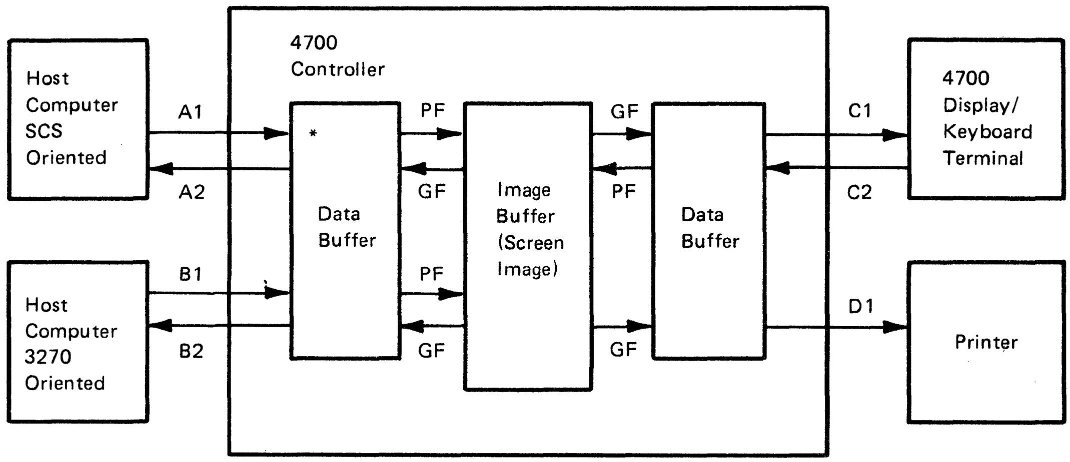
- Base attribute mode
- Kanji mode
- Extended attribute mode, including APL
- Local tracking mode (buffered operation)

Base attribute mode is the default. In this mode, no additional attribute buffer is required. Kanji mode supports the Kanji data stream, and extended attribute mode supports APL and extended color and highlighting. You must specify any mode other than base mode in the PUTDMS instruction before you process data streams.

As far as the physical properties of the 4700 terminal permit, DATSM emulates the field attributes of the 3270 terminal in the 4700 data stream that it prepares. You can use the 4704-2/-3 in either base mode or as in local tracking mode. In base mode, you can program the 4704-2/3 as a 3278. In local tracking mode, keyboard data is written directly to the display rather than through the controller.

Data Stream Flows

Figure 4-1 shows the possible data transfers between the user buffers and the image buffer.



*Defined in user segment space
GF = the GETFLD instruction
PF = the PUTFLD instruction

Figure 4-1. Data Flow Diagram

The labels for the boxes show the possible sources and destinations of data processed by the data stream mapping instructions. The arrows between the boxes show the direction of data flow. The labels for the arrows between the 4700 controller and the computer and terminals are used as reference keys in other sections of this publication.

Data Stream Types

A *display-bound* data stream is a data stream whose destination is a display device, regardless of whether the data stream originates from a host application program or from the 4700 application program.

A *processor-bound* data stream originates from a display device and whose destination is a host application program or a 4700 application program.

A *3270 data stream* is a series of commands, orders, addresses, attribute characters, and data that is exchanged between an application program and a remote 3270 display (in DATSM, the image buffer). This data stream is defined in the *3270 Information Display System Component Description,* GA27-2749.

An *SCS data stream* is a series of orders and data that is exchanged between an application program and a remote SCS display (for DATSM, the image buffer). The data streams used for 4700 displays are a subset of SCS defined in the SNA publications.

A *user-defined data stream* is a subset of SCS. It is a mixture of character data and user defined delimiters. Such a data stream can be used only after fields have been defined by use of one of the preceding types of data stream.

A display-bound data stream can be either a 3270 or SCS data stream when it is the input to DATSM but is always an SCS data stream when retrieved from DATSM. Conversely a processor-bound data stream is always an SCS data stream when it is the input to DATSM but can be either a 3270 or SCS data stream when retrieved from DATSM. Figure 4-1 on page 4-6 illustrates processor-bound and display-bound flows.

Input Data Streams: The following data streams can be the input to DATSM for mapping into the station image buffer when an application program issues a PUTFLD instruction with the appropriate control information initialized in the DMS (see the flow diagram in Figure 4-1). PUTFLD is the only instruction by which data streams can be put into DATSM.

- A1: An SCS display-bound data stream received from a host computer or a 4700 application program or an SCS display-bound data stream that contains user-defined field delimiters received from a host computer or a 4700 application program.

- B1: A 3270 display-bound data stream received from a host computer or a 4700 application program.

- C2: One or more SCS processor-bound fields of data received from a 4700 display.

Output Data Streams: The following data streams can be retrieved from the station image buffer and placed into a user's segment when an application program issues the DATSM GETFLD instruction (see the flow diagram in Figure 4-1). The appropriate control information must be specified in the DMS. A data field can also be placed in a user's data buffer because of PUTFLD instruction processing. For a more complete explanation of this function, see the sections titled "Emulation of 3270 Keyboard" and "Tab Key Emulation".

- A2: An SCS processor-bound data stream or a data stream with fields separated by user-defined delimiters for transfer to a host computer or for use by a 4700 application program.
- B2: A 3270 processor-bound data stream for transfer to a host computer or for use by a 4700 application program.
- C1: An SCS display-bound data stream for transmission to a 4700 display.
- D1: A printer-bound data stream for transmission to a 4700 printer.

The Field Control Table (FCT)

The FCT is a control block used by the DATSM instructions to control the process of data stream mapping and conversion. There is one FCT for each logical work station using the DATSM instructions. The FCT is generated during the CPGEN process by specifying the DATSM operand on the STATION configuration macro. It is initialized with mapping specifications when the DATSM PUTDMS instruction is issued by the controller application program for a logical work station. The FCT resides in user storage not directly accessible to the controller application program.

The FCT consists of:

- A header that contains work information about the areas and data processed by the DATSM instructions
- A body that consists of an individual entry for each field defined in the image buffer

Field Control Table Header

The FCT header contains internal DATSM work pointers, work areas, coordinate positions, counters, window dimensions, image buffer dimensions, and locations of several control areas.

The PUTDMS instruction is used by the controller application program to move mapping information from the DMS into the FCT header. The GETDMS instruction is used by the controller application program to reconstruct the DMS at a user-specified location. The reconstruction is made from the appropriate fields of the FCT. All communication between the controller application program and DATSM is done using four instructions: PUTDMS, GETDMS, PUTFLD and GETFLD.

Field Control Table Entries

Each data field defined in the image buffer is described by a corresponding entry in the FCT body. An entry is generated by the PUTFLD instruction when it processes a data stream; either 3270 or SCS, when the control bit in the DMS is set to select a display-bound data stream. The entry contains:

- The location in the image buffer of the first byte of the field in row, column form
- The length of the field
- A flag byte
- An attribute byte.

The length of the field is set by the PUTFLD instruction by computing the number of bytes between the beginning locations of adjacent fields.

DATSM Machine Segment

The DATSM machine segment (DMS) is a communication area consisting of a header and body. The DMS is used by both the application program and the DATSM instructions to pass mapping, control, and status information.

There is one DMS (at a given time) for each work station that uses DATSM. The DMS resides in user storage in a segment selected by the user (segment 14 is invalid). Its location is passed to DATSM in the segment PFP when the application program initializes the control areas with the PUTDMS instruction. Figure 4-2 is a diagram of the DMS.

If a DATSM instruction is issued before PUTDMS has established the address of the DMS, a program check with code 11 (X'0B') results. Since the address of the DMS is stored each time PUTDMS is issued, the address of the DMS can be altered dynamically by issuing another PUTDMS.

When GETDMS is issued, a copy of DMS is built by DATSM with all fields filled in from the values currently held in the FCT. GETDMS does not reconstruct the DMS header.

Offset	Hex	Field		Part
0	(0)	Control Byte 1	Control Byte 2	DMS HEADER
2	(02)	Status Byte 1	Status Byte 2	
4	(04)	Error Code Byte	SMSEID Value	
6	(06)	Print Line Seq	User Delimiter	
8	(08)	SMSIML Value		
10	(0A)	Field Sequence Number		
12	(0C)	DMSFLG (work state flags)		
14	(0E)	DMSMOD (Mode Field)		
16	(10)	Presentation Position		DMS BODY
18	(12)	Actual Display Dimensions		
20	(14)	Actual Display Window Coordinates		
22	(16)	Image Buffer Window Coordinates		
24	(18)	Window Size		
26	(1A)	Original Display Dimensions		
28	(1C)	Image Buffer Address *		
30	(1E)	Field Control Table Address *		
32	(20)	Reserved	DMS Segment Nr *	
34	(22)	DMS Displacement in Segment *		
36	(24)	Reserved		
38	(26)	Reserved		

Figure 4-2. DMS Diagram

Note: These fields provide DATSM information to the 4700 application programmer for program debugging use.

DMS Header

The DMS header is 16 bytes long. It is used by all the data stream mapping instructions as a user interface by which the application program communicates control information, and the instructions return status, sequence and error information. The general purpose of each byte in the DMS, other than the second, is the same for all DATSM instructions, but the meanings assigned to the control and status bits differ for each instruction. Detailed definitions for these bit assignments are in Chapter 5, "4700 Loop and DCA Assembler Instructions" with the descriptions of the DATSM instructions.

Control Bytes (DMSFLAG)

For all instructions other than GETDMS, both bytes contain flags. See Chapter 5, "4700 Loop and DCA Assembler Instructions" for separate bit assignments for all DATSM instructions. In addition, the control flags used to specify parameters to be transferred from the DMS to the FCT are identified in the descriptions of the DMS parameter fields given below.

Status Bytes (DMSSTAT)

This field is set by DATSM to show the result of a DATSM instruction execution. It can suggest the next action that the application program should take. Refer to the individual instruction descriptions beginning at "DATSM Instructions" on page 5-43 for the meanings of the status bits for each DATSM instruction.

More than one status bit can be set in DMSSTAT by a single DATSM instruction, so the order in which you test the status bits is important. Figure 4-3 shows the recommended order of testing for each DATSM instruction.

DATSM INSTRUCTION:	RECOMMENDED TEST ORDER:
GETFLD	Bit 07 Bit 06 Bit 11 Bit 12 Bit 05 Bit 00 Bit 14
GETDMS and PUTDMS	Bit 07 Bit 06
PUTFLD	Bit 07 Bit 06 Bit 11 Bit 12 Bit 05 Bit 00 Bit 01 Bit 02 Bit 08 Bit 10 Bit 03 Bit 09

Figure 4-3. Recommended DMSSTAT Test Sequences

Error Code (DMSERCD)

This field identifies certain particular error conditions. See error code definitions in Appendix F, "DATSM Sample Program and Error Codes."

SMSEID Value (DMSEID)

The EID identifies the interrupting key used for mapping data in the DATSM image buffer, and as the attention ID (AID) byte for the 3270 data stream produced by GETFLD on the B2 flow (Figure 4-1 on page 4-6). Your program must store the SMSEID value in DMSEID after performing an LREAD instruction.

If SMSEID contains a null character, the operator entered a character into the last field position, causing an automatic read break. If the resulting null enters the DMSEID field on the C2 flow, the PUTFLD instruction executes an autoskip or performs a tab to the right. Otherwise, a null character is invalid in DMSEID. If DMSEID contains zero on the B2 flow, DATSM replaces it with an AID byte of X'60' (no operator action).

Print Line Sequence Number (DMSLSEQ)

This byte contains a line sequence number set by the application program for GETFLD to use to retrieve the corresponding row from the image buffer window. The application program uses this number to retrieve lines of data for the printer. If the sequence number is zero when the GETFLD instruction is invoked, it is set to one, and the image buffer access is to row one of the image buffer window.

After each access, the GETFLD instruction increases the line sequence number by one. When the last line of the window is retrieved, the instruction resets the line sequence number to zero.

User Delimiter Character (DMSDELM)

This byte is a character defined by the application program to be used for delimiting each field in the data stream. The data stream must contain no other orders and can be used only when field characteristics for a buffer image have been predefined by the application program.

SMSIML Value (DMSIML)

If data is being read from a keyboard and the data received is to be passed to DATSM, then the SMSIML value must be stored in this field immediately after the LREAD from the keyboard completes. IML provides the input message length for mapping of the present cursor position. Note that after the SMSIML value is saved, it can be changed by other I/O activity.

Field Sequence Number (DMSFSEQ)

The content of this field can be set by the application program to show the next field to be retrieved. The GETFLD instruction updates this field as follows:

- If this is not a request for the current field (located by the presentation position field), GETFLD increases this field sequence number by one before each entry in the FCT is scanned. This sequence number is increased by one whether the field satisfies the criteria for retrieval or not. Thus DATSM sets the sequence number to the number of the field last retrieved, except in requests for every class of field. In that case, the field sequence number is set to zero when all fields have been retrieved.

One exception is when GETFLD is used to access the current field that is defined by the presentation (cursor) position. In this case, the sequence number is not used, but is set to the number for the field retrieved.

To step through all the fields of an image, set the field sequence number to zero, and the application program makes repeated field requests until the GETFLD instruction returns the "field not found" status.

Work State Flags (DMSFLG)

This field provides work state flags besides those provided in the status flags bytes. It is a two-byte field with all bits reserved except the following:

- Bit 1 (DMS byte 12) set to B'1' shows that the image buffer is formatted. Bit 1 set to B'0' shows that the image buffer is unformatted.
- Bit 2 (DMS byte 12) set to B'1' shows that there is an unprotected area in the image buffer because one or more unprotected fields have been defined by use of an appropriate data stream, or the image is unformatted. Bit 2 set to B'0' shows that there is no unprotected area in the image buffer.
- Bit 6 (DMS byte 12) set to B'1' shows insert mode. Bit 6 set to B'0' shows normal mode.

Mode Field (DMSMOD)

This field records the operating modes of any 4704 Model 2 or Model 3 attached to this station. The mode field records whether or not the attached 4704 is operating in local or controller tracking mode, as well as other states unique to the DCA-attached 4704.

DMS Body

The DMS body is 24 bytes long and is used primarily by PUTDMS to initialize and change the mapping specifications in the FCT. The PUTDMS control flags determine which DMS fields are moved into the FCT. The DMS body also contains application program debugging aids. This debugging information [marked by an (*) in Figure 4-2 on page 4-10] is filled in when each GETDMS instruction is issued. Other fields in the DMS body are left unchanged.

Presentation Position (DMSPPF)

This field contains the presentation position in row, column form. When PUTDMS is issued, this value is copied into the FCT header. This allows the user's programs to adjust the presentation position dynamically. Both PUTFLD and GETFLD set DMSPPF to the current presentation position on instruction completion.

The control bit is DMSPPFL. The startup default value is row 1, column 1.

Actual Display Dimensions (DMSACT)

This field contains the dimensions of the actual display in row, column form. This value can be set by the application program.

The control bit is DMSACTF. The startup default values are the image buffer dimensions. Example: for 3604, Model 4 (X'1040').

Actual Display Window Coordinates (DMSACD)

This field contains the coordinates of the upper left character of the window for the actual display in row, column form. These window coordinates must fall within the actual display dimensions. If they do not, status is returned. If the location specified by this field causes the window area to extend beyond the boundaries of the image buffer, the image that appears on the display is skewed.

The control bit is DMSACDF. The startup default is row 1, column 1.

Image Buffer Window Coordinates (DMSBCD)

This field contains the coordinates of the upper left character of the window in the image buffer in row, column form. The coordinates specified must be within the image buffer or status is returned by DATSM. This value is set by the application program.

The control bit is DMSBCDF. The startup default is row 1, column 1. Example: upper left (X'0101').

Window Size (DMSWIN)

This field contains the window dimensions in row, column form. One window size is used for both the image buffer and the actual display. The window specified must be totally within the image buffer, or status is returned.

The window size contained in this field is used in transmitting data between the image buffer and the display. When transmitting data between the host and the image buffer, the window size used is coincident with the original device size.

The control bit is DMSWINF. The startup default is equal to the image buffer dimensions. Example: For the 3604 Model 4, the window size is X'1040'.

Original Display Dimensions (DMSORG)

This field contains the dimensions of the original display in row, column form. When PUTDMS is issued, it moves the value for the original device dimensions from the FCT header into this field of the DMS, because the original device size is equal to the image buffer. If the original device and image buffer sizes are to be changed dynamically in response to an erase write alternate command, setting the control bit for this parameter causes the instruction to replace the values in the FCT with those from the DMS.

The control bit is DMSORGF. The default is the size of the image buffer established by the CPGEN. Example: for a 3604 Model 7 the size is X'1850'.

Image Buffer Address (DMSIBF)

This field contains the true storage location of the image buffer. The image buffer is accessible to the application program only through the DATSM instructions. This value (the true storage location of the image buffer) is set by PUTDMS during initialization so that a programmer will be able to find the image buffer if he wants to examine it directly using the system monitor.

Field Control Table Address (DMSFCT)

This field contains a halfword address of the FCT. This pointer is not of use to the application program, but it can be used by the application programmer to find the FCT during debugging. It is set by PUTDMS during initialization.

DMS Segment Number (DMSDMSEG)

This byte contains the number of the segment in which the DMS is located. The application programmer can use the value to find the location of the DMS during debugging. This field is set by PUTDMS and reconstructed by GETDMS in the specified user area.

DMS Displacement in Segment (DMSDMSPL)

This field contains the displacement of the DMS from the beginning of the segment in which it is located. This value is an aid when debugging application programs. This field is set by PUTDMS and reconstructed by GETDMS in the specified user area.

Using DATSM

Before you can use DATSM, you must include the following steps in the configuration process:

1. Specify "ERTLS=Y" for any 4700 terminal that is to be used with DATSM. This must be specified in CPGEN DCAnnnn macro instruction.

2. Specify the DSM parameter on the STATION configuration macro instruction to allocate FCT and image buffer space. DSM determines:

   - The largest number of field entries needed at any time in the station's field control table (FCT).

   - The size of the station's normal image buffer (in row/column form).

   - The size of the station's alternative image buffer, used with the "Erase/Write Alternate" command.

   - Any required extended attribute buffer.

   The image buffer size must be as large as the largest display area to be processed by DATSM (that is, the target screen size) and must be as large as the largest display image found in the data streams sent to DATSM.

At CPGEN, the space requirement for FCT is generated. At startup, space is allocated for the FCT and the image buffer in user storage. The application program must perform the following steps:

1. Initialize the DMS, FCT, and image buffer. The controller application program must pass the address of the DMS to DATSM by issuing the PUTDMS instruction with the segment PFP set to the first byte of the DMS and the FLI set to its length. If no parameters are to be changed from the startup defaults, all control bits should be zero.

   This initialization step establishes the DMS address in the FCT header. At this time, you can also set the field and table dimensions and coordinates. If they are not set, DATSM assumes the original and actual device sizes are equal to the image buffer. DATSM also assumes all coordinates are 1,1 and that the window size is equal to the smaller of each of the corresponding dimensions of either the image buffer or the actual device (if you specified the latter). The presentation position is set to 1,1 at startup. You can change any of these values with PUTDMS.

   a. The DMS is in the user's segment; its location is passed to DATSM in the segment PFP when the application program issues PUTDMS.

   b. DATSM saves the DMS location for the station in the FCT and moves the mapping specifications into the FCT. The field entries in the FCT remain empty since no data stream fields are mapped at this time.

2. Pass the user's data stream to DATSM (Flows A1, B1, or C2). After obtaining the input data stream, the application program must set the appropriate flags and fields in the DMS header, set the PFP and FLI to find the data buffer, and issue the PUTFLD instruction:

   a. The PUTFLD instruction passes the buffer's segment number to DATSM. The segment PFP and FLI give the displacement and length of the input data stream in the segment.

   b. DATSM has the location of the station FCT and therefore the location of the station image buffer and DMS.

   c. DMS gives the mapping specifications: such as SCS or 3270 data stream, processor or display-bound, treatment for modified data tags and user flags, presentation position, and field sequence number.

   d. PUTFLD arranges, or "maps", the DATSM input data stream from the user's buffer into the image buffer by coordinating control information from the instruction, FCT, DMS header and controls imbedded in the data stream.

   e. PUTFLD places status flags (and error codes when applicable) in the DMS header and returns control to the application program at the next sequential instruction (the condition code is unchanged).

   f. The application program should test the status flags for successful completion or react to specific status or error conditions.

g. The application program can repeat Step 2d on page 4-16 until the full display or processor image, as required by the program, is mapped into the image buffer.

h. When PUTFLD executes, it produces a six-byte field entry in the FCT for each new field that is mapped. This control information is subsequently used for control of the data retrieval when the GETFLD instruction executes, or for changing the data if a PUTFLD is issued.

3. Retrieve the data stream from DATSM (Figure 4-1 on page 4-6, Flows A2, B2, C1, D1). The application program must place appropriate flags and fields in the station DMS header, provide a receiving buffer of at least 50 bytes in its segment, set the PFP and FLI, and issue the GETFLD instruction:

   a. The GETFLD instruction passes the user's segment number to DATSM. The segment PFP and FLI give the displacement and length of the user's receiving buffer in the segment.

   b. DATSM has the location of the station FCT and therefore the location of the station image buffer and DMS.

   c. The FCT and DMS give the mapping specifications for retrieval: such as type of data stream, processor or display-bound, window size and coordinates, presentation position, field sequence number, field attributes.

   d. GETFLD does the retrieval of the data stream from the station image buffer into the user's buffer by coordinating the control information from the instruction, FCT (header and field entries in the body), and DMS header.

   e. GETFLD places status flags, and error codes when appropriate, in the station DMS header and returns control to the controller application program at the next sequential instruction.

   f. The application program should test the status flags for successful completion or react to specific status or error conditions.

   g. The application program can repeat Step 2e until the full display or processor-bound image as required by its application is retrieved and processed.

Data Streams

This section provides detailed information about data stream input to and output from DATSM. Figure 4-1 on page 4-6 shows data flows with DATSM processing.

A1 -- Display-bound SNA Character String (PUTFLD)

The SNA character string (SCS) display-bound data stream is received from the host or from the controller-resident application program when the DATSM instructions are being used. Two types of data streams are supported by DATSM: The first consists of data and the subset of SCS orders that are supported by the 4700. The second consists of data and user-defined field delimiters.

After the display-bound SCS has been either built or received in the user's data buffer, the application program issues the PUTFLD instruction to map the data into the image buffer.

The DMS control bytes must be set by the application program to show the characteristics of the string:

- SCS display-bound flow
- SCS orders or user delimiters

In processing a data stream, whether it contains SCS orders or user-defined delimiters, the first byte of data found either at the beginning of the string or following an order or delimiter is treated by DATSM as the first byte of a field. All the data from the first byte either to the end of the string or to the next order or delimiter belong to the same field. If there are no orders or delimiters interrupting the data characters, the complete string is treated as one field. The length of a field is determined not by the length of the data, but by the beginning position of the next field.

DATSM supports attributes in SNA Character Strings when X'28C0aa' is found in the data stream where:

- 28C0 is the attribute command and
- aa is the 3270 attribute byte.

Such an attribute byte is stored by DATSM in the FCT entry exactly as are the attribute bytes from 3270 data streams.

Field Entry Generation

The first byte of data following a positioning order is interpreted by DATSM as the beginning of a field, and an entry is generated in the FCT. Because DATSM support is field oriented, attribute bytes and positions are generated for each field. The byte position immediately preceding the first data position in the field is preempted for that purpose. If two fields of data are placed in the buffer contiguously, the last byte of the first field will be lost. For this reason, if fields are to be generated without the use of positioning orders, the last byte of each line of data should be padded with a blank.

If the attribute command sequence is encountered, its attribute byte is stored in the FCT entry and the remainder of the data is moved to the image buffer. The attribute byte is subsequently interpreted as a 3270 attribute byte.

If the data stream consists only of fields and user-defined delimiters, FCT entries for the transmitted fields must be generated before the application program presents this data stream to DATSM for processing. Thus the application program must previously present to DATSM another data stream containing orders for field definition before the data stream with user delimiters can be mapped into the image buffer. If this operation has not been done, DATSM returns a status bit to the application program specifying that the required field does not exist.

SCS Positioning Orders

The SCS positioning orders recognized by the 4700 displays are a subset of SNA orders. The following are supported for DATSM operation. Any hexadecimal values under X'40' that are not listed below are flagged as invalid orders. The PFP is set to the location of the invalid byte and the FLI is set to the residual length of the data stream in the user's segment.

- New line
- Form feed (clear screen)
- Line feed
- Carriage return
- Presentation positioning horizontal absolute
- Presentation positioning horizontal relative
- Presentation positioning vertical absolute
- Presentation positioning vertical relative
- Presentation positioning with erase
- Attribute Sequence
- Select (Treated as a No-op)
- POC (Treated as a No-op)

Print Position Orders

New Line X'15': The new line order causes PUTFLD to erase the remainder of the current line in the image buffer and set the presentation position to the first byte of the next line.

Form Feed (Clear Screen) X'0C': The form feed order causes PUTFLD to clear the image buffer of its contents, to erase all field entries from the FCT, and causes the next data stream retrieved from the image buffer on the C1 flow to be prefixed with the command necessary to clear the display. After this operation, PUTFLD sets the presentation position to the location specified by DMSBCD, the window coordinate for the image buffer.

Line Feed X'25': This order advances the presentation position one line, the column position remains the same.

Carriage Return X'0D': This order returns the presentation position to the start of the current line.

Image Buffer Positioning Orders: Positioning orders cause DATSM to reset the presentation position. If a positioning value exceeds the maximum permitted for the image buffer, the pointer is wrapped in the image buffer.

The X'34' control character is followed by a flag byte and a 1-byte positioning value.

Flag byte:

Bits 0-3 are reserved.

Bit 4:	0 = Position is an absolute value. 1 = Position is relative to the present position.
Bit 5:	0 = Position change is horizontal. 1 = Position change is vertical.
Bits 6-7:	00 = Move cursor to new position without erasing. 01 = Leave cursor at the current position, but erase from the current position the number of bytes or lines specified by the position value. 10 = Move cursor to new position and erase from the old position up to, but not including, the new position. 11 = Reserved.

The positioning byte shows a line or column number. It contains an unsigned binary value ranging from 0 to 255 to be used under direction of the flag byte; a 0 results in no operation.

Note: The flag bits (and the functions they control) can occur in any combination shown above.

Absolute Horizontal Positioning X'3400xx': This order causes PUTFLD to replace the column field of the presentation position with the value contained in the third byte of the order. If the resulting value exceeds the maximum column number of the image buffer, DATSM reduces the value repetitively by the maximum number of columns until the resulting value no longer exceeds that number. The new value is then used as the column position.

Relative Horizontal Positioning X'3408xx': This order causes PUTFLD to add the value contained in the third byte of the order to the column position for the image buffer. If the resulting value exceeds the length of a row of the image buffer, the row position is advanced to the next row, and the column position is set to the value minus the length of one row. If the final sum of the column field exceeds the length of a row, the process is repeated until the column position no longer exceeds the maximum permitted value. If at any time in this process the row field exceeds the maximum number of rows in the image buffer, the row position is set to 1 and the process continues.

Absolute Vertical Positioning X'3404yy': This order causes PUTFLD to adjust the row position to the value contained in the third byte of the order. If the resulting value exceeds the maximum row number of the image buffer, DATSM reduces the value repetitively by the maximum number of rows until the resulting value no longer exceeds that number. The new value is then used as the row position.

Relative Vertical Positioning X'340Cyy': This order will cause PUTFLD to add the value contained in the third byte of the order to the row position. If the resulting value exceeds the maximum row number of the image buffer, the value is repetitively reduced by the maximum number of rows until the resulting value no longer exceeds that number. The new value is then used as the row position.

Presentation Positioning with Erase X'3401xx', X'3402xx': This bit is used with the preceding positioning bits to cause an erase and also change the presentation position. It will cause PUTFLD to set the indicated number of bytes or lines to nulls (X'00'), and position the internal pointer appropriately. If the middle byte is X'01' the presentation position will be left at the last byte erased +1. If the middle byte is X'02', the presentation position will be reset to the position it occupied when the order was encountered. These two options can be used only in combination with one of the four image buffer positioning requests above.

Attribute Sequence Orders

Set Attribute X'28C0aa': This order causes PUTFLD to search the FCT for a field beginning at the current presentation position (set by positioning orders or by data mapping). If there is no FCT entry for this position, PUTFLD generates one. PUTFLD places the byte 'aa' in the FCT entry for the current presentation position. The byte 'aa' must correspond to a 3270 attribute byte.

Set Attribute X'284xaa': This order causes PUTFLD to define the character attribute of type '4x' for the current character position and all succeeding character positions written into the image buffer. The attribute types are defined in detail in "Programming DATSM for APL and Color Displays" on page 4-45.

Start Field Extended X'29ccttaa...ttaa': The byte 'cc' specifies the number of the type-attribute pairs 'ttaa' contained in the order. This order causes PUTFLD to search the field control table for an entry for a field beginning at the current location. If one is found, the base and extended field attributes are reset according to the attributes specified in the order. Any attributes not specified are set to their default values. The types of attributes that may be specified are:

- X'C0' base 3270 field attributes
- X'41' extended highlighting
- X'42' extended color
- X'43' programmable character set.

Note: The attribute types are defined in detail in Appendix F, "DATSM Sample Program and Error Codes" on page F-1.

Graphic Escape X'08': This order causes PUTFLD to interpret the following character to be an APL character. It sets the programmable character set bits in the extended attribute byte for the current character position to B'001', the value defined for the APL character set. This attribute is applied only to the character it precedes, not to succeeding characters.

Select X'04xx': This order is treated as a no-op by DATSM.

POC X'17xxxx': This order is treated as a no-op by DATSM.

The section "Programming DATSM for APL and Color Displays" on page 4-45 contains more information on the attribute sequence orders discussed in this area, particularly for programming APL/color displays.

A2 -- Processor-bound SCS (GETFLD)

The GETFLD instruction permits retrieval of processor-bound data from the image buffer one field at a time or all fields of a class at a time. The DMS control flags must show the following:

- SCS processor-bound flow
- SCS orders or user delimiters
- All changed fields

Then, GETFLD moves all changed fields from the image buffer to the user's buffer beginning each field with the proper SCS positioning order or user-defined delimiter to enable sorting by the application program or host. A changed field is any field altered by data from keyboard (C2 flow —see Figure 4-1 on page 4-6).

Data can also be retrieved from the image buffer a field at a time so that processing can overlap communication line time or to let the application program to process the processor-bound data before it sends the data to the host application. In this case, the controller application program can use GETFLD to retrieve the data fields in sequential order. The field to be retrieved for single field retrieval is determined by the field sequence number in the DMS header. The application program specifies the number of the field to be retrieved in the DMS field sequence number field.

When the application program uses GETFLD to retrieve fields within a specified class, rather than a specific field, the search in the FCT begins with the entry corresponding to the field sequence number plus one. If the complete table is to be searched, the number must be set to zero.

For all retrieval except getting a current field, the instruction increases the field sequence number by one before it examines each field so that after completion of the retrieval, the sequence number is set to the number of the last field retrieved or it is zeroed if all fields of a type were requested.

When data is retrieved one field at a time, the application program can specify whether the data is to be preceded by an address. Depending on the uses the application is to make of the data, either form can be useful.

The application program can also specify that the field to be retrieved is the one containing the current presentation position. The field so retrieved can be preceded by an address or not as the application program shows in the GETFLD control bytes. If the presentation position shows the middle of a field, data is retrieved from that location to the end of the field.

B1 -- 3270 Display-bound Data Stream (PUTFLD)

The PUTFLD instruction maps the content of a display-bound 3270 data stream into the image buffer. PUTFLD scans the data stream for 3270 control characters. These control characters determine the manner in which PUTFLD builds the image buffer and the type of information that is returned to the controller application program at the completion of the PUTFLD operation. This section describes the actions taken by PUTFLD when it meets the 3270 commands and orders during processing.

3270 Data stream

The first two bytes of the 3270 display-bound data stream are the command code (CC) and the write command code (WCC) respectively.

CC	WCC	DATA AND ORDERS

3270 Data Stream Format

3270 Commands

Write X'F1'

PUTFLD places the data stream into the image buffer, beginning with the first byte following the WCC. The FCT entry for each field placed into the image buffer is marked as being display-bound-modified. The output data tag (ODT) associated with the entry is set to B'1' to show that the field content has been changed on the display bound path. If the reset changed data tag (MDT) flag is on in the WCC, PUTFLD turns off the input changed data tags (MDT) in all entries in the FCT.

Erase-Write X'F5'

PUTFLD clears the complete image buffer to null characters, then moves the data stream into the image buffer. PUTFLD also sets a bit in the FCT header flag field so that when the controller application program requests that a data stream be created for the display on the C2 flow, PUTFLD will prefix the data stream with the necessary order or command to clear the display screen.

Erase-Write Alternate X'7E'

In DATSM, this command performs the same function as the Erase-Write (X'F5') command. It uses the current screen size specified by PUTDMS as the alternate screen size.

Erase All Unprotected X'6F'

DATSM replaces the data in all unprotected fields in the image buffer with null characters (X'00'). DATSM also sets the output data tag for each field cleared, so that a later GETFLD for all output-modified fields can be issued to change the display.

Read Modified X'F6'

The PUTFLD instruction sets a bit in the DMS status byte to notify the application program to issue the GETFLD instruction to generate the required processor-bound data stream.

Read Modified All X'6E'

The PUTFLD instruction sets a bit in the DMS status byte to notify the application program to issue the GETFLD instruction to generate the required processor-bound data stream.

Read Buffer X'F2'

The PUTFLD instruction sets a bit in the DMS status byte to notify the application program to issue the GETFLD instruction to generate a full buffer read for a processor-bound data stream, as described in the *3270 Component Description* manual.

TREATMENT of 3270 WCCs

DATSM supports the WCC for reset-modified data tags. All other WCCs must be supported by the application program.

3270 Orders

The 3270 orders processed by DATSM are as follows:

Start Field X'1Daa': The byte following the order, "aa", is an attribute byte. The PUTFLD instruction stores this byte in the FCT if an entry exists for the current buffer address. If no entry for that address exists, PUTFLD creates one and stores the attribute byte in that entry. PUTFLD places the start field code byte, X'1D', in the image buffer at the position that would be occupied by the attribute byte in an equivalent regeneration buffer on the 3271.

Set Attribute X'284xaa': Refer to the description under "Attribute Sequence Orders" on page 4-21.

Start Field Extended X'29ccttaa...ttaa': Refer to the description under "Attribute Sequence Orders" on page 4-21.

Graphic Escape X'08': Refer to the description under "Attribute Sequence Orders" on page 4-21.

Set Buffer Address X'11': PUTFLD alters the form of the address contained in the next two bytes by eliminating the two high-order bits in each byte and combining the remaining bits into a single twelve-bit address. The resulting value is converted into row, column form and used as the current buffer address.

Repeat to Address X'3C': This order is followed by a two-byte base-64 (6 bits) address, which in turn is followed by the character to be repeated. PUTFLD propagates the repetition character through the image buffer, starting with the position currently set in the presentation position field, and continuing up to but not including the position addressed by the second and third bytes of the order. If the order causes defined fields in the image buffer to be overwritten, the FCT entries for those fields are deleted. If the character propagation began in the middle of a field, that field is flagged as display-bound with the MDT flag on.

Erase Unprotected to Address X'12': This order causes the contents of all unprotected fields in the image buffer to be set to null characters, beginning with the current presentation position and continuing up to but not including the character position indicated by the second and third bytes of the order. The ODT is set to B'1' in the FCT entry for each cleared field.

Insert Cursor X'13': PUTFLD places the row-column form of the current image buffer address into the presentation position (cursor address) field of the FCT header.

Program Tab X'05': PUTFLD sets the presentation position field to point to the first data byte of the next unprotected field in the image buffer. If the program tab order follows a data byte, PUTFLD also sets the remainder of the field to null characters.

B2 -- 3270 Processor-bound Data Stream (GETFLD)

AID	CURSOR POS	SBAs and DATA

3270 Data Stream Format

A data stream generated for this flow can take any of three forms. All three begin with a one byte Attention Identifier (AID) and a two byte base 64 cursor position.

A data stream generated as the result of a full buffer read request follows the AID and cursor address with the complete content of the image buffer, including nulls and attribute bytes. Each attribute byte is preceded by a start field order, X'1D'.

A data stream generated as the result of a GETFLD that specifies a read-modified-all request, consists of AID byte, cursor address, and the addresses of all changed fields in SBA form, each address followed by the data from its field. A data stream generated as the result of a GETFLD that specifies neither a full buffer read nor a read changed all, will depend on the content of the DMSEID. The PA keys, the CLEAR key, and the TEST REQUEST key cause a "short read" data stream, consisting of only the AID and cursor position, to be generated. For all other values of DMSEID, the data stream contains the SBA orders, addresses and data of all changed fields.

The DMS control byte must specify:

- A 3270 processor-bound flow
- Type of fields wanted
- The type of read, example: read changed all.

Note that the AID byte is the SMSEID value related to this data stream. Since the SMSEID value in the station machine segment (SMS) can change with other I/O actions, the controller application program must move this value from the SMSEID field into DMSEID immediately after each LREAD for the data streams being processed. If the value in DMSEID is zero, DATSM generates an AID of X'60' (no operator action) for the B2 flow.

C1 -- SCS Display-bound Data Stream (GETFLD)

For this GETFLD request, the application program must specify:

- SCS display-bound
- Wanted fields. For example: all, modified.

If the clear display bit has been set because of the prior receipt of some order that would clear the display screen, the SCS data stream is prefixed with clear screen order (X'0C'). After this, if required, GETFLD will build the data stream in the user's data buffer as follows:

- Each field is preceded by two positioning orders to show the correct location of the data on the screen:

  - one vertical absolute positioning order.

  - one horizontal absolute positioning order.

- The data follows the positioning orders.

- A second pair of orders follows the data to show the cursor position.

If the data stream includes orders or is intended for the printer, when data stream generation is interrupted in the middle of a data field because of a short buffer, GETFLD appends a SELECT (X'0400') to the data stream to prevent an automatic new line when the partial data stream is written to the display or printer. If the data stream generation is interrupted after an order, no automatic new line will occur, so the SELECT is then not appended. After such an interruption because of a short buffer, the instruction can be reissued to complete the data stream.

If the appropriate flag bit in the DMS control bytes is set, GETFLD includes the attributes stored in the FCT in the data stream it generates from the image buffer for transmission to the display. GETFLD effects the non-display attribute by replacing the data from the non-displayable field in the data stream with blanks. Attributes are effected by the inclusion of the hexadecimal string X'28C0aa' where 'aa' is the 3270 attribute byte. If only one field is requested and it is non-displayable, GETFLD returns the field as requested and sets the status flag to show that the field is non-displayable.

C2 -- Processor-bound SNA Character String (PUTFLD)

The PUTFLD instruction processes data from the keyboard in the same manner as a data stream from the host processor (A1 flow) except:

- No FCT entries are generated.
- The MDT flag for the entry corresponding to the incoming data is set to B'1' to show that the field content has been changed on the processor-bound path.

PUTFLD enters the data into the image buffer beginning at the location indicated by the presentation position field. After the data is entered, PUTFLD updates the presentation position field to reflect the change of the cursor position caused by the data entry from the display. Processor-bound data streams from the 4700 display contain data only. There are no orders to be interpreted. There is one field at a time. EIDs play an important role in this flow and must be passed to DATSM by the application program in the DMSEID field of the DMS immediately after the associated LREAD from the keyboard.

Note: DATSM supports a specific list of EID values for keyboard and tab key emulation. See "Emulating The 3270 Keyboard" on page 4-29 and "Tab Key Emulation" on page 4-32 for detailed EID descriptions. For any other EIDs passed to DATSM, a status bit (DMSEOM) is set to tell the application program that such an EID was received.

A zero value in the DMSEID field has a specific meaning to DATSM. It shows a read break, and in the C2 flow, it causes an autoskip operation.

DATSM Programming Considerations

This section discusses the following considerations:

- Buffered terminal modes of operation
- Emulation of the 3270 keyboard
- Emulation of the TAB key
- Use of the user flag
- Adapting the size of the presentation
- Examples of window control
- Storage estimates
- Storage and response time

Buffered Terminal Modes of Operation

A buffered terminal may be driven in either of two modes of operation. The two modes are local tracking mode and controller tracking mode. In controller tracking mode, there is no difference in DATSM instruction usage from standard terminal programming. In local tracking mode there are some differences:

- I/O

  In controller tracking mode, I/O is implemented by the LREAD and LWRITE instructions.

  In local tracking mode, I/O is implemented as follows:

  On an initial PUTFLD, if the command code causes the image buffer to be cleared, there is no I/O. Otherwise, the controller reads the DATSM image buffer.

  If the 'force write' bit is on in the PUTFLD control bytes, the image buffer is written at the completion of the PUTFLD operation. Otherwise, the image buffer is written when a GETFLD (all fields of a class with addresses/display bound) is issued.

- Mode change

  To enter controller tracking mode, the mode bit in the DMS (DMSBFM) is set to B'0' before PUTDMS is issued.

  To enter local tracking mode, DMSBFM is set to B'1' before PUTDMS is issued. If the device is not a buffered terminal, the bit will be reset to B'0' and operation will continue.

- Error checking.

  In controller tracking mode, no I/O status is to be expected.

  In local tracking mode, I/O status may be returned after any PUTFLD or GETFLD. The error code byte in the DMS is set to X'08', the DMS status flags are set to X'0200'. When this error is detected, the device error code appears in the SMSDST field.

Comparison of Controller and Local Tracking Modes

In local tracking mode, no data is placed into the user's segment. The new cursor address is placed into the SMS, the EOM/EID is placed into the SMSEID field. SMSIML is set to zero. A processor-bound PUTFLD is still required in order to pass DATSM the EID, the IML and the new cursor address.

After a display-bound GETFLD for all of a class, the header of the user's segment is set for a zero length LWRITE. This is essentially a NO-OP and if the application is supporting only local tracking mode, the LWRITE may be omitted.

In local tracking mode, EOM/EID values between X'00' and X'40' are rejected with status, since these values are defined for keyboard emulation which is unnecessary in local tracking mode.

In controller tracking mode, the application is started at the asynchronous entry point when the first keystroke is tracked. At this time an application may be doing such things as clearing the display before tracking the data. This is possible because the incoming data is being stored in the device buffer, not on the display. In local tracking mode the application is started at the asynchronous entry point at the time of the first keystroke, but the incoming data is being directed to the display, not to the controller's device buffer. Therefore, the application's ability to manipulate the display without destroying incoming data is restricted since the data will not be read into the controller until the EOM key is pressed.

Emulating The 3270 Keyboard

A controller application program could be written in such a way that the display operator is unaware that the program is communicating with a 3270-oriented host application program. On the other hand, the controller application program could emulate many of the functions of a 3270 terminal, so that an operator already familiar with a 3270 application would find it easy to switch to a 4700 display. The same procedure could serve as a basis for screen management by a 4700-oriented application program.

DATSM emulation of a 3270 keyboard by keyboard and display devices attached to a 4700 Finance Communication System is a matter of:

- Assigning keys on the keyboard device to be equivalent to the keys on a 3270 keyboard.
- Setting up the 4700 translation tables to produce the 3270 attention ID (AID) codes and data from the assigned keys.

The installation can select any of the 4700 display keys and define them as being the equivalent of 3270 keys. As the controller application program obtains the data from the display and passes it to DATSM (by means of PUTFLD), DATSM manipulates the data stream in the image buffer and in some cases, prepares a data stream for transmission to the display, or takes other actions that help simplify the program's task of emulating a 3270 keyboard.

The PUTFLD instruction processes data from the keyboard on the C2 flow (processor-bound data stream from the keyboard). Processor-bound data streams from the 4700 display contain data only. There are no orders to be interpreted. There is one field at a time. EIDs play an important role with this flow. They are passed to PUTFLD by the application program in the DMSEID field of the DMS with each PUTFLD request involving the C2 flow. If the EID is one that DATSM handles (see the list below), PUTFLD will process it as described below. All other EIDs are passed by DATSM to the application program to handle.

For DATSM to provide this emulation support, the application programmer must include each key value in the 4700's translation table in this manner:

```
INTRTBL  (scancode,character,,EOM,EID)
```

where "scancode" represents an arbitrarily selected 4704 key position, a "character" is one of the key values listed on the next page.

Some of the key values are those defined by 3270 architecture; the remainder are hardware function keys for which 3270 has no transmission definition, and which are defined by DATSM.

X'00' Autoskip (result of read break only)
X'01' TAB right key
X'02' TAB left key
X'03' TAB up key
X'04' TAB down key
X'05' Insert mode key
X'06' Insert Reset key
X'07' Delete key
X'08' Erase to EOF key
X'09' Erase input key
X'0A' Cursor left
X'0B' Cursor right
X'0C' Cursor up
X'0D' Cursor down
X'0E' Magnetic stripe input

Key value for 3270 transmission:

X'6D' Clear key

The following table contains the AID bytes that are required to emulate the 3271. With the exception of the CLEAR key, they cause no action on the part of DATSM, but they are included in the data stream prepared for the host processor if the value listed is placed in the DMSEID field.

AID DEFINITIONS			
AID	EBCDIC	Graphic Character	Command Operation
No AID	60	—	Rd Mod
ENTER key	7D	'	Rd Mod
PF 1 key	F1	1	Rd Mod
PF 2 key	F2	2	Rd Mod
PF 3 key	F3	3	Rd Mod
PF 4 key	F4	4	Rd Mod
PF 5 key	F5	5	Rd Mod
PF 6 key	F6	6	Rd Mod
PF 7 key	F7	7	Rd Mod
PF 8 key	F8	8	Rd Mod
PF 9 key	F9	9	Rd Mod
PF 10 key	7A	:	Rd Mod
PF 11 key	7B	#	Rd Mod
PF 12 key	7C	@	Rd Mod
PF 13 key	C1	A	Rd Mod
PF 14 key	C2	B	Rd Mod
PF 15 key	C3	C	Rd Mod
PF 16 key	C4	D	Rd Mod
PF 17 key	C5	E	Rd Mod
PF 18 key	C6	F	Rd Mod
PF 19 key	C7	G	Rd Mod
PF 20 key	C8	H	Rd Mod
PF 21 key	C9	I	Rd Mod
PF 22 key	4A	¢	Rd Mod
PF 23 key	4B	—	Rd Mod
PF 24 key	4C	<	Rd Mod
PA 1 key	6C	%	Short Read
PA 2 key	6E	>	Short Read
PA 3 key	6B	,	Short Read
CLEAR key	6D	—	Short Read

Figure 4-4. Program Function (PF), Program Access (PA), and ENTER Keys

DATSM processes the EIDs defined for keyboard emulation. All other EIDs are passed to the application program with the appropriate status flag set on. When the application program issues the GETFLD instruction for a 3270 processor-bound data stream, DATSM uses the value from the DMSEID field as the AID byte in the 3270 data stream. It is the responsibility of the application program to make sure that any value in DMSEID that is so used is recognizable to the host program. DATSM converts a DMSEID value of X'00' to X'60' (no operator action when using it to produce an AID byte.) The application program must issue the GETFLD instruction to specify what kind of data stream is to be produced for data flow A2 or B2.

Enter Key, EID = X'7D'

When PUTFLD receives the EID value for the enter key, it sets a status bit to inform the application program that the end of a logical message has been reached and that the content of the image buffer is ready for processing by the application program in the controller, or for transmission to the application program in the host computer. PUTFLD also resets the insert mode, if on. This treatment is much like that given for the EID values for the PA and PF keys.

CLEAR Key, EID = X'6D'

PUTFLD fills the image buffer with nulls, clears the FCT entries and generates a clear screen character followed by cursor positioning orders to clear the 4700 display and position the cursor to the upper left corner of the window. PUTFLD resets insert mode, if on, and then sets the status bytes to indicate an EOM and to indicate that an LWRITE of the clear screen character to the display is required.

Tab Key Emulation

TAB key emulation requires a double exchange between the application program and the PUTFLD instruction. When DATSM recognizes the EID as one of the TAB key characters:

- It sets the presentation position field to the tabbed-to field.
- It generates an SCS data stream in the user buffer to reset the cursor position to the beginning of the tabbed-to field on the display.
- It sets flags to indicate to the application program that there is data to be written to the display.
- It sets the segment PFP and SFP in anticipation of an LWRITE by the application program.

The application program should issue an LWRITE to the display without an intervening LREAD. This causes the cursor to be positioned to the tabbed-to field. Then the application program reissues the PUTFLD instruction. This is the first exchange.

Upon being reissued, PUTFLD moves the content of the newly located field from the image buffer to the user's data buffer so that new data from the keyboard can be written over the old data. Note that PUTFLD has performed the same function as a GETFLD instruction to move the tabbed-to data field into the user's buffer. PUTFLD sets a flag in the status byte of the DMS to indicate that a read from the display is to be issued by the application program and returns control to the application program. For this operation PUTFLD moves only data located in the display window.

The application program performs the LREAD to the display. This completes the second exchange.

In both exchanges, the segment header is set appropriately by PUTFLD so that an LWRITE or LREAD can be issued by the application program without further action. For this purpose, PUTFLD uses as much of the segment area as is required from the beginning of data pointer (PFP) to the end of the segment, without reference to the FLI setting for the incoming data. For this reason, it is the responsibility of the application program to ensure that sufficient segment space is available for this use.

When tabbing and if there is no unprotected field in the image, PUTFLD sets the pointer to 1,1 and returns to the application program with an error code in the DMS header.

AUTOSKIP Tabbing Function

The 4700 controller automatically terminates an LREAD when the last character position of the read is filled. When this occurs the EID is set to zero. After processing the last byte of data, PUTFLD tabs right and sets up the segment pointers for an LREAD of the next unprotected field. If field segmenting was in effect and the read break was caused by reaching the end of a window row, the cursor is positioned to the next row of the field in the window, or to the beginning of the next field. The user must code the operand ERTLS=Y on the DEVxxxx configuration macro so that an LREAD processes correctly for DATSM use in emulating the 3270.

TAB Right Key, EID = X'01'

PUTFLD sets the presentation position field to point to the next unprotected field to the right of the current presentation position in the image buffer. If there are no unprotected fields in the image buffer when this or any of the other tab key orders are encountered, PUTFLD sets the presentation position field to point to the beginning of the image buffer (Row 1, Column 1).

TAB Left Key, EID = X'02'

PUTFLD sets the presentation position field to point to the first data byte of the nearest unprotected field to the left of the current position. If the current position happens to be the first byte of the field, PUTFLD moves the pointer to the beginning of the next unprotected field to the left. If the current pointer is pointing elsewhere in the field, PUTFLD moves the pointer to the beginning of the same unprotected field.

TAB Up Key, EID = X'03'

PUTFLD sets the presentation position field to point to the first byte of data in the leftmost unprotected field in the first line above the current line that contains an unprotected field. If there is no unprotected field defined above the current line, the search moves (wraps) to the bottom line and continues upward.

TAB Down Key, EID = X'04'

PUTFLD sets the presentation position field to point to the first byte of data of the leftmost unprotected field in the line immediately before the current line. If there is no unprotected field located below the current line, the search moves (wraps) to the top of the image and proceeds downward.

Insert Mode Key, EID = X'05'

When PUTFLD receives an insert mode key command (X'05') in a 3270 data stream it enters insert mode processing as follows:

1. Sets the PFP and FLI for a one character LREAD KB by the application program.

2. Passes control to the application program with status flags set as follows:

   a. Application program should issue LREAD KB.

   b. Application program should then reissue PUTFLD to pass the one character to DATSM.

3. When the application program has done the processing outlined above, PUTFLD processes the character to be inserted as follows. It inserts the character received from the application program into the image buffer, starting at the current presentation position. As PUTFLD inserts each character, it increases the presentation position by one. It shifts the characters to the right of the current position to make room for the new character. A null character is removed from the right end of the field for each character that is inserted. PUTFLD then returns to Step 1, which continues the insert process.

The processing of insert mode continues until there are no more available nulls in the field or an INSERT RESET key EOM is received.

The DEV4704 configuration macro must specify ERTLS=Y.

Insert Reset Key, EID = X'06'

PUTFLD terminates the insert mode process described above.

Delete Key, EID = X'07'

PUTFLD shifts the characters that are to the right of the current presentation position one byte to the left, eliminating one character. A null character is inserted in place of the vacated right-most byte of the field. If the field exceeds one line of the screen, data on succeeding lines are not affected. The field is flagged in the FCT entry as being display-bound-modified. A 4700 display-bound data stream is generated for the field and placed in the data buffer to be written to the display.

Erase to End-of-Field Key, EID = X'08'

PUTFLD replaces existing characters in a field with null characters beginning at the current presentation position and continuing through the end of the field. If the screen is unformatted, PUTFLD clears the image buffer from the current presentation position to the end of the buffer. The presentation position will be unchanged. PUTFLD generates a 4700 display-bound data stream for transmission to the display.

Erase Input Key, EID = X'09'

PUTFLD replaces the contents of all unprotected fields with nulls. A data stream is generated by PUTFLD that when written to the display clears the affected fields. PUTFLD sets the presentation position to the first data byte of the first unprotected field. If there are no unprotected fields defined, the presentation position is set to 1, 1. If the screen is unformatted, the entire screen is cleared and the presentation position is set to 1, 1.

Cursor Movement keys:

The cursor movement keys differ from all other keyboard emulation EIDs in that as a result of their use, the presentation position may rest on an attribute byte or in a protected field. The application program must detect this condition and protect against improper alteration of the screen contents of the field thus entered. In the case of the presentation position lying on an attribute byte, status bit 12 will be set to B'1', the same indication that is used for the field not found condition. If the position lies within the data area of a protected field, bit 14 will be set to B'1'.

CURSOR LEFT KEY, EID = X'0A'

The column number of the presentation position is reduced by one. If the resulting column number is less than that of the left side of the image buffer window, it is replaced with the maximum column number of the image buffer window, and the row number is decreased by one. If the resulting row number is lower than that of the top row of the window, it is set to the number of the last row of the image buffer window.

CURSOR RIGHT KEY, EID = X'0B'

The column number of the presentation position is increased by one. If the resulting column number is greater than the maximum column number of the window, it is replaced with a value of one, and the row is increased by one. If the resulting row number exceeds the maximum number of rows in the window, it is changed to the number of the top row of the image buffer window.

CURSOR UP KEY, EID = X'0C'

The row number of the presentation position is reduced by one. The column number is unchanged. If the resulting line number is less than that of the top line of the image buffer window, it is replaced with that of the last line of the image buffer window.

CURSOR DOWN KEY, EID = X'0D'

The row number of the presentation position is increased by one. The column number is unchanged. If the resulting line number is greater than that of the last line of the image buffer window, it is replaced by that of the top line of the window.

MAGNETIC STRIPE INPUT, EID = X'0E'

To emulate 3270 magnetic stripe support, the start of stripe character may be defined as EOM/EID X'0E'. When this EID is received by PUTFLD on the processor-bound path, it will generate a non-displayable attribute for the current cursor position if no field already exists at that location and then set up a read for the resulting field.

Inconsistencies with 3270

The use of the characteristics of the terminal hardware, DATSM 3270 keyboard emulation is inconsistent in some operations with the 3271. The following list describes the inconsistencies.

- Entry of invalid characters into numeric fields on the 3270 causes the keyboard to lock and refuse to accept the invalid character. DATSM does not see the invalid character until the completion of the read. For that reason, the invalid characters are mapped into the image buffer, but the presentation position is set to the location of the leftmost invalid character and DATSM will refuse to recognize any EOM/EID character such as a tab, ENTER, or INSERT until the invalid character has been replaced by a valid one.

- In addition to the keyboard reset, a separate INSERT RESET key is required to leave insert mode.

- Insert mode is terminated by the receipt of an EOM/EID.

- There is no automatic signal light for insert mode. An application program can use the indicators for this purpose.

- Autoskip is enforced for all protected fields. Autoskip is extended to the cursor movement keys such as cursor left and cursor right. DATSM does not move the presentation position into a protected field.

User Flag

The user flag is for use by application programs that require local processing of fields in the controller before the programs transmit them to either the host processor or to the terminal. The user flag can be used as a search argument by GETFLD. The flag is set by PUTFLD. When the application program sets this flag in the DMS prior to issuing the PUTFLD instruction for display-bound flow, PUTFLD turns on the user flag in the corresponding FCT entries. If more than one field is processed by PUTFLD, each field is flagged.

The user flag can be turned off only by removing the field entry, which occurs each time an erase-write or clear screen order is sent to DATSM or when the field beginning is overwritten by data from a display bound data stream.

Presentation Size Adaptation

The data stream conversion instructions aid a controller application program that is presenting a data stream intended for one screen size to a device having a different screen size.

Using parameters initially supplied by the controller application program, DATSM keeps track of the rectangular boundaries of the two screens. When a data stream for a large screen (such as the 3270 Model 2) is to be sent to a smaller screen (such as the 4704 Model 4 or the 3270 Model 1), the controller application program selects which rectangular subset of the larger display is to be used. When the data stream is converted, DATSM prepares a data stream that has the correct dimensions for the smaller screen.

The controller application program could, for example, define a protocol between itself and the operator of the smaller display in which the operator requests that the display be shifted up and down or from side to side.

When a data stream for a small screen is to be presented on a larger screen, the controller application program selects which rectangular subset of the larger display is to receive the data.

The rectangular boundaries and subareas that are used by the controller and the controller application program can be visualized as three rectangular areas.

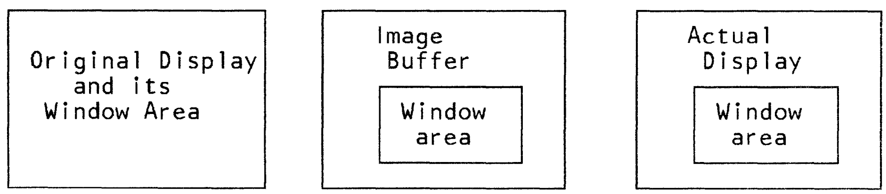


Figure 4-5. Logical Space Definition

The outer rectangle of each part of the figure represents the total area of the described space. This size is set in the field control table header by means of the PUTDMS instruction, and is set in row, column form. The inner rectangle represents the area of each space that is to be addressed by the data stream conversion instructions. In this publication, the smaller area is referred to as the "window," and it is described by two parameters set by means of the PUTDMS instruction. They are the window coordinates and the window size. Both are maintained in row, column form. The window coordinates specify the position of the upper-left corner of the window.

The leftmost rectangle represents the display area for which the data stream was originally intended and is used for processing data streams to and from the host or the application program. The center rectangle represents the internal display image, and the rightmost rectangle represents the display area of the device to which the data stream is actually being sent.

The window size is meaningful when a data stream is being retrieved from the image buffer and prepared for a terminal device. The window determines the positioning of display-bound data. The window process uses three values. Two of the values are the row, column positions of the upper left corner of the window for each of the logical areas, image and actual. The third value is the size of the window.

The window size and position must be adjusted by the application program if it is necessary to make changes, since DATSM will make no assumptions concerning device size or positioning. The application program has maximum flexibility in sub-setting data and positioning it, as long as the areas addressed are in a rectangular form.

Programming Examples of Window Control

Overall View of Using Windows

The following diagrams illustrate various uses of the window function. The diagrams use the following conventions:

////// represents data mapped from the data stream
XXXXXX represents data prepared as a window for display

The window area can be adjusted for display. The window used to map data to or from the host processor is always the same size as the original device.

```
/////////////////   XXXXXXXXXXXXXXXXX   XXXXXXXXXXXXXXXXX
/////////////////   XXXXXXXXXXXXXXXXX   XXXXXXXXXXXXXXXXX
/////////////////   XXXXXXXXXXXXXXXXX   XXXXXXXXXXXXXXXXX
/////////////////   XXXXXXXXXXXXXXXXX   XXXXXXXXXXXXXXXXX
/////////////////   XXXXXXXXXXXXXXXXX   XXXXXXXXXXXXXXXXX
/////////////////   XXXXXXXXXXXXXXXXX   XXXXXXXXXXXXXXXXX

ORIGINAL DISPLAY     IMAGE BUFFER       ACTUAL DISPLAY
    24 X 80            24 X 80             24 X 80
```

Figure 4-6. Original and Actual Displays Equal

```
/////////////////   XXXXXXXXXXXXXX///   XXXXXXXXXXXXXX
/////////////////   XXXXXXXXXXXXXX///   XXXXXXXXXXXXXX
/////////////////   XXXXXXXXXXXXXX///   XXXXXXXXXXXXXX
/////////////////   XXXXXXXXXXXXXX///   XXXXXXXXXXXXXX
/////////////////   /////////////////
/////////////////   /////////////////

ORIGINAL DISPLAY     IMAGE BUFFER       ACTUAL DISPLAY
    24 X 80            24 X 80             16 X 64
```

Figure 4-7. Actual Display Smaller than Original

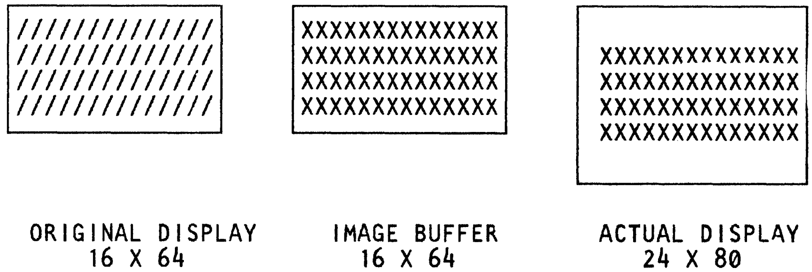


Figure 4-8. Original Display Smaller than Actual

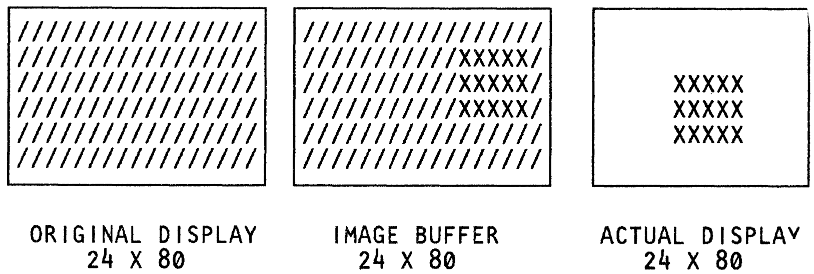


Figure 4-9. Subarea of Image Buffer Displayed

Large Image Buffer, Small Actual Device

Assume that the display for which the data stream was originally intended for a 1920 character display (24 rows high and 80 columns wide), and that the device to which the data stream is actually being sent is a 1024 character 4700 (16 rows high and 64 columns wide). The image buffer has been defined to match the size of the largest area addressed, 1920 bytes, 24 X 80 characters.

Values in the DMS are in hexadecimal format. Addresses and sizes are expressed as row, column numbers with a maximum value of 255 decimal (X'FF'). The following values are set by the application program by means of the PUTDMS instruction:

I-Buffer coordinates 1,1
Actual coordinates 1,1
Original dimensions 24,80
Actual dimensions 16,64
Window dimensions 16,64

Values in the DMS are of one origin. Transferred to the field control table, most of the values remain so with the exception of the coordinate positions. These values are altered to zero origin numbers only in the field control table. When DATSM receives data from the terminal, it must set the window size to 16,64. If the coordinate position for the target area plus the length of the window area exceeds the maximum size of the target area, DATSM returns an error status bit with an accompanying error code to specify the error.

```
Coord 1,1                   Coord 1,1
ABCDEFGHIJKLMNOPQRST        ABCDEFGHIJKLMNOPQRST
UVWXYZ0123456789ABCD        UVWXYZ0123456789ABCD
EFGHIJKLMNOPQRSTUVWX        EFGHIJKLMNOPQRSTUVWX
YZ0123456789ABCDEFGH        YZ0123456789ABCDEFGH
IJKLMNOPQRSTUVWXYZ01        IJKLMNOPQRSTUVWXYZ01
23456789ABCDEFGHIJKL        23456789ABCDEFGHIJKL

ORIGINAL DISPLAY            IMAGE BUFFER
    24 X 80                   24 X 80
```

Figure 4-10. Window Equal to Original Device

PUTFLD has mapped the entire data stream into the image buffer, it returns a status code of X'0100' to indicate the successful completion of the operation.

Assume that the application writes the received data to the 4700 display. Since the display area is too small to receive all of the information, the choice is made first to display the left quadrant of the image buffer. For this purpose, the coordinates are not changed, but the window size is reduced to the dimensions of target device. The application program issues PUTDMS to set the window size to 16, 64.

```
Coord 1,1   Window 16,64    Coord 1,1
ABCDEFGHIJKLMNOPQRST        ABCDEFGHIJKLMNOP
UVWXYZ0123456789ABCD        UVWXYZ0123456789
EFGHIJKLMNOPQRSTUVWX        EFGHIJKLMNOPQRST
YZ0123456789ABCDEFGH        YZ0123456789ABCD
IJKLMNOPQRSTUVWXYZ01
23456789ABCDEFGHIJKL

  IMAGE BUFFER              ACTUAL DEVICE
    24 X 80                   16 X 64
```

Figure 4-11. Window for Upper Left Quadrant

If the application needs to display the lower right quadrant of the image buffer, it sets the coordinates of the image buffer to 16,64 and the window size to 8,16 which is the size of the image buffer area to be displayed without having the quadrants overlap. The application issues PUTFLD with flags set to request all fields in the image buffer and a 4704 display-bound data stream.

```
Coord 16,64    Window 8,16    Coord 1,1

ABCDEFGHIJKLMNOPQRST           YZ01
UVWXYZ0123456789ABCD           IJKL
EFGHIJKLMNOPQRSTUVWX
YZ0123456789ABC
IJKLMNOPQRSTUVW YZ01
23456789ABCDEFG IJKL

   IMAGE BUFFER                 ACTUAL DEVICE
     24 X 80                       16 X 64
```

Figure 4-12. Window for Lower Right Quadrant

Since the request is for all fields, PUTFLD precedes the generated data stream with a clear screen character to clear all presently displayed data and replace the screen image with the data from the defined window. If the application program were to display the upper right quadrant, the arrangement might be to set the coordinates for the image buffer to 1,64, the window size to 16,16, and the coordinates for the target display to 1,1. This would result in the upper right quadrant of the image buffer being displayed in the left hand side of the actual device.

```
Coord 1,64    Window 16,16    Coord 1,1

ABCDEFGHIJKLMNO QRST          QRST
UVWXYZ012345678 ABCD          ABCD
EFGHIJKLMNOPQRS UVWX          UVWX
YZ0123456789ABC EFGH          EFGH
IJKLMNOPQRSTUVW
23456789ABCDEFGHIJKL

   IMAGE BUFFER                 ACTUAL DISPLAY
     24 X 80                       16 X 64
```

Figure 4-13. Window for Upper Right Quadrant

If only one window at a time is sent to the actual device, the processor-bound data from the terminal is mapped to the image buffer with identical settings for coordinates as are used for the display-bound strings. In this way, the incoming data will be put into the correct positions in the image buffer.

Small Image, Large Actual Device

The data within a field occupying more than one line of a window can be retrieved in either of two modes. In the standard mode, all of the data that appears within the window is retrieved. In segmented mode, the data contained within the window portion of the field is retrieved one line at a time. This function is useful when data is to be read from a field occupying more than one line of a window that is smaller than the actual device on which it appears.

Internally, a sample field might look like this:

```
THIS |S A FIELD THA| RUNS
FOR M|RE THAN ONE L|INE.
```

Image Buffer Mapping

A GETFLD current without field segmenting will have the following data in its buffer:

```
S A FIELD THARE THAN ONE L
```

The precise form of the data is determined by the fact that GETFLD retrieves data only from within the window area in the image buffer. Data typed into the keyboard is laid over this data as follows:

```
XXXXFIEXXXXHARXXXXAN XXXXL
```

This data is mapped into the image buffer in the following manner:

```
THIS |XXXXFIEXXXXHA| RUNS
FOR M|RXXXXAN XXXXL|INE.
```

As the example shows, the data is mapped into the correct position in the image buffer. Because the hardware is unaware of the window size, however, the display screen looks like this after the input operation:

```
     S A FIELD THA                XXXXFIEXXXXHA XXXX
     RE THAN ONE L           XXXX RE THAN ONE L
```

BEFORE INPUT   AFTER INPUT

You can restore the display appearance by issuing a GETFLD ALL FIELDS and writing the resulting data stream to the display. The distortion of the display contents can be avoided by use of field segmentation. If the segmenting control bit in both GETFLD and PUTFLD is set to B'1', the GETFLD operation moves only one line of the field at a time. This means that each line is a separate read operation, and between read operations, the cursor is positioned at the left side of the window. The resultant tracking preserves correspondence between the display contents and the image buffer:

```
THIS  S A FIELD THA  RUNS
FOR M RE THAN ONE L INE.
```

IMAGE BUFFER MAPPING. Using the same sample field as in the prior example, the user data buffer contains only one line of the field when a GET LD CURRENT FIELD with segmenting is issued.

```
S A FIELD THA          XXXXFIEXXXXHA
```

BEFORE LREAD   AFTER LREAD

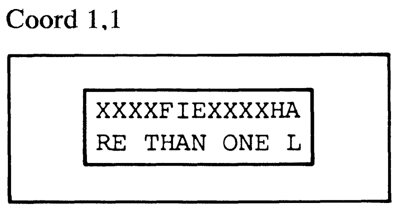
When an LREAD is issued for this portion of the field, the display has the following appearance:

Coord 1,1	Coord 1,1
S A FIELD THA RE THAN ONE L	XXXXFIEXXXXHA RE THAN ONE L
BEFORE INPUT	AFTER INPUT

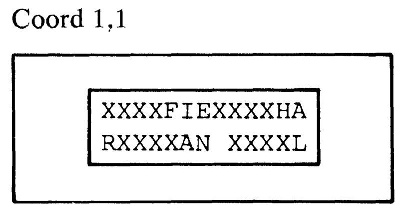
If field segmenting is specified, following the read break at the end of the line, the application program issues a PUTFLD instruction to map in incoming data into the image buffer. After mapping the data, PUTFLD performs a tabbing operation, producing an address string to be written to the display to position the cursor at the start of the next line of the window. A status bit is set to instruct the application to write the address string to the display and to reissue the instruction. Upon reissue, PUTFLD performs a GETFLD CURRENT function which produces the following contents in the user's data buffer. It also sets a status bit to instruct the application program to read from the keyboard into the prepared segment.

RE THAN ONE L	RXXXXAN XXXXL
BEFORE LREAD	AFTER LREAD

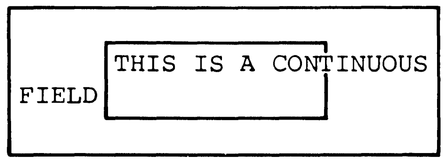
Before and after the second LREAD, the display has the following appearance.



BEFORE INPUT



AFTER INPUT



DISPLAY APPEARANCE

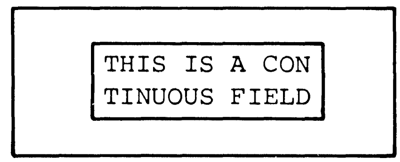


IMAGE BUFFER MAPPING

Display Design

The amount of time required to process each field on the display screen is directly proportional to the length of the field. Regardless of the number of characters entered into a field by the operator, the entire field must be processed by DATSM. Because of this, fields ought to be defined to be only as necessary as for the data expected.

Note that the Insert and the Delete functions require that the entire field be processed from the current presentation position to the end of the field for every character inserted or deleted. Additional I/O exchanges are also necessary for each character, which results in poorer response time.

Programming DATSM for APL and Color Displays

DATSM instructions and DCA support the Graphic Escape (X'08') for APL, and the Start Field Extended (X'29'), and the Set Attribute (X'28') data stream orders. (DATSM also supports Modify Field, X'2C'.) The formats of the instructions follow.

Graphic Escape

08	char.

The Graphic Escape order indicates that the following byte in the data stream is to be read as an APL character, and causes both the DATSM instructions and the DCA code to set the programmable character-set bits to B'001' in the currently-addressable, extended-character attribute byte. This attribute is not inheritable when specified by means of the Graphic Escape order.

Start Field Extended

29	count	type	attr	...	type	attr

The following types are supported for the Start Field Extended order:

X'C0' 3270 type attribute
X'C1' (reserved)
X'41' Highlighting attribute
X'42' Color attribute
X'43' Programmed Character Set

The Start Field Extended order generates a standard field attribute byte in the regeneration/image buffer, and generates an extended field attribute byte in the extended attribute buffer. Any attributes that are not expressed in the order are given their default values (nulls). That is, all bits that represent the unexpressed attribute are set to 0. If the count is 0, the system sets field attributes and extended field attributes to their default values. If the count is zero, DATSM sets all attribute bytes to zero.

Set Attribute

28	type	attr

The following types are supported for the Set Attribute order:

X'00' Reset (the attribute following must also be X'00')
X'C0' 3270 type field attribute
X'41' Highlighting character attribute
X'42' Color character attribute
X'43' Programmed Character Set character attribute

Note: The three character sets available are basic 3270, APL, and Kanji. Loadable character sets are not supported.

Attributes specified in a character attribute byte in the extended attribute buffer cancel the attributes specified in the extended field attribute byte. When no attribute is specified in the character attribute byte, the field attribute is the default. If neither the field nor the character attribute bytes specify an attribute, the default value is used.

Character Attribute Inheritance

Every 8-bit attribute byte contains three attribute specifications as follows:

```
Bits 0,1 Highlighting

         00xxxxxx   Normal (no highlighting)
         01xxxxxx   Blink
         10xxxxxx   Reverse video
         11xxxxxx   Underline character

Bits 2-4 Color

         xx000xxx   Base color
         xx001xxx   Blue
         xx010xxx   Red
         xx011xxx   Pink
         xx100xxx   Green
         xx101xxx   Turquoise
         xx110xxx   Yellow
         xx111xxx   White

Bits 5-7 Programmed Character sets

         xxxxx000   Base ROS (184 character)
         xxxxx001   APL ROS (128 character)
         Others     Reserved
```

Character attributes are inheritable. That is, when an attribute is specified for a given character, all characters in the data stream following are given the same attribute until another attribute specification of the same type either resets it by specifying a null value or changes it by specifying a different value.

Note: Although DATSM supports types X'41', X'42', and X'43', there is no support for loadable character sets.

PUTFLD--Outbound Operation

PUTFLD supports the set of orders required for extended attributes for both 3270 and SNA data streams.

Erase Orders: All commands and orders that clear the image buffer also cause PUTFLD to clear the extended attribute buffer.

Start Field Extended: When the Start Field Extended order is encountered in the display data stream, PUTFLD performs the same consolidation procedure on the field attribute that it performs on the character attribute. The extended field attribute is placed in the extended attribute buffer after all attribute pairs in the Start Field Extended string have been combined. If any attributes are not specified in the string, they are set to their default values before being stored. Attributes are placed in both the image buffer and the extended attribute buffer.

Set Attribute Order: PUTFLD combines the attribute bits for the type specified (highlighting, color, or character set) with the existing current character attribute.

Character Data: PUTFLD determines whether the mode of operation is extended attribute or normal. If the mode of operation is extended attribute, PUTFLD locates the corresponding position in the extended attribute buffer and writes the accumulated character attribute to that location. PUTFLD then stores the character data and advances the image buffer pointer, wrapping to the start of the buffer if necessary.

Graphic Escape: When a graphic escape control character is detected in the data stream, PUTFLD retrieves the saved character attribute byte and ORs in the APL bit. PUTFLD stores the attribute byte into the position in the attribute buffer corresponding to the current image buffer position. The image buffer pointer is unchanged after this operation. The byte with the ORed bit is not saved; it is not inheritable if it was set because of Graphic Escape.

GETFLD--Outbound Operation

When DATSM is operating in extended attribute mode, field attributes are sent to the DCA device processor by means of the Start Field Extended code, X'29'. When an attribute is to be generated, DATSM GETFLD code has already reset the address to the attribute position from the location of the first byte of data. At this point, GETFLD retrieves the extended attribute byte and generates the Start Field Extended code to communicate both the extended and the normal attributes. The Start Field Extended code causes the image/regeneration buffer address to be increased by one after the attribute bytes have been inserted.

When GETFLD retrieves a byte of data from the image buffer, if operation is in the normal mode, GETFLD places the byte in the data stream. If operation is in extended attribute mode, GETFLD compares the new character attribute to the saved character attribute and, if there is a difference, generates the necessary attribute definition strings to signal the change in the data stream. If the APL bit is on in the new character attribute, the called subroutine generates a graphic escape character in the data buffer.

PUTFLD--Inbound Operation

Keyboard Input: When data is received from the keyboard in APL mode without a preceding Graphic Escape order, the APL bit is reset in the corresponding extended attribute byte. If there is a Graphic Escape order in the data stream, the APL bit is set.

The rules for attribute inheritance on input from the keyboard are the same as those for output. If no set-attribute order is received from the keyboard, all incoming data is paired with null extended attributes.

Set Attribute: To avoid three-byte control strings from the input translation table (X'284xnn'), the scan codes for the Extended Attribute Set keys are defined as EOM/EID. The values for each of the attributes is given below.

Highlighting

X'10' Normal (no highlighting)
X'11' Blink
X'12' Reverse
X'13' Underline

Programmed Symbol Set

X'20' Base EBCDIC character set
X'21' APL character set

Color

X'30' Base color
X'31' Blue
X'32' Red
X'33' Magenta
X'34' Green
X'35' Turquoise
X'36' Yellow
X'37' White

When an EID that specifies an attribute is received, the corresponding bits are saved. When data is mapped into the image buffer, the attribute bits replace the attribute byte already located in the extended buffer position corresponding to the data position.

Graphic Escape: When a Graphic Escape character is detected in a data stream received from a device, DATSM retrieves the current character attribute, OR the APL bit onto it, and stores the resulting attribute byte in the attribute buffer position corresponding to the current image buffer location. The image buffer pointer is not advanced after the attribute byte is stored. When the APL bit is set because of the receipt of a Graphic Escape order, it will not be saved; in this case it is not an inheritable attribute.

GETFLD--Inbound Operation

GETFLD processes the APL bit and the character attributes on the data going to the host in the same way it processes data going to the terminal; the data streams produced are identical.

If you code EAB in the DSM operand of the STATION macro, the system allocates the extended attribute buffer at startup.

The APL On/Off key is defined in the translation table for case 3 of both the EBCDIC table and the APL table as (X'01',X'C1'), which causes DCA to change to the APL input translation table. This key is not passed to the application program.

The other attribute keys are identified as a combination of a control bit for DCA and EOM/EID for DATSM, as shown below:

EID/EOM Definition

CASE 1:

Scan code	EOM/EID	Attribute
X'2D'	x'11'	Blink highlighting
X'1D'	x'12'	Reverse highlighting
X'3D'	x'13'	Underlined highlighting
X'42'	x'21'	Blue
X'50'	x'22'	Red
X'60'	x'24'	Green
X'52'	x'27'	White

CASE 2:

Scan code	EOM/EID	Attribute
X'50'	x'23'	Magenta
X'60'	x'26'	Yellow
X'42'	x'25'	Turquoise
X'0F'	x'10'	Cancel highlighting
X'52'	x'20'	Cancel color

Communication Programming Considerations

This section is intended for users who normally write the controller application programs in 4700 assembler language. It presents several important considerations for writing FCL programs to control communication to and from a 4700 system that uses DATSM.

When using DATSM, you might want to receive the 3270 data stream using the same SNA protocol as supported by the 3274 or 3276 control units (SLU-2). If so, you must write an application program that communicates in the same way as a 3274 or 3276. This operation is described in the SNA-SDLC communications section of the *3270 Information Display System Component Description*, GA27-2749.

The complexity of the program depends on the extent to which the 3274 or 3276 is to be emulated. Basically, however, the main considerations are the SNA protocols, since the 4700 controller handles the SDLC line protocols.

The 3274 or 3276 controls I/O devices without application programming. The 3274/76 responses are designed to cover all error conditions and possible misuse of the device as well as the normal protocol sequences. When setting up the communications discipline, your major concern is processing the exception and error conditions.

If the network with which you must deal is already established, you can normally assume that the SNA error logic in the attached devices is not heavily used. You should, therefore, write your application program with very little error recovery or BIND recognition capability. Your program can ignore small inconsistencies, but can record the more significant ones and end the session when they occur. If the network has frequent problems, you can write additional error recovery within the SNA session (defined by the SNA architecture).

Because the 3274 or 3276 controls I/O devices, the state of an attached device is reflected in the SNA protocol. If, for example, a 3278 display is not connected and the 3274 or 3276 control unit receives a BIND request for the SLU associated with the head, the control unit generates a negative response (X'08'). The 4700 SLU, however, has no device restraints and therefore issues no negative response. This example illustrates that many of the states within a 3274 or 3276 related to specific hardware are not necessarily applicable to a particular 4700 application. Further examples of the differences you can choose to exercise are:

- The 3274 or 3276 must respond to a switched-off 3278 display by issuing an error response, but your program can process the transaction either by switching the message to another 4700 station or by making an entry in the log about the transaction.
- If you want your program to emulate the 3274 or 3276 attachment by using a 4704 assigned to a particular station (4700 SLU), then the powered-off state is of concern, and a negative response must be generated.

Functions of the 4700 Controller

Because the 4700 controller, like the 3274 or 3276 control units, uses SNA protocol, the 4700 controller processes some of the requirements of SNA protocol so that your program need not provide the code to handle them. For example, the 4700 controller controls the SSCP-PU and SCP-LU sessions, including the processing of the ACTPU, DACTPU, ACTLU, and DACTLU session control commands. The controller also enforces valid command sequences for these session flows.

When the SSCP-LU session is established, the application program receives a ready indication. When the SSCP-LU session terminates, the application program receives an indication of a loss of control. You can use the activate and deactivate logical unit commands (ACTLU and DACTLU) as specified in the SSCP-LU flow. If you transmit the activate logical unit command, your application can receive a procedure error command on the same flow.

The 4700 controller also takes care of the following functions:

- Enforces the traffic (Reset - Active) state and the quiescent state for messages sent by the application in the 4700. Although you could add code to the application program to enforce these states on messages the program receives, the responsibility for enforcement belongs to the sender.
- Presents only valid commands on the LU-LU session and the procedure error command in the SSCP-LU session to the application program. The transmission header (TH), request header (RH), and the command portion of the RU have been verified by the controller so that you can assume them to be correct.

- Can send and receive all the commands that the 3274 or 3276 can send or receive.

- Verifies the sequence numbers on sequential message flow in the LU-LU session.

- Controls both inbound and outbound pacing (see 4700 documentation for the description of this support).

- Supports outbound segmenting but not inbound segmenting. If your FCL program is to perform 3274 or 3276 emulation, the BIND command's parameters must specify an RU size equal to or smaller than 256 bytes, and the program should transmit RUs only within that size range to the host application program.

- Detects the error conditions resulting in path error sense code X'80xx' as does the 3274 or 3276. Your FCL program need not concern itself with this sense code for negative responses.

- Handles the following additional error conditions:

  - X'0201' - Sequence number error

  - X'0202' - Chaining error

  - X'1003' - Function not supported (for invalid commands only)

  - X'0805' - Session limit exceeded

Responsibilities of the Application Program

Your program must perform the following functions:

- Bind parameter checking with the exception of inbound pacing parameters.

- Processing of all commands and data received on the LU-LU session flow.

- Sending any necessary commands and data on the LU-LU session flow.

- Managing the bracket state, including the protocol and its states. The size of this function depends on the amount of error checking required by the messages received.

- Controlling the change direction indicator.

Finally, you must consider the host application program and the amount of support it provides for a 3274 or 3276. The amount of support you must provide in the application program increases with the support provided in the host program.

Chapter 5. 4700 Loop and DCA Assembler Instructions

This chapter describes the 4700 assembler instructions you must use to read from, write to, and control the various 4700 terminals, devices, and ports. These descriptions often refer to DEVnnnn and DCAnnnn configuration macros described in Volume 6 of this library.

The first section of this chapter describes the normal 4700 assembler instructions; the second part describes the 3270 Data Stream Compatibility (DATSM) assembler-level instructions. For coding and syntax rules, refer to the *4700 Controller Programming Library, Volume 1.*

4700 Assembler Instructions

The instructions that are described on the pages immediately following are used in both DATSM and normal 4700 terminal/device I/O programming. The instructions described in this chapter are:

ASSIGN Assign device/components.

DCACTL Control the device cluster adapter (DCA) ports and devices.

DEVPARM Control device/component operating parameters.

DPOOL Control device/component pool assignment.

GETDMS Move the DCA field control table (FCT).

GETFLD Get DCA image buffer contents.

LCHECK Test terminal/component write status.

LREAD Read data/status from a device/component.

LWRITE Write (send) data to a device/component.

PUTDMS Change the DCA field control table (FCT).

PUTFLD Lay out ("map") data into DCA image buffer.

SIGNAL Switch on/off device indicators or PIN keypad.

STPLPS Stop a device/component loop ("B–loop").

STRLPS Start a device/component loop ("B–loop").

SWAPTT Exchange ("swap") translation tables.

ASSIGN -- Terminal Component Assignment

ASSIGN assigns a terminal component to either an available logical device address (LDA) in a specified station, or to a "free pool" of unassigned devices. Unassigned devices are those defined by configuration (CPGEN) but not in any work station or device pool.

ASSIGN changes only the station ID and LDA assignments. If the component is shared, one side (either A or B) must be specified.

The physical device address, the shared device indicator, the station ID, and LDA for the component to be assigned are specified in a parameter list; the ASSIGN instruction points to this list. Refer to Figure 5-1, which shows the ASSIGN parameter list for LDA assignment for loop and DCA devices.

If the assignment is completed successfully, ASSIGN changes the station ID and LDA fields in the parameter list to show the previous assignment of the component. To assign the component to its previous state (either the station or free pool), issue a second ASSIGN instruction using the same parameter list (the list updated by the first ASSIGN instruction).

Name	Operation	Operand
[label]	ASSIGN	{ defld2 / seg2,disp2 / (reg2) / (defrf2) }

operand 2
: Defines the start of the parameter list. A DEFRF instruction label must always be in parentheses. The length of this operand is ignored and the first 5 bytes are assumed to be the parameter list. The parameter list is illustrated in Figure 5-1 on page 5-4. The segment number cannot be 14.

Loop Device:

Loop	Terminal	Com-ponent	Modulus Value	Shared Indicator	Station ID	LDA
Byte: 0		1		2	3	4

DCA Device:

X'A'	Port	Com-ponent	X'0'	Shared Indicator	Station ID	LDA
Byte: 0		1		2	3	4

Loop

The four-bit binary loop number, or set to X'A' for DCA.

Terminal

This four-bit binary value in the terminal's address switches.

Port

This is the four-bit binary address *X'F'* '15' of the DCA port to which the component (3262, 3278/3279, 3287, or 4704-2/3) is connected.

Component

Is the 4-bit terminal component address, as described in Chapter 1, "General Terminal and Device Programming."

Modulus Value

Is the four-bit base modulus value returned only if X'FFFF' was specified for the device address. For DCA, set this field to X'0'.

Shared Indicator

Is the shared device indicator (C'A' or C'B'); if the device is not shared, this byte must contain a C'A'.

Station ID

This eight-bit field contains the station ID.

LDA

Is the 8-bit binary number specifying the logical device address (LDA).

Figure 5-1. Device Parameter Lists Used by ASSIGN

Condition Codes: One of the following is set:

Hex Code:	Explanation:
01	The assignment was successful.
02	Unsuccessful assignment: The LDA specified already has a device assigned to it, or—for an earlier release having no LEXIT—the LDA has no device assigned to it.
04	The device was busy (it was being used by another station or the controller requested an early release with the device still busy); or your program attempted to assign a 3614/3624 with its front panel open.
08	The parameter list was invalid (station ID, A/B field, or device specification). The station ID was specified as 0 or X'FF'.

Program Checks: 1, 2, or 27 can be set.

Programming Notes

- Specifying a receiving station ID of 0 assigns the terminal component to the free pool.
- If bytes 0 and 1 of the parameter list contain X'FFFF', ASSIGN returns the parameter list with the first two bytes set as described in Figure 5-1, and returns the shared device indicator of the LDA for the specified station. If no device was assigned to the specified LDA, the parameter list is not changed.
- If bytes 0 and 1 of the parameter list contain X'0000', ASSIGN permits the program's station to give up the specified terminal. A station using ASSIGN this way can release the terminal without issuing an LEXIT instruction. Early release will not take place, however, unless you first issue LCHECK to test both for any data transmission instructions that may be pending for the terminal, and to ensure that the device is available.

DCACTL -- DCA Control

The DCACTL instruction controls the DCA adapter and the associated adapter ports. DCACTL uses the following three-byte parameter list:

Byte 1:	Contains the request code.
Bytes 2-3:	Contain data required by the instruction.

The contents of the parameter list are as follows:

Function:	**Request Code (Byte 1):**	**Port Number (Byte 2):**	**Reserved (Byte 3):**
Start DCA Adapter	00	00	00
Stop DCA Adapter	01	00	00
Enable Port	02	00-07	00
Disable Port	03	00-07	00

Name	Operation	Operand
[label]	DCACTL	{ defcon2 | defld2 | seg2,disp2 | (reg2) | (defrf2) }

operand 2
Defines the parameter list. The length must be 3.

Condition Codes: The following may be set:

Hex Code:	**Explanation:**
01	The instruction executed successfully.
02	Status is stored.
04	The device was busy. DCACTL was not executed.

Program Checks: 1, 2, or 27 can be set.

DEVPARM -- Change Component Parameters

DEVPARM sets or returns either the device operating parameters and characteristics in a nine-byte parameter list, or the component's identifier, assignment status, and user-defined data in an eight-byte parameter list. Operand 2 defines the parameter list location in either case.

When setting or returning operating parameters, DEVPARM dynamically changes the characteristics of the 3616, 4710, and 4720; you must issue DEVPARM to set any needed document print station (DPS) parameters before actual printing begins. When DEVPARM specifies SETRET=SET for the 3610, 3611, 3612, or 3615, the component's registers are reset to line 1, column 1. For a 3610, 3611, or 3612, DEVPARM opens the platen and stops the motors.

If you specify SETRET=SET, the new characteristics are contained in a 9-byte list referred to by DEVPARM. All parameters for the component, whether changed or not, must be specified in the list. If the parameter list contains fewer parameters than required for the component, none of the parameters are changed and DEVPARM sets a condition code of X'02' and returns a status code indicating command reject in SMSDST. (See Figure 5-2 on page 5-12 for the format of the list and the parameters required for each component.)

For the SETRET=RETURN option, DEVPARM returns the terminal operating parameters to the parameter list in the same format as required for SET, permitting the characteristics to be restored easily after being changed.

Name	Operation	Operand

```
                               ( defld2            )
                               | (defrf2)          |  [   ( WAIT   ) ]
[label] DEVPARM  {lda  }       { seg2,disp2,len2   }  [ , {        } ]
                 {mnem }       | (reg2)            |  [   ( NOWAIT ) ]
                               | seg2              |
                               ( (defcon2)         )

                 [ (                               ( Y )   )  ]
                 [ (          { RETURN [ ,EXP= {     } ] } )  ]
                 [ ( SETRET=  {                ( N )   }   )  ]
                 [ {          { SET                        } }  ]
                 [ (                                       )  ]
                 [ (            ( RETURN )                 )  ]
                 [ ( EXTRACT= {          }                 )  ]
                 [ (            ( SET    )                 )  ]
```

lda
: Is a decimal number (0-7) indicating the logical device address assigned to the terminal.

mnemonic
: Is the code of the component referred to in the instruction. Each mnemonic sets the LDA used as a default during controller configuration. Refer to Chapter 1, "General Terminal and Device Programming" on page 1-1 for the mnemonics and their LDAs.

operand 2

Defines the start of the parameter list. The label of a DEFRF instruction must always be in parentheses. The length associated with this operand is ignored, and the first 1 to 9 bytes (depending on the device type) are assumed to be in the parameter list. The parameter list is defined in Figure 5-2 on page 5-12 . When you specify the RETURN option for either SETRET= or EXTRACT=, do not specify the label of a DEFCON statement.

WAIT

Specifies that the application program waits until the DEVPARM instruction is completed and status is stored before execution continues with the next sequential instruction.

NOWAIT

Specifies that the application program continues execution after issuing a DEVPARM instruction without waiting for the DEVPARM instruction to be completed.

SETRET

Sets or returns the operating characteristics of the selected device. SETRET=RETURN sets the current characteristics of the terminal to the specified parameter list area in the same format required for the SET option (see Figure 5-2 on page 5-12). The operand is invalid for the 3616.

SETRET=SET is the default if you specify no DEVPARM operand except the device or component. SETRET=SET causes the new characteristics stored in the parameter list to become the new set of characteristics for the specified terminal. SET is the default option.

EXP

Specifies whether or not expanded display status is returned. If you code EXP=Y with SETRET=RETURN, the display returns the number of characters per line, lines per screen, the cursor line position, and the cursor column position in bytes 2 through 5 (see Figure 5-2 on page 5-12, part 1). Specifying EXP=N (the default) causes only the indication "8" for the 3278 or 3279 to be returned in byte 1. Do not specify EXP=Y with SETRET=SET or with the EXTRACT= parameter, or an assembly error occurs.

EXTRACT

When EXTRACT specifies SET, DEVPARM sets an eight-byte device/component definition parameter list selected by Operand 2. Specifying RETURN cause DEVPARM to read device/component status into the parameter list. Do not specify EXTRACT with the SETRET EXP option.

Condition Codes: One of the following is set:

Hex Code:	**Explanation:**
01	The component parameters were changed.
02	Status returned; the status code is contained in SMSDST. Refer to Appendix D, "Terminal and Device Status Codes" for an explanation of the status code.

The following are invalid characteristic specifications that set a condition code of X'02' and return a status code of 0480 in SMSDST:

Device:	**Invalid Specification:**
3610, 4710, or 3612 Printer	1. Flag byte indicating forms type is not X'01' or hex 02. 2. Page size or warning line is negative (a value greater than 127 produces a negative number). 3. Flag byte is hex 02, but the printer uses only cut-forms mode.
3612 Passbook Printer	1. Page size is 0. 2. Page size, center fold begin, center fold skip, step offset, or line offset is negative (a value greater than 127 produces a negative number). 3. The sum of the page size, center fold skip, and line offset is greater than 42.
3615 Administrative Terminal Printer	1. Page size is negative (a value greater than 127 produces a negative number).
3616 Journal Printer	1. Flag byte not X'02' or X'82'. 2. Warning line greater than page size. 3. Line length greater than 57 at 12 characters per inch, or 47 at 10 characters per inch. 4. SETRET=RETURN
3616 Passbook/Document Printer	1. Flag byte not X'00' or hex 01. 2. Auto-start and shared both specified. 3. Line length greater than 100 at 12 characters per inch, or 83 at 10 characters per inch (**Note:** *For vertical-fold passbooks, skip is included in line length*). 4. Page size greater than 72 at 6 lines per inch; or 60 at 5 lines per inch. 5. SETRET=RETURN 6. Page size is 0.
4710 Printer	1. Flag byte not X'01', X'02', X'20', or X'82'. 2. Flag byte not X'01' or X'20', and "journal not present" switch is set. 3. Both "autostart" and "shared" are specified. 4. Line length greater than allowed.

Program Checks: 1 or 2 can be set.

	4704/3604/ 3278/3279 Keyboard	4704/3604/ 3278/3279 Display	3608 Printer	3610 or 3612 Document Printer	3611 or 3612 Passbook Printer	3615 Administrative Terminal Printer
Flag Byte	EOM-set selection mask (if set to X'00', mask is not altered)	—reserved—	First line print types: X'00'—do not change line one print type X'80'—first line is ten pitch X'C0'—first line is seven pitch	Forms types: X'01'—cut-forms mode X'02'—continuous-forms mode (restricts concurrent sharing) X'41'—cut-forms mode (specifies automatic start) X'82'—continuous-forms mode (specifies concurrent sharing)	—reserved—	Forms types: X'01'—cut-forms mode X'02'—continuous-forms mode (no concurrent sharing) X'41'—cut-forms mode (specifies automatic start) X'82'—continuous-forms mode (concurrent sharing)
Data Byte 1	—reserved—	model number*	Maximum form width (0.1 inch)	Page size in lines	Page size in lines	Page size in lines
Data Byte 2	—reserved—	—reserved*	Initial offset (0.1 inch)	Warning line number	Center fold begin line number	Warning line number
Data Byte 3	—reserved—	—reserved*	Page spacing (0.1 inch)	—reserved—	Center fold skip in number of lines	—reserved—
Data Byte 4	—reserved—	—reserved*	—reserved—	—reserved—	Step offset in number of steps	—reserved—
Data Byte 5	—reserved—	—reserved*	—reserved—	—reserved—	Line offset in number of lines	—reserved—

*The number 8 should be set, or will be returned, for a 3278 Model 2. If EXP=Y, DEVPARM with SETRET=RETURN presents characters per line for any display in byte 2, lines per screen in byte 3, cursor line position in byte 4, and cursor column position in byte 5.

	3262/3287 DCA Printers		
Flag Byte	X'02' Continuous-forms mode	X'08' Activate time-out. Timer set during I/O.	X'09' Deactivate time-out. Timer not set during I/O.
Data Byte 1	Device Characteristics Bit 0 (unused) Bit 1 = 0: no data chaining* Bit 1 = 1: data chaining* Bits 2–6 = unused Bit 7 = 0: no FM header follows in data stream Bit 7 = 1: FM header for structured fields follows**	(reserved)	(reserved)

Notes:

*3262 data chaining is necessary to achieve rated printing speed. Refer to "Chapter 3, Programming for Printers," to determine how data should be used. Data chaining is not necessary on the 3287 printer.

**If bit 7 is 1, the data stream that follows this DEVPARM is an FM header defining the data format and length. Refer to the appropriate printer's Component Description manual.

Figure 5-2 (Part 1 of 4). DEVPARM/SETRET Parameter List and Values

	3616, 4710, and 4720 Journal Printers	3616 and 4720 Document Printers	3616, 4710, and 4720 Printers	Any 3616 Printer	Any 3616, 4710, or 4720 Printers
Flag Byte	X'02' continuous forms mode/no concurrent sharing X'82' continuous forms mode/allow concurrent sharing	X'00' Passbook Mode	X'01' Cutforms Mode	X'08' activate timeout—timer is set during I/O X'09' deactivate timeout—timer is not set during I/O	X'20' Table load — load special character table (National Use Graphics or user-defined)
Data Byte 1	Page size in lines (see device chapter)	Page size in lines (see device chapter)	Page size in lines (see device chapter)	reserved	Table ID — see note 1
Data Byte 2	Warning line number (see device chapter)	Centerfold begin column/line number (see device chapter)	Warning line number (see device chapter)	reserved	reserved
Data Byte 3	Line length (see device chapter)	Centerfold skip in number of lines/char's (see device chapter)	Step offset in number of steps (4710: reserved) – see device chapter	reserved	reserved
Data Byte 4 (Journal control bits 0-3 & 5 are unused by 4720; bits 0-6 are unused by 4710)	Device characteristics: Bits 0-1: unused Bit 2 = 0: nonshared = 1: shared Bit 3 = 0: start key required = 1: autostart Bits 4-5: unused **(except 4720)** Bit 4 **(4720 only:)** = 1: extension in byte 5 = 0: no extension Bit 6 **(4720 only)**: = 0: 5 lpi = 1: 6 lines per inch Bit 7 = 0: 10 cpi = 1: 12 cpi	Step offset in number of steps. This value must be less than the total steps per line (<12 for 5 lpi, <10 for 6 lpi). See byte 8, bit 06 **(4720 only)** Specify up to 255 steps	Line offset in number of lines	reserved	reserved
Data Byte 5	4720 characteristics Bits 0-3: reserved set to zeros Bits 4-5: print pitch: 00 = use bit 7 of byte 4 01 = 10 cpi 10 = 12 cpi 11 = 16-2/3 cpi Bits 6-7: reserved set to zero	Line offset in number of lines (must be non-zero) (see device chapter)	Line length (see device chapter)	reserved	reserved

Figure 5-2 (Part 2 of 4). DEVPARM/SETRET Parameter List and Values

	3616, 4710, and 4720 Journal Printers	3616 and 4720 Document Printers	3616, 4710, and 4720 Printers	Any 3616 Printer	Any 3616, 4710, or 4720 Printers
Data Bye 6 (cut form control bits 1 and 6 unused by 4710)	reserved	Left margin column number (see device chapter)	Device Characteristics: Bit 0: unused Bit 1 = 0: no data chaining (see note 4) = 1: data chaining Bit 2 = 0: non-shared = 1: shared Bit 3 = 0: start key required = 1: auto start **Bit 4 (4720 only):** = 1: extension in byte 7 = 0: no extension Bit 5 = 0: no auto new line = 1: auto new line Bit 6 = 0: 5 cpi = 1: 6 cpi Bit 7 = 0: 10 lpi = 1: 12 lpi	reserved	reserved
Data Byte 7	reserved	Line length (see device chapter)	4720 extended definition: Bit 0 -Print font: =0: regular font =1: quality font — see note 5 Bit 1: -Model 2/4 Journal control: =1: advance journal =0: no advance Bits 2-3: unused Bits 4-5: print pitch: 00 = use byte 6, bit 7 01 = 10 cpi 10 = 12 cpi 11 = 16-2/3 cpi Bits 6-7 - Forms skew: 00 = 1.37mm/100mm (.05 in./4 in.) 01 = 2.05mm/100mm (0.08 in./4 in.) 10 = 2.74mm/100mm (.11 in./4 in.) 11 = 3.42mm/100mm (.13 in./4 in.)	reserved	reserved
Data Byte 8	reserved (for 4720, see cut forms mode bit 1)	Device characteristics: Bit 0 = 0: horizontal fold =1: vertical fold Bit 1: unused Bit 2 = 1: shared =0: non-shared Bit 3 =0: start key req. =1: autostart Bit 4 **(4720 only):** =0: no extension =1: extension in byte 9 Bit 5 = 0: no auto new line =1: auto new line Bit 6 = 0: 5 lpi =1: 6 lpi Bit 7 = 0: 10 cpi =1: 12 cpi	reserved	reserved	reserved

Figure 5-2 (Part 3 of 4). DEVPARM/SETRET Parameter List and Values

	3616, 4710, and 4720 Journal Printers	3616 and 4720 Document Printers	3616, 4710, and 4720 Printers	Any 3616 Printer	Any 3616, 4710, or 4720 Printers
Data Bye 9	reserved	4720 extended definition: Bits 0-3: reserved Bits 4-5: print pitch: 00 = see byte 8, bit 7 01 = 10 cpi 10 = 12 cpi 11 = 16-2/3 cpi Bits 6-7 - skew: 00 = 1.37mm/100mm (.05 in./4 in.) 01 = .68mm/100mm (.03 in./4 in.) 10,11 = 2.05 mm/100 mm (0.08 in./4 in.)	reserved	reserved	reserved

Notes:

1. Table ID is ID of either user-defined special characters table or IBM Table 5A: National Use Graphics Special Character Table:

07	User-defined (normal print)	13	Belgium	18	Italy	23	Hebrew	1A	Portugal
10	U.S.	14	Brazil	19	Japan	24	Arabic	1B	Spain
11	International	15	Denmark/Norway	20	Yugoslavia	25	Greek	1C	Spanish Speaking
12	Austria/Germany	16	Finland/Sweden	21	Turkey	47	User-defined 4720 quality print.	1D	United Kingdom
		17	France	22	South Africa			1E	Canadian French
								1F	Katakana

2. Refer to the appropriate device chapter in part 2 for characters generated for the IBM Table 5A: National Use Graphics.
3. The table must be specified, for the terminal, via the OUTBHDR, OUTSPEC, and/or CHARSET macros.
4. Combining cut-form mode with data chaining causes complete overlapping of data transmission and printing. LWRITEs are posted as complete before completion of printing. This permits subsequent LWRITE to be transmitted while the previous LWRITE is still printing. Issuing an LWRITE containing a form feed ensure completion of printing. Data chaining does not apply to the 4710 or 4720.
5. Specifying quality printing on a Katakana 4720 or with 16-2/3 cpi causes DEVPARM format errors.

Figure 5-2 (Part 4 of 4). DEVPARM/SETRET Parameter List and Values

Bytes 0 and 1

Four-digit device type, in hexadecimal. For example, a 4710 is represented in these two bytes as X'4710'. Refer to the rest of this figure for the type codes allowed and returned.

Bytes 2-4

Device model, encoded as characters (one character per byte), or ''C''. For example, a device model ''B02'' is specified as X'C2F0F2', and a device model ''12'' with a leading blank (' ') as X'40F1F2'. Refer to the rest of this figure for the model codes allowed and returned.

Byte 5

Component ID. These binary IDs are:

1 Keyboard, or KB

2 Display, or DS

3 Magnetic stripe device, or MS

4 Printer/first or only component

5 Printer/second component

6 Financial Services terminal (3606/08)

7 Document processor

8 Consumer transaction facility (3614/3624)

Byte 6

User data defined during configuration of the device's DEVnnnn or DCAnnnn macro, using the *UDD . . .* = parameter. This can be any value ranging X'00' to X'FF'.

Byte 7

Assignment status:

X'20' Device or component is sharing concurrently.

X'40' Device or component is sharing nonconcurrently with another station.

X'80' Device or component is assigned to this station's device pool.

Figure 5-3 (Part 1 of 3). DEVPARM/EXTRACT Parameter List and Values

Loop-Attached Devices			
Device:	**Model:**	**Component:**	**Description:**
3604	1	01	Keyboard (with or without magnetic stripe reader and PIN keypad)
		02	Display
		03	Mag Stripe (magnetic stripe reader/encoder)
	2	01	Keyboard (with or without magnetic stripe reader and PIN keypad)
		02	Display
		03	Mag Stripe (magnetic stripe reader/encoder)
	3	01	Keyboard (with or without magnetic stripe reader and PIN keypad)
		02	Display
		03	Mag Stripe (magnetic stripe reader/encoder)
	4	01	Keyboard (with or without magnetic stripe reader and PIN keypad)
		02	Display
		03	Mag Stripe (magnetic stripe reader/encoder)
	5	01	Keyboard (with or without magnetic stripe reader and PIN keypad)
		02	Display
	6	01	Keyboard (with or without magnetic stripe reader and PIN keypad)
		02	Display
	7	01	Keyboard (with or without magnetic stripe reader and PIN keypad)
		02	Display
3606		06	(keyboard, display, magnetic stripe reader)
3608		04	Printer
		06	(keyboard, display, magnetic stripe reader)
3609	1	01	Keyboard (with or without magnetic stripe reader and PIN keypad)
		02	Display
		03	Mag Stripe (magnetic stripe reader/encoder)
	2	01	Keyboard (with or without magnetic stripe reader and PIN keypad)
		02	Display
		03	Mag Stripe (magnetic stripe reader/encoder)
	3	01	Keyboard (with or without magnetic stripe reader and PIN keypad)
		02	Display
		03	Mag Stripe (magnetic stripe reader/encoder)
3610	1	04	Document Printer
	2	04	Document Printer
	3	04	Document Printer
	4	04	Document Printer
	5	04	Document Printer
	12	04	Document Printer
	13	04	Document Printer
3611	1	04	Passbook Printer
	2	04	Passbook printer

Figure 5-3 (Part 2 of 3). DEVPARM/EXTRACT Parameter List and Values

Loop-Attached Devices			
Device:	**Model:**	**Component:**	**Description:**
3612	1	04	Document Printer
		05	Passbook Printer
	2	04	Document Printer
		05	Passbook Printer
	3	04	Document Printer
		05	Passbook Printer
	12	04	Document Printer
		05	Passbook Printer (CPGEN: MODEL=12P)
	13	04	Document Printer
		05	Passbook Printer (CPGEN: MODEL=13P)
3614		08	Consumer Transaction Facility
3615	1	04	Administrative Printer
	2	04	Administrative Printer
3616		04	Journal Print Station
		05	Document Print Station
3624		08	Consumer Transaction Facility
4704	1	01	Keyboard (with or without magnetic stripe reader and PIN keypad) (CPGEN: MODEL=11 or 12)
		02	Display (CPGEN: MODEL=11 or 12)
		03	Mag Stripe (magnetic stripe reader/encoder) (CPGEN: MODEL=11 or 12)
4710		04	Receipt/Validation Printer
4720	1	04	Cut form Printer
	2	04	Cut form Printer with Journal
	3	04	Cut form/Passbook Printer
	4	04	Cut form/Passbook Printer with Journal

Figure 5-3 (Part 3 of 3). DEVPARM/EXTRACT Parameter List and Values

DPOOL -- Assign Device Pool Terminals and Components

DPOOL assigns components back and forth between the work station and its device pool. DPOOL performs basically the same operations as the ASSIGN instruction; however, instead of specifying the physical device address, you specify a component type; DPOOL then searches for a component of that type.

DPOOL transfer components between the station and the device pool according to the operands you specify and the 19–byte parameter list selected by Operand 2. You can define a DPOOL parameter list with the format shown in Appendix B, "4700 Fields and Parameter Lists" using the COPY DEFPOL instruction. The device pool itself is created during configuration by the DPOOL operand of the STATION macro, and by the DEFADDR/DCAPORT macros.

DPOOL performs the following operations:

- Assigns a component to the work station (GET and GETX).
- Reassigns a component to the device pool (RETURN).
- Allocates or deallocates a device for assignment eligibility (ACT and DEACT).
- Requests information about a component in the device pool (QUERY).

Name	Operation	Operand
[label]	DPOOL	{GET | GETX | RETURN | QUERY | ACT | DEACT} , {defld2 | seg2,disp2 | (reg2) | (defrf2)}

GET and GETX

These operands make DPOOL search the station's device pool for a component with the component identifier defined by the parameter list, and assign the component to the work station. GET/GETX use only the first seven component identifier bytes in the search. GET can be used to assign components of a multi-component device to different work stations. GETX gives the work station exclusive use of the component or device. GET and GETX do not assign devices that were deactivated by DPOOL DEACT.

DPOOL performs a priority search that depends on the type of assignment you request. DPOOL GET searches first for an available component of a terminal with other components already assigned to any work station, and assigns that component to the work station. DPOOL GETX searches first for a component of a terminal with other components already assigned to this same work station.

If neither DPOOL GET or GETX find their preferred components during the first search, they search the device pool once again for the first available requested unassigned component, and assign it to the work station.

Before issuing DPOOL GET/GETX, you must set the parameter list as follows:

Byte 0
Set to X'00'.

Byte 1
Set to the logical device address (LDA) to be assigned to the component.

Bytes 2 – 9
The identifier of the component being searched for in the device pool. Refer to Figure 5-3 on page 5-16 for a description of the component ID.

The rest of the parameter list should be zero. After DPOOL GET/GETX executes, the parameter list contains the following information about the requested component:

Bytes 0 – 9
Unchanged

Bytes 10 and 11
The physical identifier for the component, comprising the component's physical address and modulus (refer to Figure 5-1 on page 5-4).

Byte 12
The DPOOL status code. If DPOOL returns a condition code of X'02', the execution was unsuccessful; the program should test this byte for the cause:

Status Code:	Cause:
X'02'	The specified LDA is already assigned.
X'10'	The specified component is unavailable for exclusive assignment with GETX.
X'30'	Invalid parameter list (byte 0 was not zero, LDA invalid, and so on).
X'40'	The specified component is already assigned.
X'50'	The component is either not in the station's device pool, or was deactivated.
X'60'	The station has no device pool.

The last six parameter list bytes have no meaning.

RETURN

This operand makes DPOOL return a component (specified by the LDA in byte 1 of the parameter list) to the station's device pool. You must first set the parameter list as follows:

Byte 0

Set to X'00'.

Byte 1

Set to the logical device address (LDA) of the component to be reassigned to the device pool.

After DPOOL RETURN executes, the parameter list contains the following:

Bytes 0 – 11

Unchanged.

Byte 12

The DPOOL status code. If DPOOL returns a condition code of X'02', the execution was unsuccessful; the program should test this byte for the cause:

Status Code:	Cause:
X'04'	The selected device was busy, or its front panel was open.
X'20'	The specified LDA has no assigned component.
X'30'	Invalid parameter list (byte 0 was not zero, LDA invalid, and so on).
X'50'	The component is either not in the station's device pool, or was deactivated.
X'60'	The station has no device pool.

The last six parameter list bytes must be zero.

ACT and DEACT

These operands make a device in the issuing station's own pool eligible (ACT) or ineligible (DEACT) for assignment with GET or GETX.

ACT and DEACT search the device pool for a device with a physical address matching the one you specify in the DPOOL parameter list. If DPOOL finds a matching address, it turns on or off the component deactivated flag in component identifier byte 8. DPOOL returns the deactivated device to the station's device pool if the device was assigned to the station.

ACT and DEACT have the following DPOOL parameter list format:

Byte 0

Set to X'00'.

Bytes 1–9

These reserved bytes must be zero.

Bytes 10 and 11

The physical device address for the desired component. This address comprises four 4-bit fields: loop, slot, address, and modulus. When searching for the component, the modulus is unused.

The last six parameter list bytes must be zero.

After DPOOL ACT or DEACT completes, the parameter list contains the following:

Bytes 0–11

Unchanged.

Byte 12

The DPOOL status code. If DPOOL returns a condition code of X'02', the operation failed. Your program should test this byte for the cause:

Status Code:	Cause:
X'04'	The selected device was busy, or its front panel was open (DEACT request).
X'30'	Invalid parameter list (byte 0 was not zero, LDA invalid, and so on).
X'40'	The specified component is already assigned.
X'50'	The component is not in the station's device pool.
X'60'	The station has no device pool.

QUERY

DPOOL QUERY provides information about components in the station's device pool. To execute DPOOL QUERY, you must first set the parameter list as follows:

Byte 0
Set to X'00'.

Byte 1
Set to the station ID of the device pool being queried. Specify X'00' for this station.

Bytes 2 – 9
Reserved—set these bytes to zero.

Bytes 10 and 11
Set to the physical identifier of the last device queried. Specifying X'0000' selects the first device in the selected device pool.

After DPOOL QUERY executes, the parameter list contains the following information about the selected device:

Bytes 0 and 1
Unchanged

Bytes 2 – 9
Contain the component identifier for the queried device. This identifier is assigned to the device during configuration.

Bytes 10 and 11
Contain the physical identifier for the queried device. If this value is X'FFFF', the DPOOL QUERY operation searched to the end of the pool.

Byte 12
The DPOOL status code. If DPOOL returns a condition code of X'02', the execution was unsuccessful; the program should test this byte for the cause:

Status Code:	**Cause:**
X'30'	Invalid parameter list (byte 0 was not zero, LDA invalid, and so on).
X'60'	The station has no device pool.

Byte 13
Set to the station ID where the component is currently assigned. If this value is X'00', the component is not assigned.

Byte 14
The LDA of the component if it is assigned to a station.

Byte 15

The number of components in the queried component's terminal.

Byte 16

The use indicator, which has the following meanings:

X'00' All components of the queried component's terminal are available.

X'FF' One or more of the terminal's components are assigned. The assigned components may be shared between stations.

X'nn' One or more components of the terminal are assigned for the exclusive use of station X'nn'.

The last two parameter list bytes are reserved, and have no meaning.

Operand 2

Specifies the address of the DPOOL parameter list. The length of the list is assumed to be 19 bytes. The parameter list cannot be in segment 14.

Condition Codes: One of the following is set:

Hex Code:	Explanation:
01	DPOOL executed successfully.
02	DPOOL completed unsuccessfully. The DPOOL status code in byte 12 of the returned parameter list defines the reason for the failure. Refer to each DPOOL function description, above, for the appropriate DPOOL status codes.

Program Checks: 1, 2, or 27 can be set by DPOOL.

Programming Notes

1. A DPOOL instruction issued to a device assigned during configuration or an ASSIGN instruction issued to a device in the device pool will not execute.

2. For DPOOL to operate, the OPTMOD macro must specify "P42".

LCHECK -- Check the Status of a Terminal Component

LCHECK determines the status of a terminal component attached to the controller and synchronizes data transmission between the terminal and application program.

Before LCHECK operation completes, it stores the status in SMSDST and sets the condition code accordingly. If there is no outstanding write operation, LCHECK returns a zero status code and a condition code of X'01'.

IF LCHECK did not specify TIO and a write operation to the terminal is still not completed when LCHECK is issued, LCHECK places this work station in wait state until the write operation ends. If TIO was specified, LCHECK sets the condition code to indicate whether or not the write operation was successful. LCHECK also resets any intervention required indication.

Name	Operation	Operand
`[label]`	`LCHECK`	`lda [,TIO ]`

lda

Is the logical device address (LDA) assigned to the terminal during the controller configuration procedure. The default LDA values are described at the beginning of this chapter.

TIO

indicates that a test I/O operation is to be performed. The application program retains control whether the I/O operation being checked has completed or not.

Condition Codes: One of the following is set:

Hex Code:	Explanation:
01	I/O has completed, zero status is returned.
02	I/O has completed, nonzero status is returned; the status code is contained in SMSDST. (See Appendix D, "Terminal and Device Status Codes" for an explanation of the status codes.)
04	I/O has not completed (applies only for TIO option).

Program Checks: None are set.

LREAD -- Read from Terminals

LREAD reads data from input terminals. The data read from the keyboard/displays can come from either the keyboard, PIN keypad, or the magnetic stripe reader although only the keyboard is specified in the LREAD instruction.

LREAD reads data into the specified segment beginning at the fixed field address or the primary field pointer (PFP), and continuing for the length of the fixed field or field length indicator (FLI) or until one of the other ending conditions listed below occurs:

1. The controller detects a CPGEN-defined EOM character (keyboard/display, 3606, 3608) or receives the last message byte (3606, 3608, 3614/24).

2. The read operation reaches the end of the data field.

3. The read operation passes the end of the segment, and the FLI is 0.

4. The input buffer is full and ERTLS=Y was specified on the terminal's DEV/DCAnnnn macro (no status is returned).

5. The operator signals attention.

Data read from a 3614 or 3624 goes directly to the station's storage. This EBCDIC data requires no translation. If the read stops before all data has been read, LREAD sets status in SMSDST and the unread data is lost.

In the case of the 3606, 3608, 3614, or 3624, the read completes with error status if the device or slot group (depending on the setting of SMSTGU) has not presented an attention. At the end of the operation, the PFP is unchanged, and the controller stores the message length in SMSIML.

Name	Operation	Operand
[label]	LREAD	lda, { seg2 | defld2 | seg2,disp2,len2 | (reg2) | (defrf2) } [{ NOTRACK | TRACK }] [{ NOWAIT | WAIT }]

lda

Is the logical device address (0-7) for the input device. If LDAs were assigned by default during the configuration procedure, the LDA values are as described in Chapter 1, "General Terminal and Device Programming."

operand 2

Is the location of the data that is read. The segment number cannot be 14.

TRACK

Indicates that data read from the keyboard will be displayed immediately, or tracked. When you specify KB, tracking is assumed unless you also specify NOTRACK. If you specify a mnemonic other than KB, no tracking is assumed unless you also specify TRACK. Do not specify TRACK when reading from the 3606, 3608, 3614/24, or from the 4704-2/3 when it operates in local tracking mode.

NOTRACK

Indicates that data read from the keyboard is not to be displayed. Do not specify this operand when reading from a 3606, 3608, 3614/3624, or from a 4704-2/3 while it is operating in local tracking mode.

NOWAIT

Specifies that the application program continues processing after issuing an LREAD instruction without waiting for the LREAD to be completed. If you specify NOWAIT, you should follow the LREAD with an LCHECK instruction before performing another I/O operation on the device.

WAIT

Specifies that the application program waits until the LREAD is completed and status is stored before processing continues with the next sequential instruction.

Condition Codes: One of the following is set:

Hex Code:	**Explanation:**
01	The instruction was executed successfully.
02	Status is returned; the status code is contained in SMSDST. (See Appendix D, "Terminal and Device Status Codes" for an explanation of the status codes.)

Program Checks: 1 or 2 can be set.

Programming Notes

1. You must set SMSTGU before issuing LREAD if reading is from specific 3606 or 3608 keyboards; however, you can set SMSTGU to zero if your program accepts data from any 3606 or 3608 using that loop slot.

2. When 3606 and 3608 terminals share a loop address and your program performs consecutive LREAD operations from those devices without performing an intervening LWRITE, your program may have to save the SMSTGU value resulting from the first LREAD for any subsequent LWRITE to the first device, or that SMSTGU value could be lost.

3. An LREAD to a 4704-2/3 in local tracking mode completes when the operator presses an EID–generating key. No data returns to the controller, but the SMS holds the EID and the current cursor address for use by DATSM. Refer to "Comparison of Controller and Local Tracking Modes" on page 4-28 for specific DATSM local tracking information.

Figure 5-4 is an example of an instruction sequence for reading from a keyboard.

```
TELLINPT       DEFLD       INPUTSEG,0,100        1
ATDENTRY       WRTI        DS,X'0C'              2
               SETFPL      TELLINPT              3
               LREAD       KB,INPUTSEG           4
               BRAN        ST,KBERR              5
```

1 Define the input area.
2 Clear the display and set the cursor to line 1, position 1.
3 Set the PFP of INPUTSEG.
4 Read up to 100 bytes and track it on the display.
5 Check for status.

Figure 5-4. Reading from a Keyboard

LWRITE -- Write to Terminals

LWRITE sends data to the display or output components attached to the controller. The data is written from Operand 1.

Operand 1 can specify either a segment (variable field) or fixed field:

1. LWRITE selects the variable field with the segment's primary (PFP) and secondary (SFP) field pointers. The SFP points to the beginning of the output area and the PFP points one byte beyond the output area.

2. For fixed fields, LWRITE specifies the specific segment, length, and displacement of the data to be written.

If the length of the data to be written is defined as 0, no operation takes place. When writing to a slot-sharing device, the terminal group unit field (SMSTGU) must be set prior to issuing the LWRITE.

After your program issues LWRITE and a dispatch cycle passes, execution continues with the next sequential instruction (NSI). If LWRITE specified the WAIT option, the application program waits until the LWRITE completes and stores status before processing the NSI. When WAIT is not specified, you must either issue LCHECK or another LWRITE to the same device to obtain completion status for the write operation. The output data area should not be changed until the program receives completion status.

```
Name     Operation  Operand

                    { defcon1          }
                    { defld1           }
[label]  LWRITE     lda  { seg1,disp1,len1  }  [ { NOWAIT } ]
                    { (reg1)           }  [ { WAIT   } ]
                    { seg1             }
                    { (defrf1)         }
```

lda

Is a decimal number (0-7) indicating the logical device address (LDA) assigned to the terminal. If the LDAs were defaulted during the controller configuration procedure, they are as described in Chapter 1, "General Terminal and Device Programming" on page 1-1.

operand 1

Defines the data to be written.

When *seg1* is coded as the second operand, a 2-byte machine instruction is generated; otherwise, the machine instruction is 6 bytes long.

WAIT

Specifies that the application program waits until the LWRITE is completed and status is stored before processing continues with the next sequential instruction.

NOWAIT

Specifies that the application program will continue processing after an LWRITE is issued without waiting for the LWRITE to be completed. The status of the completion of the operation is obtained when an LCHECK instruction or another LWRITE to the same terminal is executed.

Condition Codes: One of the following is set:

Hex Code:	Explanation:
01	The write operation was successful.
02	Status is returned; the status code is contained in SMSDST. (See Appendix D, "Terminal and Device Status Codes" for an explanation of the status codes.

Program Checks: 1, 2, or 27 can be set.

Programming Notes

1. Refer to the appropriate device chapter for an explanation of terminal control characters that can be embedded in the data stream.

2. If status in SMSDST contains bit 03 (prior operation), the status pertains to a previous failing operation for that device, and not to the current LWRITE. Refer to "Synchronizing I/O Operations" on page 1-14.

3. Writing data with a length greater than zero to a 4704-2/3 that is operating in local tracking mode causes a condition code of X'02' and status in SMSDST of X'0480'.

Figure 5-5 is an example of writing a variable field to a printer.

```
PRNTRTN  WRTI      PB,X'15'     1
         BRAN      ST,PBERR     2
         SETSFP    PRNTOUT      3
         SETFPL    PRNTEND      3
         LWRITE    PB,PRNTSEG   4
         BRAN      ST,PBERR     2
         LCHECK    PB           5
         BRAN      ST,PBERR     2
```

1 Request a new line.
2 Check status.
3 Set PFP and SFP (PFP points to end, plus one).
4 Write the data to the passbook printer.
5 Wait for completion status so same data area can be reused.

Figure 5-5. Printing a Passbook

SIGNAL -- Set/Reset Indicators

SIGNAL sets or resets the indicators of the terminal component specified in the parameter byte addressed by operand 1. The parameter byte comprises a 1-bit on/off action indicator and a bit map of the indicator lights which are to be set or reset:

Bit:	*Explanation:*
0	On/off action indicator: 1 = switch on specified indicators 0 = switch off specified indicators
1-3	Reserved
4	System/check indicators
5	Indicator 3
6	Indicator 2
7	Indicator 1

Indicators 1, 2, and 3 are meaningful for the display and magnetic stripe encoder; indicators 1 and 2 are meaningful for the 3610, 3611, and 3612. If the corresponding indicator bit is not set in the parameter byte or if the indicator bit is set but the indicator does not exist, SIGNAL completes with a successful condition code.

When SIGNAL refers to a keyboard, the parameter bits are ignored. The check indicator (bit 4) is turned on, and the keyboard is set into purge mode. Purge mode causes all input to be discarded until the operator presses the Reset key. Refer to Chapter 2, "Programming for Displays and Keyboards" for more information on purge mode operation.

The SIGNAL parameter byte for the 4710, 4720, and 3616 is as follows:

Bit:	*Explanation:*
0	On/off action indicator 1 = switch on specified indicators 0 = switch off specified indicators
1-2	(reserved)
3	Journal print station forms insert (4710 and 4720: reserved)
4	Document print station forms insert (4720: reserved)
5	Indicator 3 (4710: reserved)
6	Indicator 2
7	Indicator 1

Name	Operation	Operand
[label]	SIGNAL	lda, { defcon1 | defld1 | seg1,disp1 | (reg1) | (defrf1) } [, { WAIT | <u>NOWAIT</u> }]

lda

Is a decimal number (0-7) indicating the logical device address (LDA) assigned to the terminal component. If the LDAs were assigned by default during the controller configuration procedure, the LDA values are as defined at the beginning of this chapter.

operand 1

Defines the parameter byte described earlier.

WAIT

Specifies that the application program waits until the SIGNAL instruction is completed and status is stored before execution continues with the next sequential instruction.

NOWAIT

Specifies that the application program continues execution after issuing a SIGNAL instruction without waiting for the SIGNAL instruction to be completed.

Condition Codes: One of the following is set:

Hex Code:	**Explanation:**
01	The instruction executed successfully.
02	Status is returned; the status code is contained in SMSDST. (See Appendix D, "Terminal and Device Status Codes" for an explanation of the status codes.)

Program Checks: 1, 2, or 27 can be set.

STPLPS -- Stop Loops

STPLPS deactivates one or all loops that are currently in error recovery. STPLPS uses a 3-byte parameter list that names all loops or the individual loop to be stopped. The parameter list is in the following format:

Byte:	*Explanation:*
0	X'00'—Stop all inoperative loops. X'08'—Stop the loop indicated in byte 2.
1	Reserved.
2	If byte 0 equals hex 08, this byte contains the binary loop number. If this loop number is invalid, no operation takes place.

Name	Operation	Operand
[label]	STPLPS	{ defcon1 | defld1 | seg1,disp1 | (reg1) | (defrf1) }

operand 2
Defines that the start of the parameter list. The length associated with this operand is ignored, and the first 3 bytes are assumed to be the parameter list.

Condition Codes: The code is not changed.

Program Checks: 1, 2, or 27 can be set.

STRLPS -- Start Loops

STRLPS activates one or all deactivated loops. STRLPS uses a 3-byte parameter list that names all loops or the individual loop to be started. The parameter list is in the following format:

Byte	*Explanation:*
0	X'00'—Start all inoperative loops.
	X'08'—Start the inoperative loop indicated in byte 2.
1	Reserved.
2	If byte 0 is hex 08, this byte contains the binary loop number. If the loop number is invalid, no operation takes place.

Name	Operation	Operand
[label]	STRLPS	{ defcon1 / defld1 / seg1,disp1 / (reg1) / (defrf1) }

operand 2
Defines the start of the parameter list. The length associated with this operand instruction is ignored, and the first 3 bytes are assumed to be the parameter list.

Condition Codes: The code is not changed.

Program Checks: 1, 2, or 27 can be set.

SWAPTT -- Exchange (Swap) Translate Tables

This instruction exchanges, or “swaps”, the input or output translate table for an LDA in this program’s work station with another that you define. The table name and the LDA to which it applies are in a parameter list that you must define.

Name	Operation	Operand
[label]	SWAPTT	{ defld2 | seg2,disp2 | (reg2) | (defrf2) }

operand 2

Defines the location of a nine-byte parameter list with the following format:

Byte:	*Explanation:*
Byte 1	LDA whose table is exchanged, in binary (X‘00’—X‘07’) or character (C‘0’—C‘7’) form.
Bytes 2—9	The label of an input (TRTBHDR) or output (OUTRTBL) translate table that is exchanged for the table currently assigned to the LDA.

You must define and initialize the parameter list. When SWAPTT is completed, the parameter list contains the name of the translate table removed from the LDA. A later SWAPTT instruction can then restore the old table. An identifier less than eight bytes long is padded in the high-order positions with blanks (C‘ ’).

Condition Codes: The following codes are set:

Hex Code:	**Explanation:**
01	Successful completion
02	Translate table not found

Program Checks: 1, 2, or 27 can be set.

Programming Note: Do not issue a SWAPTT instruction to a device with a PSSLST macro defined during CPGEN.

WRTI -- Write Immediate

WRTI writes 1 or 2 bytes of data to the specified terminal. Refer to the appropriate device chapter for control characters that can be sent to a terminal. When writing to a 3606 or 3608, you must first set the terminal group unit (SMSTGU) field before issuing WRTI.

When SPLIT=Y is specified, the WRTI instruction expands to a DEFCON and an LWRITE instruction.

Name	Operation	Operand
[label]	WRTI	{ lda | mnemonic } ,immdata2 [, { WAIT | NOWAIT }]

lda
Is a decimal number (0-7) indicating the LDA assigned to the terminal. If LDAs were assigned by default during the controller configuration procedure, the LDA values are as defined at the beginning of this chapter.

operand 2
Is 1 or 2 bytes of immediate data.

WAIT
Specifies that the application program waits until the WRTI instruction is completed and status is stored before execution continues with the next sequential instruction.

NOWAIT
Specifies that the application program will continue processing after issuing a WRTI instruction without waiting for the WRTI instruction to be completed.

Condition Codes: One of the following is set:

Hex Code:	Explanation:
01	The WRTI instruction completed successfully.
02	Status was returned in SMSDST. Refer to Appendix D, "Terminal and Device Status Codes" for an explanation of the status codes.

Program Checks: None are set.

Programming Notes

- If the WRTI is located so that the generated DEFCON instruction is beyond the 4K fixed addressing limit, then the LWRITE instruction will produce an assembly error.
- Only one DEFCON instruction is generated within each program section for the same *immdata* value.
- Figure 5-4 on page 5-29 and Figure 5-5 on page 5-32 show WRTI examples.

DATSM Instructions

These instructions require that optional modules P23 and P70 be specified by the OPTMOD configuration macro. The instruction descriptions in this section each begin on a new page so they can be easily removed and reorganized according to your particular site procedures.

DATSM Instruction Condition, Status, and Error Codes

The DATSM instructions set condition codes just as other 4700 instructions, but there are additional conditions that the codes represent for DATSM that do not apply to other instructions. Refer to Appendix C, "Program Check Codes" for the descriptions of all programming checks.

Each DATSM instruction can set its own status codes. These codes are in DMSSTAT, the DATSM status field. Don't confuse these codes with the status returned by 4700 assembler instructions in SMSDST, which is not affected by the DATSM instructions. The DATSM status codes define actions to take, depending on the conditions that result from issuing a DATSM instruction.

Each of the instruction descriptions in this section defines the possible DMSSTAT status codes that the instruction can set, and their meanings. The status is significant by bit. A DATSM instruction can return more than one DMSSTAT bit. Your program should therefore test the DMSSTAT bits in the order shown in Figure 4-3 on page 4-11 .

DATSM also returns error codes in the DMSERCD field under certain conditions. The error code depends on both the value you specify in the control bytes for the instruction, and on the instruction's operation. If the control byte descriptions for a given DATSM instruction say that an error code can occur, your program should test for that condition in DMSERCD. Appendix F, "DATSM Sample Program and Error Codes" describes the DATSM error codes and their meanings.

GETDMS -- Move Field Control Table

GETDMS reconstructs the content of the DMS body of any station generated to use DATSM. It reconstructs the DMS control area from the contents of the Field Control Table (FCT) header (not from the current DMS) so that if a comparison between contents of the user's DMS and those of the FCT is required, it can be made easily. GETDMS does not return the DMS header.

The 4700 application program must provide a receiving data buffer that is equal to or larger than the DMS control area. A program check will result if the data buffer length in the FLI is of insufficient length.

The PFP and the FLI are unchanged. The DMS constructed from the FCT by GETDMS can be used as an active DMS for instruction control only if the user ensures the presence of the necessary DMS header bytes. The DMS can be put in any segment except segment 14. The newly constructed DMS is not used by the DATSM instructions unless its location is communicated to the instructions by use of PUTDMS.

The application program must set the segment PFP to the beginning of the area that will receive the reconstructed DMS, and the FLI must be equal to or greater than the length of DMS or to zero. From the PFP to the end of segment must be greater than or equal to the DMS length. The application program can set the FLI to zero and PUTDMS will then assume the length of the area to be from the PFP to the end of segment. Failure to set the FLI correctly will cause a program check.

Name	Operation	Operand
[label]	GETDMS	{ seg2 | (reg2) }

operand 2
: Selects the area to receive the contents of the FCT. Do not specify segment 14, or a refreshable object.

Condition Codes: The code is not changed.

Program Checks and Error Codes: GETDMS can set program checks 3, 11, and DATSM error codes in DMSERCD. Refer to Appendix F, "DATSM Sample Program and Error Codes" for a description of the error code meanings.

Status Codes: Figure 5-7 defines the possible status codes set by GETDMS. Refer to Figure 4-3 on page 4-11 for the recommended test sequence of the status bits.

xxxxnnnn xxxxxxxx	Station number Bits 8-15 specify the number of the station for which the DMS is reconstructed. A zero value in this byte specifies the current station. GETDMS does not use the DMS control byte flags. If the application program addresses a station whose configuration does not permit the use of DATSM, DATSM returns error code 10 with bit 6 of the status byte set.
0000xxxx 00000000	Reserved.

Figure 5-6. GETDMS Control Bytes

000000xx xxxxxxxx	Reserved.
xxxxxx1x xxxxxxxx **xxxxxx0x xxxxxxxx**	Error code byte set. No error code byte set. Bit 6 set to 1 indicates that an error condition exists and that an error return code has been set in the DMS header. Error code 10 can be set.
xxxxxxx1 xxxxxxxx **xxxxxxx0 xxxxxxxx**	Successful completion. Status returned. Bit 7 set to 1 indicates that the instruction has completed its task successfully and that no additional status was returned.
xxxxxxxx 00000000	Reserved.

Figure 5-7. GETDMS Status Codes

GETFLD -- Retrieve Image Buffer Contents

GETFLD locates and transfers a field or set of fields in the image buffer to the user data buffer specified in the segment operand of the instruction. The data form is specified by the data definition control bytes located in the DMS header.

The PFP of the segment indicates the first byte of the user area. The FLI contains the length of the user area which receives the data. If the FLI equals zero, the area from the PFP to the end of segment is used. At least 50 bytes of segment space must be available, or a program check will result.

GETFLD can retrieve the following data streams (refer to Figure 4-1 on page 4-6):

- A2—An SCS processor-bound data stream or a processor-bound data stream with user delimiters
- B2—A 3270 processor-bound data stream
- C1—An SCS display-bound data stream
- D1—A printer-bound line of data

After execution of the instruction to retrieve the content of a single field, the PFP is unchanged, and the FLI indicates the length of data moved.

After execution of the instruction to retrieve all fields of the indicated class, the SFP points to the first byte of the generated string, the PFP points one byte beyond the end of the generated string, and the FLI is set to zero.

Name	Operation	Operand
[label]	GETFLD	{ seg2 / (reg2) }

operand 2
Selects the user area to receive the moved data. Do not specify segment 14, or a refreshable object.

Condition Codes: The code is not changed.

Program Checks and Error Codes: PUTDMS can set program checks 3, 11, and DATSM error codes in DMSERCD. Refer to Appendix F, "DATSM Sample Program and Error Codes" for a description of the error code meanings.

Status Codes: Figure 5-9 defines the possible status codes set by GETDMS. Refer to Figure 4-3 on page 4-11 for the sequence in which status bits should be tested.

Summary of GETFLD Segment Pointer Operation:

Case 1 = when the GETFLD is for one field of any class.

Case 2 = when GETFLD is for all fields of any class or when "generate cursor address only" is specified.

Field	**Before Execution**	**After Execution**
Case 1	PFP start of data	unchanged
	FLI length or 0 for end-of-segment	length of returned data field
	SFP used	unchanged
Case 2	PFP	EOD + 1, up to FLI length
	FLI (same as above)	FLI-(PFP-SFP), or zero
	SFP	equal to PFP

00xxxxxx xxxxxxxx Produce display-bound SCS.

Bits 0 and 1 set to 00 indicate that a display-bound SNA Character String should be produced (C1 flow).

01xxxxxx xxxxxxxx Produce processor-bound SCS.

Bits 0 and 1 set to 01 indicate that a processor-bound SNA Character String should be produced (A2 flow).

10xxxxxx xxxxxxxx Reserved. Flagged as error.

11xxxxxx xxxxxxxx Produce processor-bound 3270 data.

Bits 0 and 1 set to 11 indicate that a processor-bound 3270 data stream should be produced (B2 flow).

Note: The only valid control bytes for this flow are X'C080', X'CA00', and X'EA00'. The reissue bit may also be set; in this case, all other bits are ignored. Any other combination is rejected with status. The data stream produced begins with the attention ID (AID) byte and is followed with the cursor address (two bytes).

Figure 5-8 (Part 1 of 5). GETFLD Control Bytes

xx1xxxxx xxxxxxxx	Produce read-modified all data stream.
xx0xxxxx xxxxxxxx	Do not produce read modified all data.
	Bit 2 set to 1 causes GETFLD to generate a processor-bound 3270 data stream of all modified fields (MDTs set to 1) regardless of the EID value. See the *3270 Component Description* manual for data stream definition.
xxx1xxxx xxxxxxxx	Produce data only, no orders.
xxx0xxxx xxxxxxxx	Produce a normal data stream.
	Bit 3 set to 1 causes order generation to be suppressed for SNA character strings only; 3270 order generation cannot be suppressed. If this control flag is set for a GETFLD on the C1 flow, the data is returned to the application program as it is found in the image buffer, nulls (X'00') included, no orders. For the A2 flow, blanks (X'40'), are substituted for 4 controls the data stream produced.
xxxx1xxx xxxxxxxx	Get all fields of the requested class.
xxxx0xxx xxxxxxxx	Get one field of the requested class.
	Bit 4 set to '1' indicates that GETFLD is to scan the FCT for all fields having the specified attributes. When set to '0', bit 4 causes GETFLD to return all fields with the specified attributes to be returned in the order of their positions on the display.
	For operations other than "get current presentation position field", DMSFSEQ determines the first field obtained. If the request is for the field selected by the current field sequence number, that field is obtained. Otherwise, DMSFSEQ increments by one and GETDMS searches until the requested field or fields are found. When completed, GETDMS sets the sequence number of the last field obtained into DMSFSEQ if the request was for one field. When requests are for more than one field, GETDMS sets DMSFSEQ to zero.
	If all fields in a particular class are requested, DMSFSEQ must be set to zero before GETDMS is issued. This ensures that the first field is selected.

Figure 5-8 (Part 2 of 5). GETFLD Control Bytes

xxxxx000 xxxxxxxx	Get current field If the current field is specified, bit 4 controls the data produced. If bit 4 is set to B'1', the SFP will be set to the beginning of the stream and the PFP will be set to the end plus one. If bit 4 is set to B'0', the PFP will be set to the beginning of the field and the FLI will be set to the length. In either case, the data from the current field only will be returned. The current field is the field in which the cursor is currently positioned. If the cursor is positioned at an attribute byte when the current field is requested, a status indication of "field not found" (bit 3) is returned.
xxxxx001 xxxxxxxx	Get display-bound modified fields.
xxxxx010 xxxxxxxx	Get processor-bound modified fields.
xxxxx011 xxxxxxxx	Get user flagged fields.
xxxxx100 xxxxxxxx	Get unprotected fields.
xxxxx101 xxxxxxxx	Get the field sequence number field.
xxxxx111 xxxxxxxx	Get all classes.
xxxxxxxx 1xxxxxxx **xxxxxxxx 0xxxxxxx**	Read full buffer required. Read full buffer not needed. Bit 8 set to 1 causes GETFLD to produce the data stream required for a 3270 full buffer read command. All data is read out of the image buffer. GETFLD places a Start Field (X'1D') character in front of each attribute byte and all bytes of data, including nulls, into the data stream. The application program should issue a GETFLD instruction with this flag set if PUTFLD returns status (X'0080') indicating that a full buffer read is requested (B1 flow). The application program should send the data received from the GETFLD to the host. This control bit is valid only on the B2 flow; it is ignored on the C1 and A2 flows. This bit is valid only if bits 0 and 1 are set to B'11'.

Figure 5-8 (Part 3 of 5). GETFLD Control Bytes

xxxxxxxx x1xxxxxx	This is a reissue of GETFLD.
xxxxxxxx x0xxxxxx	This is not a reissue.
	Bit 9 set to 1 indicates that GETFLD is being reissued in response to a GETFLD buffer overflow (X'0010'). This control bit signals to continue processing at the point where interrupted. If this bit is set, all other control bits are ignored.
xxxxxxxx xx1xxxxx	Provide attributes for SCS fields.
xxxxxxxx xx0xxxxx	Do not provide attributes for SCS.
	Setting bit 10 to 1 indicates that GETFLD should include attributes in a display-bound SNA character string. In this case, all SCS fields begin with the attribute string, X'28C0aa', where "aa" is a 3270-type attribute byte. DATSM processes the nondisplayable attribute byte by replacing any data from a nondisplayable field with blanks. Refer to the description of GETFLD status bit for nondisplayable fields. This bit has meaning only on the C1 flow (see Figure 4-1 on page 4-6). DATSM ignores this bit on the C2 flow, and it causes DMSSTAT status on the B2 flow.
xxxxxxxx xxx1xxxx	Retrieve data for printer.
xxxxxxxx xxx0xxxx	Printer format is not requested.
	This bit indicates output intended for a printer. GETFLD moves a single image buffer line, without addressing orders, to the data buffer and appends a "new line" order to the end. Blanks and nulls are suppressed, beginning with the rightmost non-blank, non-null character and continuing to the end of the line. GETFLD translates nulls within the line to blanks (X'40'). GETFLD uses the line sequence number field (DMSLSEQ) in the DMS to determine the line to retrieve. A value of 0 or 1 in the line sequence number field identifies the first line. GETFLD increases the line number by one after each line is returned. After returning the last line, GETFLD sets this value to zero.
	As for all GETFLD operations, the line sequence number is applied to the window area currently defined for the image buffer. This control bit should be the only control bit set, but if bit 9 is set, that function takes precedence.

Figure 5-8 (Part 4 of 5). GETFLD Control Bytes

xxxxxxxx xxxx1xxx	Insert user delimiters.
xxxxxxxx xxxx0xxx	Do not insert delimiters.
	Bit 12 set to 1 indicates that GETFLD will precede each data field with the DMS delimiter found in DMSDELM when placing the output data stream into the user's data buffer. Effectively, the delimiter is replacing the control orders. This control bit has value only on the A2 flow. Status is returned on the B2 flow, and the bit is ignored on the C1 flow.
xxxxxxxx xxxxx1xx	Produce cursor address only.
xxxxxxxx xxxxx0xx	Produce full data string.
	Bit 13 set to 1 causes GETFLD to generate a string consisting only of the cursor address in 4700 form. Upon completion, the PFP is set to the first byte beyond the string, the SFP is set to the first byte of the string. This control bit has meaning only on the C1 flow: It is ignored on the A2 flow and status is returned if it is set on a B2 flow request. This bit should be the only control bit set, but if bits 9 or 10 are set also, they will take precedence respectively.
xxxxxxxx xxxxxx1x	Set PPF address to address of field.
xxxxxxxx xxxxxx0x	Do not change PPF address.
	Bit 14 set to 1 instructs GETFLD to set the presentation position field in the FCT header to the address of the first byte of data of the last field accessed. This is valid on all flows.
xxxxxxxx xxxxxxx1	Segment field retrieved.
xxxxxxxx xxxxxxx0	Do not segment field retrieved.
	Bit 15 set to 1 causes GETFLD to segment the current field one window row at a time. This function applies only to the retrieval of the current field. The bit is ignored for all other GETFLD operations. It is primarily useful when the window to be displayed is smaller than the actual device. Reading one line of a multi-line field at a time enables the input to be tracked within the boundaries of the window. A further description appears under "Programming Examples of Window Control." PUTFLD must also set this bit.

Figure 5-8 (Part 5 of 5). GETFLD Control Bytes

1xxxxxxx xxxxxxxx	Requested field was non-displayable.
0xxxxxxx xxxxxxxx	Field requested was displayable. Bit 0 set to 1 indicates that data from a non-displayable field has been placed in the user's data buffer. If bit 4 in the control bytes is set to 1 (get all fields of a requested class), this status flag will not be set. This flag is set for any flow and may be set along with bit 5. Action: No action is needed.
x000xxxx xxxxxxxx	Reserved.
xxxx1xxx xxxxxxxx	Write to host required.
xxxx0xxx xxxxxxxx	Write to host not required. Bit 4 set to 1 indicates that a data stream is placed into the data buffer for the host. The data stream is the result of input to the PUTFLD instruction that required data transmission to the host. This flag is set only on the C2 flow and can be set with Bit 2 or 3. Action: The segment pointers are set up for an LWRITE CP to the host. The application program should issue the LWRITE CP to send the prepared data stream to the host.
xxxxx1xx xxxxxxxx	User flag turned on for field.
xxxxx0xx xxxxxxxx	User flag was not turned on. Bit 5 set to 1 indicates that the user flag was turned on for the field just retrieved for a single field request. Action: Presented for informational purposes only; no action required.
xxxxxx1x xxxxxxxx	Error code set.
xxxxxx0x xxxxxxxx	No error code set. Bit 6 set to 1 indicates that an error condition was found by GETFLD and that the error code byte of the DMS header contains an identifying number. GETFLD sets this flag on for any flow without accompanying bits. Action: The program can test the error code to determine the type of error; but in most cases, no dynamic recovery is possible.

Figure 5-9 (Part 1 of 3). GETFLD Status Codes

xxxxxxx1 xxxxxxxx	Successful completion, no status.
xxxxxxx0 xxxxxxxx	Completion, status returned. Bit 7 set to 1 indicates that GETFLD successfully completed, and no additional status was returned. If this flag is set to 0, other status is available, and your program must perform additional checks. GETFLD sets this flag for any flow without accompanying bits. Action: No action required.
xxxxxxxx 000xxxxx	Reserved.
xxxxxxxx xxx1xxxx	Buffer overflow.
xxxxxxxx xxx0xxxx	No buffer overflow. Bit 11 set to 1 indicates that the user buffer is not long enough for the data being returned. This indicator can segment data from the image buffer, reducing the storage requirements of the application program. If this status is detected when requesting a single field (bit 4 equal to 0), there is a good chance that an application program problem exists. The user buffer should be large enough to permit handling of the largest field to be processed. Action: If the current field is requested with no orders to initialize an area for an LREAD operation (C1 flow), and this error occurs, only that portion of the field that fits in the user buffer is returned for use on the LREAD. Field segmenting can be used to avoid this condition. If the application program uses multiple fields for initializing the LREAD area, it is up to the user to take the appropriate action. When this indicator is returned when requesting all fields of a class, the application program should write the data to the host using a chaining protocol (A2 or B2 flow). Reissuing a GETFLD instruction with the reissue flag set continues the process. For the C1 flow, the application program should write the data to the screen followed by another GETFLD instruction with the reissue flag set.

Figure 5-9 (Part 2 of 3). GETFLD Status Codes

xxxxxxxx xxxx1xxx Field not found.
xxxxxxxx xxxx0xxx Field found or not requested.

Bit 12 set to 1 indicates that the field specified by DMSFSEQ was not found during the GETFLD operation. This can be caused by one or more of the following:

1. The application program requested the next field of a class and no more fields of that class existed.

2. The application program searched past the last field without realizing it.

3. The PPF was pointing at the attribute byte of a field when the GETFLD was issued.

Action: User-defined. This status bit can be set by either normal operation, or to indicate a program problem.

xxxxxxxx xxxxx0xx Reserved.

xxxxxxxx xxxxxx1x Field is protected.
xxxxxxxx xxxxxx0x Field is unprotected.

Bit 14 set to 1 indicates that the field retrieved is protected. If bit 4 in the control bytes (get all fields of the requested class) was set to 1, this status flag is not set.

Action: Presented for information purposes only; no action is required.

xxxxxxxx xxxxxxx1 The field retrieved was marked KANJI.
xxxxxxxx xxxxxxx0 The field retrieved was not KANJI.

Bit 15 set to 1 indicates that GETFLD has retrieved a KANJI field. This flag is set when a single field has been requested by the application program.

Action: The application program should take appropriate action to permit correct processing of the KANJI field.

Figure 5-9 (Part 3 of 3). GETFLD Status Codes

Summary of GETFLD Segment Pointer Operation:

Case 1 = when the GETFLD is for one field of any class.

Case 2 = when GETFLD is for all fields of any class or when "generate cursor address only" is specified.

Segment Addressing:

Field	Before Execution	After Execution
Case 1	PFP start of data	unchanged
	FLI length or 0 for end-of-segment	length of returned data field
	SFP unused	unchanged
Case 2	PFP	EOD + 1, up to FLI length
	FLI (same as above)	FLI-(PFP-SFP), or zero
	SFP	equal to PFP

Register Addressing:

Field	Before Execution	After Execution
Case 1:	Displacement: start of data	unchanged
	Length: length or 0 for End-of-segment	length of returned data field
Case 2:	Displacement: (same as above)	start of data
	Length: (same as above)	length of returned data field

PUTDMS -- Modify Field Control Table

The Field Control Table (FCT) controls DATSM, and is not accessible by the application program. The content of the FCT is produced by the PUTDMS instruction, which communicates user-supplied values to DATSM, and by the DATSM instructions themselves as they process the data from the data streams and the image buffer.

The program must issue PUTDMS to set the required values from the DATSM machine segment (DMS) into the FCT before issuing any of the other DATSM instructions. Your program can also issue PUTDMS any time to change the data mapping specifications.

The DMS control bytes determine the initial settings and later changes to the FCT. Each bit in a DMS control byte corresponds to a DMS field. When your program issues PUTDMS with a DMS control bit set to 1, the instruction transfers the values in the corresponding DMS field to the related FCT field. The application program can later set or change any or all of the DMS fields that PUTDMS moves into the FCT.

Note: To prevent the change of selected fields, leave the appropriate control bits set to 1.

PUTDMS saves the address of the DMS in the FCT. If any other DATSM instruction is issued before the location of the DMS has been so established, DATSM returns a program check of 11.

The application program must set the segment PFP to the beginning of DMS and the FLI equal to or greater than the length of DMS or zero (from the PFP to the end of segment must be greater than or equal to the DMS length). An incorrect FLI setting causes a program check of 3.

Name	Operation	Operand
[label]	PUTDMS	{ seg2 | (reg2) }

operand 2
: Selects the segment containing the DMS. Do not specify segment 14 or a refreshable object.

Condition Codes: The code is not changed.

Program Checks and Error Codes: PUTDMS can set program checks 3, 11, and DATSM error codes in DMSERCD. Refer to Appendix F, "DATSM Sample Program and Error Codes" for a description of the error code meanings.

Status Codes: Figure 5-13 defines the possible status codes set by GETDMS. Refer to Figure 4-3 on page 4-11 for the sequence in which status bits should be tested.

rxxrrrxr rrrrrrrr	Reserved bits (indicated as "r").
x0xxxxxx xxxxxxxx	Screen is not formatted.
x1xxxxxx xxxxxxxx	Screen is formatted.
xx0xxxxx xxxxxxxx	No unprotected areas on screen.
xx1xxxxx xxxxxxxx	Unprotected areas exist on screen.
xxxxxx0x xxxxxxxx	Normal mode (not insert) of operation.
xxxxxx1x xxxxxxxx	Insert mode of operation.

Figure 5-10. DMS Screen Status Flags (DMSFLAG)

0xxxxxxx xxxxxxxx	Presentation position not present.
1xxxxxxx xxxxxxxx	Presentation position present.
x1xxxxxx xxxxxxxx	Original display size present.
x0xxxxxx xxxxxxxx	Original display size not present.
xx0xxxxx xxxxxxxx	Reserved.
xxx1xxxx xxxxxxxx	Actual display dimensions present.
xxx0xxxx xxxxxxxx	Actual display dimensions not present.
xxxx0xxx xxxxxxxx	Reserved.
xxxxx1xx xxxxxxxx	Actual display window coordinates present.
xxxxx0xx xxxxxxxx	Actual display window coordinates absent.
xxxxxx1x xxxxxxxx	Image buffer window coordinates present.
xxxxxx0x xxxxxxxx	Image buffer window coordinates not present.
xxxxxxx1 xxxxxxxx	Window size present.
xxxxxxx0 xxxxxxxx	Window size not present.
xxxxxxxx 00000000	Reserved.

Figure 5-11. PUTDMS Control Bytes

0000xxxx xxxxxxxx	Base attribute mode of operation.
100xxxxx xxxxxxxx	Kanji mode
010xxxxx xxxxxxxx	Extended attribute mode
001xxxxx xxxxxxxx	4704-2/-3 local tracking mode
xxx1xxxx xxxxxxxx	SCS default field generation

Figure 5-12. PUTDMS Mode Bytes

00000xxx xxxxxxxx	Reserved.
xxxxxx1x xxxxxxxx **xxxxxx0x xxxxxxxx**	Error code byte set. No error code byte set. Bit 6 set to 1 indicates that an error condition exists and that an error return code has been set in the DMS header (DMSERCD). Coordinate positions must not exceed the dimensions of the associated area. The window dimensions to be applied to the image buffer, when positioned according to the buffer coordinates must not overrun the image buffer area in either dimension. An error code of 12 will be returned for any of the conditions above. When an invalid parameter is detected, PUTDMS processing stops, leaving the rest of the parameters unexamined. When an error is returned, DATSM marks the field control table as uninitialized and a valid PUTDMS must be issued before other DATSM instructions can be issued successfully.
xxxxxxx1 xxxxxxxx **xxxxxxx0 xxxxxxxx**	Successful completion. Bit 6 was set to B'1'. Bit 7 set to 1 indicates that the instruction has completed successfully and that no additional status was returned
xxxxxxxx 00000000	Reserved.

Figure 5-13. PUTDMS Status Bytes

PUTFLD -- Map Data to Image Buffer

PUTFLD maps DATSM input data streams from a user's data buffer into the image buffer. It accepts the following input data streams (see Figure 4-1 on page 4-6):

A1 — A display-bound SNA character string or a display-bound data stream with user-defined field delimiters

B1 — A 3270 display-bound data stream

C2 — A processor-bound SNA character string (see Figure 7-1)

Your program must ensure that data stream mapping controls are complete before issuing a PUTFLD instruction. This responsibility comprises three areas: following for each invocation of PUTFLD:

- The DMS control byte flags.
- The DMS header fields (including DMSIML and DMSEID for the C2 flow).
- The fields in the DMS body in effect from the last PUTDMS.

To send data on the C2 flow, the DMSIML and DMSEID values must reflect the values of SMSIML and SMSEID, respectively, at the termination of the LREAD KB operation. PUTFLD uses the DMSIML value to determine the current cursor position and DMSEID to determine the action to be taken.

Your program must set the PFP to select the first data byte and set the FLI to the data length. If your program performs tabbing or keyboard emulation, the area from the PFP to the end of segment must be available for PUTFLD use. This area must always be at least 50 bytes long, or a PUTFLD program check occurs.

If you issue PUTFLD as a result of status from a preceding PUTFLD (the reissue flag is set in the DMS control field), the PFP and FLI are ignored. If the PUTFLD reissue passes a middle- or last-in-chain block on the A1 or B1 flows, PUTFLD uses the PFP and FLI values you set to define the input data area.

The length specified in the FLI for PUTFLD use on the C2 flow should be equal to the current field length from the PPF to the end. The contents of the current field from the PPF to the end should have been placed in the user buffer (using GETFLD) prior to issuing a read to the keyboard (see DATSM Programming Considerations established earlier in this chapter).

After execution of the instruction, the PFP, SFP, and the FLI remain unchanged if an LREAD, LWRITE, or a reissue is not requested by return status. If any of these actions are requested in status returned, the PFP, SFP, and FLI will be altered depending on the status bit set (see status field description to determine which segment values are affected).

Depending on the data passed to DATSM, status bits will be set in the DMS header by PUTFLD to indicate the next action, if any, to be taken by the application program. If the application program issues a PUTFLD instruction before DMS has been initialized, program check 11 is returned.

Name	Operation	Operand
[label]	PUTFLD	{ seg2 | (reg2) }

operand 2

Selects the area containing the data stream or character string being mapped into the image buffer. Do not specify segment 14 or a refreshable object.

Condition Codes: The code is not set.

Program Checks: PUTDMS can set program checks 3, 11, and DATSM error codes in DMSERCD. Refer to Appendix F, "DATSM Sample Program and Error Codes" for a description of the error code meanings.

Status Codes: Figure 5-15 defines the possible status codes set by PUTFLD. Refer to Figure 4-3 on page 4-11 for the sequence in which status bits should be tested.

00xxxxxx xxxxxxxx	SCS display-bound (A1 flow). PUTFLD constructs a field entry in the FCT for every field of data following an order, and for every attribute definition in the character string. Field entries are not constructed for data streams with user-defined delimiters. Field services for this stream must be constructed and entered into the FCT previously. DATSM does not permit mixing of user delimiters and positioning orders in this stream. An error code (invalid order or command) is set if mixing is found by PUTFLD.
01xxxxxx xxxxxxxx	SCS processor-bound (C2 flow). PUTFLD does not generate field entries in the FCT. Instead, the previously generated field entries are available for this data. The program must initialize DMSIML and DMSEID before issuing PUTFLD with this control bit combination.
10xxxxxx xxxxxxxx	3270 display-bound flow. PUTFLD generates field entries in the FCT for every start field order X'1D' in the data stream.
11xxxxxx xxxxxxxx	Invalid combination. PUTFLD will return an error code if this combination of bits is encountered.
xx0xxxxx xxxxxxxx	Reserved.
xxx1xxxx xxxxxxxx **xxx0xxxx xxxxxxxx**	Reset modified data tags. Do not reset modified data tags. PUTFLD returns error status if bits 0-1 are set to B'10' (3270 display-bound flow) and this bit is set to 1; otherwise, bit 3 set to 1 causes all modified data tags (MDT) to be reset. This facility is available for use with SCS field oriented display management.
xxxx1xxx xxxxxxxx **xxxx0xxx xxxxxxxx**	Set user flag in FCT entry. Do not set user flag. Bit 4 set to 1 causes PUTFLD to turn on the user flag for each field addressed by PUTFLD. This flag is detected by both PUTFLD and GETFLD.

Figure 5-14 (Part 1 of 3). PUTFLD Control Bytes

xxxxx1xx xxxxxxxx	Use sequence number to find field.
xxxxx0xx xxxxxxxx	Do not use sequence number.
	Bit 5 set to 1 causes PUTFLD to use the field sequence number in the DMS (DMSFSEQ) to locate the field entry for the data passed to the instruction. The first string of data in the data stream is placed into the location for that field. If there are user delimiters mixed with the data, the DMSFSEQ field is increased by one for each delimiter. When PUTFLD completes, DMSFSEQ is set to the number of the last field that was accessed.
	A programmer who uses this facility must understand the field structure, DATSM does not check for protected or numeric fields when it maps display-bound data streams into the image buffer.
xxxxxx1x xxxxxxxx	This string has user delimiters.
xxxxxx0x xxxxxxxx	This is not a delimited string.
	If bits 0-1 are set to B'01' (3270 display-bound flow) and this bit is set to 1, error status is returned; otherwise, bit 6 set to 1 indicates that the user-defined delimiter found in DMSDELM is to be used. This delimiter is used to separate fields. Data streams containing user delimiters must not contain SCS positioning orders; additionally, this data stream cannot be used to define fields.
	A programmer who uses this facility must understand the field structure because DATSM makes no checks for protected or numeric fields when it maps display-bound data streams into the image buffer.

Figure 5-14 (Part 2 of 3). PUTFLD Control Bytes

xxxxxxx0 0xxxxxxx	Reserved.
xxxxxxxx x1xxxxxx	This is a PUTFLD reissue.
xxxxxxxx x0xxxxxx	This is not a reissue.
	Bit 9 set to 1 indicates a reissue. It is set by the application program when it reissues PUTFLD as a result of previous reissue status. This occurs on the C2 flow. It can be used when processing a screen image in segments (SNA communication protocol refers to these as chains). When processing chains (segments), the first-in-chain is passed with this indicator set to 0; all remaining elements of the chain are passed with this indicator set to 1. All orders that span elements of the chain are resolved correctly. This is used on the A1 and B1 flows. It has little use for the C2 flow.
	Errors in the data stream when PUTFLD resumes processing of broken orders are undetectable by DATSM, since the data values following 3270 or SCS orders are unpredictable. If PUTFLD is interrupted in the middle of processing an order, the first bytes of the continuing data stream are treated by PUTFLD as if they are the continuation of the order that was being processed.
xxxxxxxx xx1xxxxx	Insert attribute in current FCT entry.
xxxxxxxx xx0xxxxx	Do not insert attribute in FCT.
	When both the field sequence number bits 05 and 10 are set to 1, PUTFLD places the data byte in the user segment in the attribute byte field of the FCT entry currently being addressed. This process occurs on the A1 flow (SCS display-bound data).
xxxxxxxx xxx0000x	Reserved.
xxxxxxxx xxxxxxx1	Segment current field.
xxxxxxxx xxxxxxx0	Do not segment.
	Bit 15 set to 1 causes PUTFLD to retrieve only one segment of the current field at a time when PUTFLD performs a GETFLD function. This bit must be left on as long as current field segmenting is in effect.

Figure 5-14 (Part 3 of 3). PUTFLD Control Bytes

Summary of PUTFLD Segment Pointer Operation:

Case 1 = PUTFLD status bits do not indicate that LREAD, LWRITE, or reissue is required by the application program.

Case 2 = PUTFLD status bits indicate an LREAD is required by the application program.

Case 3 = PUTFLD status bits indicate an LWRITE is required by the application program.

Case 4 = PUTFLD reissued as a result of reissue status from the previous PUTFLD.

Segment Addressing:

Field	Before Execution	After Execution
Case 1	PFP start of data	unchanged
	FLI length of data	unchanged
	SFP unused	unchanged
Case 2	PFP start of data	unchanged
	FLI length of data	length of read through end of segment
	SFP unused	unchanged
Case 3	PFP start of data	end of data +1
	FLI length of data	residual length in segment
	SFP unused	start of data (PFP)
Case 4	PFP ignored	case 1, 2, 3, results possible
	FLI ignored	
	SFP ignored	

Field	Before Execution	After Execution
Case 1	Displacement: start of data	unchanged
	Length: length of data	unchanged
Case 2	Displacement: start of data	unchanged
	Length: length of data	length of read through end of segment
Case 3	Displacement: start of data	unchanged
	Length: length of data	length of data
Case 4	Displacement: ignored	case 1, 2, or 3
	Length: length of data	results are possible

1xxxxxxx xxxxxxxx	Issue LREAD NOTRACK to terminal.
0xxxxxxx xxxxxxxx	Do not issue LREAD NOTRACK to terminal. Bit 0 set to 1 indicates that PUTFLD has placed the current field into the user's buffer and that the program should issue a keyboard read instruction to read into this area with the NO TRACK option. This action is indicated during TAB and INSERT key processing for nondisplayable fields (see "Emulation of 3270 Keyboard" and "Tab Key Emulation"). This flag is set by PUTFLD only on the C2 flow and can be set along with bit 2 or 9. This flag is not set when buffer overflow has occurred. Action: PUTFLD has set the PFP and FLI in the segment header to read the data from the keyboard into the field in the user's buffer. Your program should issue an LREAD KB with the NO TRACK option so that no data is displayed.
x1xxxxxx xxxxxxxx	Issue LREAD TRACK to terminal.
x0xxxxxx xxxxxxxx	Do not issue LREAD track to terminal. Bit 1 set to 1 indicates the same as bit 0 (LREAD KB NO TRACK) except that the TRACK option should be used for the keyboard read. Action: PUTFLD has set the PFP and FLI in the segment header to read the data from the keyboard into the field in the user's buffer. The application program should issue an LREAD KB TRACK so that the data entered is displayed correctly.
xx1xxxxx xxxxxxxx	Reissue PUTFLD.
xx0xxxxx xxxxxxxx	Do not reissue PUTFLD. Bit 2 set to 1 indicates that all operations required by the data in process are not complete. This occurs during TAB and INSERT key processing. This flag is set only on the C2 flow and is set along with bits 0, 1, or 5. Action: The application program should set the *Reissue PUTFLD* bit in the DMS control byte to 1, and reissue the instruction for the same segment as for the original invocation. The application program should reissue PUTFLD only after the LREAD or LWRITE status requests are satisfied.

Figure 5-15 (Part 1 of 5). PUTFLD Status Codes

xxx1xxxx xxxxxxxx xxx0xxxx xxxxxxxx	User EID processing required. No EID processing required of user. Bit 3 set to 1 indicates that the DMSEID field contained a CLEAR key (X'6D') EID or an EID not supported by DATSM when PUTFLD was issued after an LREAD from the keyboard. PUTFLD sets this flag only on the C2 flow and can also set bit 5. Action: The application program should in most cases, issue GETFLD to generate a data stream for the host. If the user has defined EOM's not supported by host processing, they must be processed by the program. Note that you issue GETFLD for a 3270 processor-bound data stream, the DMSEID field is used for an attention ID (AID) byte.
xxxx1xxx xxxxxxxx xxxx0xxx xxxxxxxx	Write to host required. No write to host required. Bit 4 set to 1 indicates that a data stream is placed into the data buffer for the host. The data stream was the result of a PUTFLD instruction that transmitted to the host. This flag is set only on the B1 flow and can be set with bit 2, 3, or 11. Action: The segment pointers are set up for an LWRITE CP to the host, which should be issued by the program.
xxxxx1xx xxxxxxxx xxxxx0xx xxxxxxxx	Issue LWRITE to display. Do not issue LWRITE to display. Bit 5 set to 1 indicates that a data stream is in the data buffer for the terminal. The data stream is the result of input to the PUTFLD instruction that required a display screen action such as clearing the screen or tabbing to the next unprotected field. The display performs the action. This flag occurs only on the C2 flow and can be set with bits 2 or 3. Action: The segment pointers are set to issue an LWRITE DS instruction to the display; the application program can issue the instruction to write the data stream.

Figure 5-15 (Part 2 of 5). PUTFLD Status Codes

xxxxxx1x xxxxxxxx	Error code byte set.
xxxxxx0x xxxxxxxx	No error code byte set.
	Bit 6 set to 1 indicates that an error condition has been detected and that the error code byte of the DMS header contains an identifying number. PUTFLD sets this flag on any flow without accompanying bits.
	Action: The application program should test the error code byte; however in many cases, no recovery is possible.
xxxxxxx1 xxxxxxxx	Successful completion, no status set.
xxxxxxx0 xxxxxxxx	Status returned.
	Bit 7 set to 1 to indicate that PUTFLD operated successfully. If this flag is 0, other status is available and should be checked. PUTFLD sets this flag without any other bits. This flag is not set on the C2 flow after processing a *Reset Insert Mode* EOM.
	Action: Issue GETFLD to obtain the current C2 flow data, and then read from the keyboard. This is recommended on the C2 flow any time DATSM does not specify the function required next.
xxxxxxxx 1xxxxxxx	Issue GETFLD for read full buffer.
xxxxxxxx 0xxxxxxx	No read full buffer needed.
	Bit 8 set to 1 indicates that a read full buffer command code was received in the data stream. PUTFLD sets this flag only on the B1 flow and without any other bits.
	Action: Issue a GETFLD for a full read and use an LWRITE instruction to write it to the host.

Figure 5-15 (Part 3 of 5). PUTFLD Status Codes

xxxxxxxx x1xxxxxx	This field has the user flag set.
xxxxxxxx x0xxxxxx	This field has no user flag set.
	Bit 9 set to 1 indicates that PUTFLD encountered a field with a user flag set during tab processing. PUTFLD sets this flag only on the C2 flow and always with bits 0 and 1.
	Action: None, unless defined by your program.
xxxxxxxx xx1xxxxx	Read modified required.
xxxxxxxx xx0xxxxx	No read modified required.
	Bit 10 set to 1 indicates that a *read modified* command code was received in the input to the PUTFLD instruction. This flag is set only on the B1 flow and is set without any other bits.
	Action: Issue GETFLD for all modified fields. Depending on the EID value (in DMSEID) supplied by the program, the data stream you read contains either the AID and the cursor address only (short read); or the AID, the cursor position, and the contents of all fields marked as input-modified (MDTs set to 1). Issue an LWRITE instruction to write the result to the host.
xxxxxxxx xxx1xxxx	Buffer overflow.
xxxxxxxx xxx0xxxx	No buffer overflow.
	Bit 11 set to 1 indicates that the user buffer is not long enough for the data being retrieved after a tabbing operation.
	Action: This can be a program problem. This bit setting is not an error code since the program can read data into a partial field. The write-to-terminal bit, which is normally set as the result of the tabbing operation, is not set when buffer overflow has occurred. When single fields extend beyond one line of the display, field segmenting can be used to avoid this condition.

Figure 5-15 (Part 4 of 5). PUTFLD Status Codes

xxxxxxxx xxxx1xxx	Field not found.
xxxxxxxx xxxx0xxx	Field found or not requested.

Bit 12 set to 1 indicates that the field specified by DMSFSEQ could not be found. PUTFLD returns this status only if bit 5 of the control bytes (use sequence number to find the field) was set to 1 before PUTFLD instruction was issued, or when PUTFLD performs a "GETFLD-type" operation during keyboard emulation. This flag is set for all flows without accompanying bits.

Action: The action is user-defined. This status could indicate a program problem or simply that the field did not exist.

xxxxxxxx xxxxx1xx	Read modified all required.
xxxxxxxx xxxxx0xx	Read modified all not required.

Bit 13 set to 1 indicates that a command code of read modified all was received in the input to the PUTFLD instruction. This flag is set only on the B1 flow and is set without accompanying bits.

Action: The application program should:

1. Issue GETFLD all input-modified.

2. Use an LWRITE to write the resulting data stream to the host.

xxxxxxxx xxxxxx1x	Field is protected.
xxxxxxxx xxxxxx0x	Field is unprotected.

Bit 14 set to 1 indicates that PUTFLD has performed the GETFLD function on a protected field because of receiving an EID specifying one of the cursor movement keys. This bit occurs in conjunction with bits 0 or 1 and bit 2.

Action: The application program should take appropriate action to protect the contents of the protected field when issuing the LREAD KB indicated by bits 0 or 1 then reissue PUTFLD as indicated by bit 2.

xxxxxxxx xxxxxxx1	The field retrieved was marked KANJI.
xxxxxxxx xxxxxxx0	The field retrieved was not KANJI.

Bit 15 set to 1 indicates that PUTFLD has performed a GETFLD function on a KANJI field as the result of a keyboard emulation operation. This bit occurs with bits 0 or 1 and bit 2.

Action: The application program should issue an LREAD KB with tracking or not as indicated by bits 0 or 1 and then reissue PUTFLD as indicated by bit 2.

Figure 5-15 (Part 5 of 5). PUTFLD Status Codes

Part II. Terminal and Device Reference

This second part of the 4700 loop and DCA device volume contains device-oriented programming information. Where appropriate, each device type has its own chapter beginning with an introduction to the basic device features and continuing with configuration, data translation, device control, and other programming information unique to that device. For device status code, condition code, and DEVPARM device control parameters, refer to the appendixes at the back of the book.

Terminals and Devices Available on the 4700

The terminals that you can attach to a 4700 controller are:

- The IBM 4704 Display, Models 1 - 3, comprising one or two numeric or alphameric keyboards or an administrative keyboard and one of three 480- or 1920-character display monitors with optional tinted antiglare filters and adjustable holding cradle. Model 1 displays attach to the banking loop; Models 2 and 3 attach over the Device Cluster Adapter (DCA) feature. Optional attaching devices are a magnetic stripe reader or reader/encoder, and a normal or encrypting personal identification number (PIN) keypad.

- The IBM 4710 Document Printer, a combination of a receipt/validation printer and a journal printer. It can print a 96-character set in standard or bold sizes, and can be shared between two work stations.

- The IBM 4720 Printer, Models 1 - 4, providing cut-form, journal, and passbook printing capability. All models provide either normal dot matrix printing suitable for most transactions, or an optional high-resolution printing for administrative needs. These table-top printers operate at up to 120 characters per second.

- The IBM 3604 Keyboard Display, a combination of a numeric or alphameric keyboard and a 120-, 240-, 480-, 1024- or 1920-character display.

  The display character set is a fixed set of up to 153 unique characters (including characters for all supported languages). Magnetic stripe readers and reader/encoders can also be attached. A program-controlled audible alarm is available on Models 5 and 6.

- The IBM 3606 and 3608 Financial Services Terminals, which are designed for payment processing at the point of sale (charge card/credit card verification, account validity and status, and capture of sales data are examples). The IBM 3606 Financial Services Terminal has a keyboard, indicator lights, an 8-position numeric display, and a magnetic stripe reader. The IBM 3608 Printing Financial Services Terminal has the same configuration as the 3606, but with a printer capable of printing one to three lines on cut forms of one to three parts, card stock, and single-part cut forms. The printer has a standard printing set of 45 alphameric characters. An Optical Character Recognition (OCR) 7B font may be ordered for printing the top line of the three print lines. Ten numeric characters compose the OCR 7B font.

- The IBM 3610 Document Printer, which can be ordered as a cut-forms, a continuous-forms, or a journal-roll printer. It can print a 48-, 64-, 96-, or 128-character set (customized by country).

- The IBM 3611 Passbook Printer, which is a passbook printer that can also be used to print single or multipart cut forms. Form width can vary. Printing character sets of 64 or 96 characters are available (customized by country).

- The IBM 3612 Passbook and Document Printer, which is a combination of a document printer and a passbook printer. The two units are separately addressable and operate independently; however, data is transferred to only one printer at a time. The same character sets and printing capabilities available with the 3610 (with the exception of the 48-character set) are available with the 3612.

- The IBM 3614 Consumer Transaction Facility is used by the customers of a financial institution to perform transactions. The 3614 has numeric and transaction keypads, a credit card magnetic stripe reader, 40-character display, and a cash dispenser. Optional features are a receipt printer and depository. The 3614 can attach directly to the controller or through a communication link and 3704 or 3705 to the host. Refer to the *3614 Programmer's Guide and Reference*, GC66-0005, for detailed information about the 3614.

- The IBM 3615 Administrative Terminal Printer, which is a medium-speed, tabletop, matrix printer, can be used to print on a variety of cutforms and fanfold continuous forms. A character set of 128 alphameric characters can be obtained in U.S. English, Canadian English, and Canadian French.

- The IBM 3616 Passbook and Document Printer, which combines two logically independent matrix print stations, passbook and journal, into one terminal. The passbook station can handle multiple sizes of passbooks and cut forms. The journal station allows the use of one-part or two-part journal forms. In addition, the journal station prints cut forms that require printing for validation. The 3616 can print 120 characters per second at 10 or 12 characters per inch.

- The IBM 3624 Consumer Transaction Facility, which is used by the institution's customers, is similar to the 3614 but with an more function and features. The 3624 uses portable cartridges to make loading and issuing currency easier. With minor programming changes, the 3624 attaches to the same system as a 3614. Refer to the *3624 Programmer's Reference and Component Descriptions*, GC66-0009, for detailed information about the 3624.

- The IBM 3262 Line Printer is available in two models, depending on the printing speed desired. The Model 3 prints 132-character lines at up to 650 lines per minute. The Model 13 prints 132-character lines at up to 325 lines per minute. Both models attach to an IBM 4700 Finance Communication System controller through the Device Cluster Adapter (DCA) feature.

- The IBM 3178 Display Station, which attaches to the 4700 device cluster adapter (DCA). The 3178 operates identically with a 3278, Model 2. Refer to the 3178 operating instructions, which includes procedures for operating the 3178 when attached to a 4701 controller.

- The IBM 3278 Display Station Model 2 comprises a cathode ray tube (CRT) display component that displays up to 1920 characters, a 75- or 87-key keyboard that attaches with a cable to the display component, and an optional magnetic slot reader and accompanying control feature for reading American Banking Association (ABA) 10-character/"F"-EOS magnetic stripes. All components of the 3278 are separate, movable, tabletop units. Program control for the 3278-2 is the same as for the 3604 units. The 3278-2 attaches to the IBM 4700 Finance Communications System controller through the Device Cluster Adapter (DCA) feature. Refer to the *IBM 3270 Information Display System Component Description*, GA27-2749, for detailed information on the 3278-2.

- The IBM 3279 Display Stations, Models 2A-2B, are 1920-character color displays comprising a four- (Model 2A) or seven-color (Model 2B) display component, a standard or data entry keyboard, and the same optional magnetic slot reader and control available on the 3278 Display Station. Refer to the *IBM 3270 Information Display System Component Description*, GA27-2749, for detailed information on the 3279-2A and 3279-2B.

- The IBM 3287 Line Printer, Models 1 and 2, are compact, movable, tabletop printers. They are bidirectional wire matrix printers that can print up to 132 characters per line at up to 80 characters per second (Model 1) or up to 120 characters per second (Model 2). The printer is attached to an IBM 4701 controller via the Device Cluster Adapter (DCA) feature. Refer to the *IBM 3287 Printer Models 1 and 2 Component Description*, GA27-3153. for detailed information regarding the 3287 printers.

- The IBM 5210 Printer, which attaches to the 4700 device cluster adapter (DCA).

- The following devices and systems also attach to the 4700 Device Cluster Adapter (DCA) feature:

  - The IBM Personal Computer and Personal Computer/XT with the 3278/3279 Emulation Adapter, allowing the computer to emulate the 3278 U.S. English keyboard and 3278 Model 2 or the 3279 Model 2A Display Station.

  - IBM Displaywriter that supports the 3270 AW feature and 3274/3276 Dual Attachment (FC 8332). When operating in communications mode, the Displaywriter operates as a 3278 Model 2 Display Station, and the attached printers operate as IBM 3287 Model 1 or 2 printers.

  Refer to the appropriate device chapter later in this part of the manual for restrictions and differences from the normal operation when operating thes devices when linked to the 4701 controller.

Terminal Attachment Unit

The IBM 3603 Terminal Attachment Unit connects remote locations to the 4700 controller. The 3603 is designed for unattended operation and has no impact on programming support. The IBM 3603 Terminal Attachment Unit is not discussed elsewhere in this publication.

Device Cluster Adapter

The Device Cluster Adapter (DCA) attaches the following terminals to the 4700 controller: the 4704 Display Station Models 2 and 3, the IBM 3287 Printer Models 1 and 2, the IBM 3278 Keyboard/Display Model 2, the IBM 3279 Display Station Models 2A and 2B, and the IBM 3262 Line Printer.

Translation Table 5a References

The term "Table 5a" is used throughout this manual to refer to the *Data Processing National Use Graphics Table 5a*.

Chapter 6. IBM 4704 Display Station Model 1

The IBM 4704-1 Display (Figure 6-1) attaches to the banking loop and comprises one of two display monitors (the screen), one or two of four keyboards, and a power/logic control module. This display subsystem can also include a personal identification number (PIN) keypad and a magnetic stripe device.

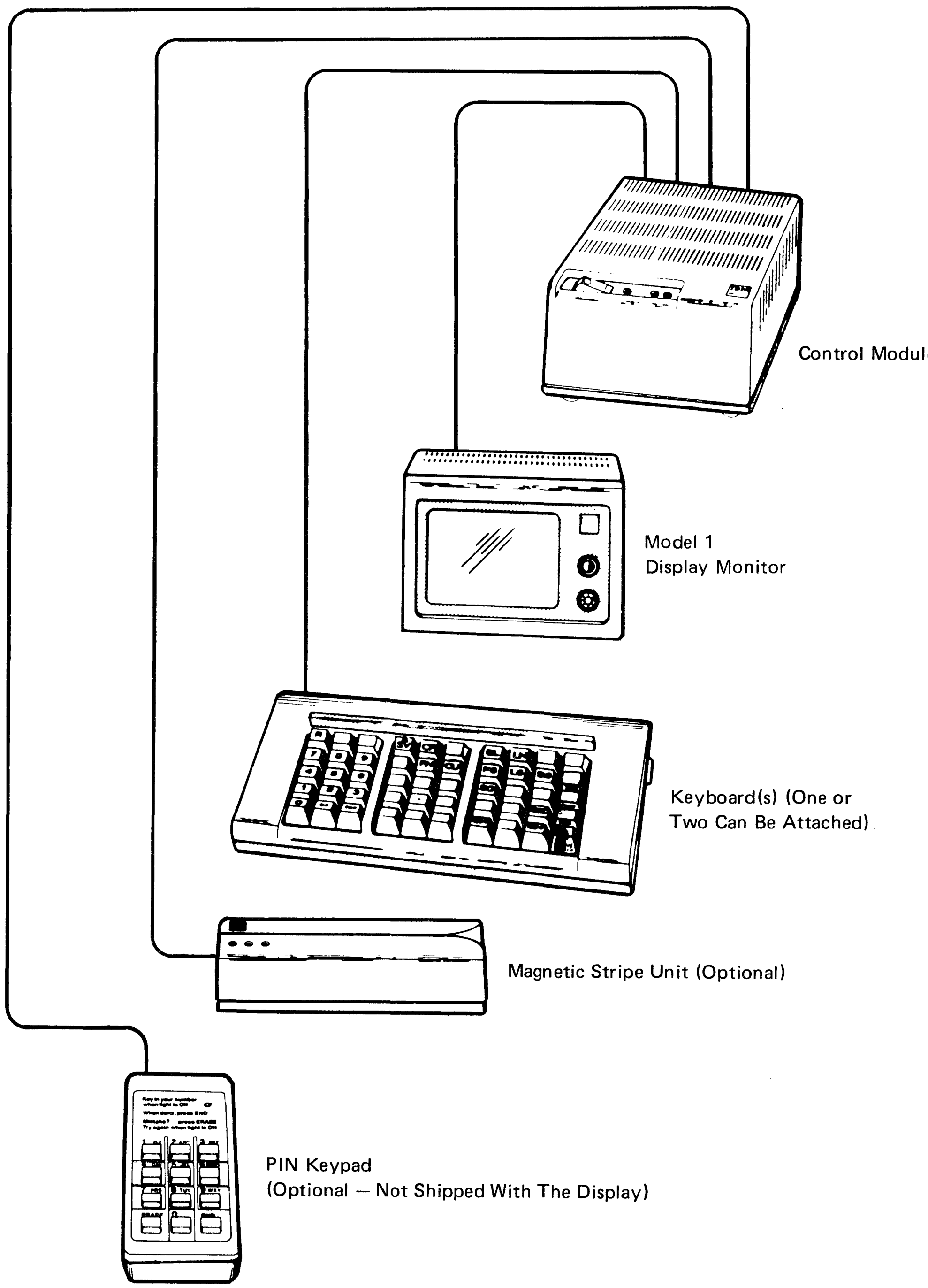


Figure 6-1. IBM 4704-1 Display Terminal

Functions and Features

The 4704 has:

- Two display screen sizes: the 140 mm (5.5 in.) screen can display 480 standard characters; the 230 mm (9 in.) screen can display 1920 standard characters or 480 large characters.
- Uppercase and lowercase characters
- Normal and intensified display characters
- Optional antiglare filters and support cradles
- Four keyboards: 50, 62, 77, and 107 keys
- A 112-key keyboard combination conprising the 62- and 50-key keyboards attached to the primary and secondary ports

Keyboard

The 50-key keyboard is a numeric/function keyboard. The 62-key keyboard (63 for Katakana) is a complete alphameric keyboard. The 77-key expanded alphameric keyboard (78 for Katakana) contains the entire 62-key keyboard along with 15 additional keys. The 107-key keyboard combines the 62-key alphameric keyboard with 15 control keys and 30 programmable keys.

Display Monitor

The 4704 display monitor is available as either a 480-character or 1920-character display screen. The display can present uppercase and lowercase characters in normal or intensified brilliance. The 480-character screen contains 40 vertical columns of 12 horizontal rows each. The 1920-character screen contains 80 columns of 24 rows each. The 140-mm (5.5-in) screen displays 480 characters only; the 230-mm (9.0-in) screen have either a 480- or 1920-character display, depending on its CPGEN definition.

Control Module

The display control module provides power and logic for all other components of the 4704-1 display station. Each component connects to the control module by a communication cable.

PIN Keypad

Each 4704 display station can have a normal or encrypted-PIN keypad, connected by cable to the control module, for users to enter their personal identification numbers.

Magnetic Stripe Device

Each 4704 display terminal can have a magnetic stripe reader or magnetic stripe reader/encoder, connected by a cable to the display control module.

Dual Intensity Capability

Dual intensity is controlled using the set attribute (X'28C1aa') display control character (where 'aa' is the desired attribute). An attribute of X'40' or X'C4' indicates normal intensity, an attribute of X'4C' indicates blanks (no display), and an attribute of 'C8' indicates high intensity. These attributes, when written to the display by the "set attribute" control character, have the following characteristics:

- They each occupy a display position.
- They are displayed as blanks (space).
- They cause the characters following their positions and continuing up to the next attribute character to be displayed with the specified intensity. Attribute control continues from the lower right-hand corner to the upper left-hand corner of the display.

For example; if the following data stream were written to the display:

$\text{X'}\underbrace{\text{0C}}_{1}\underbrace{\text{F1}}_{2}\underbrace{\text{28C1C8}}_{3}\underbrace{\text{F2F3}}_{4}\underbrace{\text{28C140}}_{5}\underbrace{\text{F4}}_{6}\text{'}$

the following would be displayed at the upper left-hand corner of the display screen.

Row \ Col	1	2	3	4	5	6
1	1		**2**	**3**		4
2						

Notes:

1. Causes the screen to be cleared (ensuring that attributes are not inadvertently left on the display).
2. Causes the character '1' to be displayed in column 1 of row 1 with normal intensity.
3. Causes the high intensity attribute byte to be sent to the display in column 2 of row 1. Note that the attribute is displayed as a blank.
4. Causes the characters '2' and '3' to be displayed in high intensity.
5. Causes the normal intensity attribute byte to be sent to the display in column 5 of row 1. Note that the attribute is displayed as a blank. All display positions following this attribute will be generated at normal intensity; this includes the character '1' in column 1 of row 1 (because of the attribute control wrapping).
6. Causes the character '4' to be displayed in column 6 of row 1 in normal intensity.

Controls and Indicators

Display Control Module Controls and Indicators

Power On/Off Switch

This two-position switch on the control module controls the electrical power to the other display station components. When this switch is set ON, power is applied to all units attached to the display control module, and the subsystem performs initial tests.

Power-On Lamp

This indicator, a red light on early models of the control module, is on whenever power is applied.

Loop Ready Lamp

This indicator, a green light on early models of the control module, is on whenever the loop and all powered-on terminals on the loop are capable of sending and receiving signals between the terminal and the controller.

Test Lamp

This indicator, a yellow light on the control module, is on during the automatic tests that are performed when the terminal is switched on. When testing is completed, this indicator remains on only if an error occurs.

Loop Address Switches

Set these four switches to a binary number (from 0 to 15, where the binary number 0 equates to address 16) representing the base loop address assigned to the device. The device may share this base address with another terminal, such as a 4710 printer, that does not have the same set of subaddresses (1, 2, or 3). If you change the address, you must turn the power off and then on again.

Loop Speed Switches

Set these switches for operating at 600, 1200, 2400, or 4800 bits per second. The 4701 controller does not support 600 bps.

Keyboard Alarm Control and PIN Keypad Light

Audible Alarm

This control adjusts the volume of the audible alarm. It is on the underside of the keyboard.

PIN Keypad Light

This red lamp indicates that keypad data can be entered. When the indicator is on, all other keyboard input except the Reset key is disabled.

Display Monitor Controls and Indicators

Contrast Control

This control, on the display monitor, adjusts the intensity difference between the high- and the normal-intensity characters.

Brightness Control

This control on the display monitor adjusts the brightness of all displayed characters.

Operation Indicators

These indicators are symbols displayed at the bottom of the display monitor's screen:

Not Operational: This indicator is displayed when one of the power-on tests fails. A two-digit error code is displayed next to this indicator.

MSE Ready to Encode: Encoding data has been sent to this terminal, and the magnetic stripe encoder is ready to encode data. This indicator is removed when a magnetic-striped card is passed through the encoder.

MSR/E Check: This indicator is displayed when an error occurs at the magnetic stripe reader/encoder.

Too Much Data: The terminal's buffer is full. The terminal's buffer contains 64 bytes. When the buffer contains 63 bytes, pressing the next key switches this indicator on; no further keys are read until the contents of the buffer are transmitted to the controller. The keyboard is in Purge mode until the operator presses the Reset key.

PIN Keypad Enabled: This symbol indicates that the PIN keypad is enabled, and all other terminal input (except the Reset key) is disabled.

User-Programmable Indicators: Three user programmable indicators and a System Check indicator are available. These indicators are controlled by the SIGNAL instruction in the application program. If a PIN keypad is attached and active, you control it by switching indicator 3 on and off with the SIGNAL instruction. Refer to "Personal Identification Number (PIN) Keypad" on page 6-7 for detailed information on how this indicator controls the PIN keypad.

Magnetic Stripe Reader and Reader/Encoder

The International Standard ISO 3554 specifies the physical and magnetic characteristics for a magnetic stripe. The stripe can have three tracks. The magnetic stripe reader reads the data on track 2 (the ABA track) on a credit or ID card encoded at 75 bpi. Either the magnetic stripe reader/encoder or 75/210 bpi magnetic stripe reader reads 75-bpi or 210-bpi track 2 data in accordance with the 4-bit IBM Specifications code. The magnetic stripe reader/encoder encodes IBM 210 bpi data on track 2 of a magnetic stripe. ABA 75 bpi data cannot be encoded. The reader and the reader/encoder are connected to the display control module by a communication cable.

The readers and reader/encoders have three lights to inform the operator of various conditions. Two indicators on the display screen are associated with operation of the magnetic devices. These are MSE READY to ENCODE, and MSR/E CHECK. Both of these indicators cause an audible tone when they are set on. The operation of these indicators is described below.

Read Operation

During a read operation, the indicators are:

GREEN
Ready to read. The indicator is on whenever the magnetic device is ready to read. The indicator turns off after the magnetic medium is moved through the reader, at which time the yellow indicator comes on.

YELLOW
In process, wait. This indicator comes on after the medium is moved through the reader; magnetic data is pending transmission to the controller. After the data is transmitted, this indicator goes off and the green indicator returns. However, if an error is detected, the red indicator comes on with the green indicator.

RED
Error detected, retry. If a read error occurs (no data detected, parity or longitudinal redundancy check error), the red indicator comes on (along with the green). This indicates an error has occurred. The red indicator remains on until either a good read occurs, or the operator presses the Reset key while the Alternate key is held down.

For the data to be transmitted to the controller, the application program must issue an LREAD. After the controller has selected the device for the read, input can occur from the keyboard, PIN keypad, and magnetic device in any sequence. However, only one device can operate at a time.

Encode Operation

During encoding, the indicators are:

GREEN
Flashing -- encoder enabled; ready to encode. The indicator is normally on steadily to indicate that a read can occur. When the controller enables the encoder to encode a document, the indicator starts to flash 2 or 3 times per second. The magnetic medium may now be passed through the slot to encode. When this operation is started, the green indicator is turned off, and the yellow indicator is turned on.

YELLOW
In process; wait. This indicator is turned on immediately after a medium is passed through the encoding slot. It stays on until after the read-back check is complete, or for 0.5 second, whichever is longer. If the read-back check is good, the yellow indicator is turned off, and the steady green indicator returns. If the read-back check fails, the yellow indicator is turned off, and the red indicator is turned on and the green indicator resumes flashing.

RED

Error detected; retry. The red indicator signifies an error during the encoding operation. It stays on until one of the following events occur:

- A subsequent good encode occurs.
- Local Reset occurs (press Reset while holding Alt).
- Programmed disable (described below) is issued.

After any of these conditions occur, the green indicator returns.

Programmed Disable

The encoder can be disabled either by a command reset or by another LWRITE before the first encode is completed. This condition is recognized by the 4704, which disables the encoder and sets the "Encode Error" status bit on and all other bits off in the status byte.

Personal Identification Number (PIN) Keypad

The PIN keypad consists of ten data keys, two function keys, and a red "enter PIN" indicator. The keypad attaches by cable to the 4704-1 Display Control Module or directly to the 4704-2 or 4704-3 display. An encrypting keypad has one additional switch to enable an authorized operator to enter an encryption key value.

The PIN keypad operates as an extension of the keyboard. The application program enables the PIN keypad by setting on programmable indicator 3 using the SIGNAL instruction. While the PIN keypad is enabled, both programmable indicator 3 on the display screen and the red lamp on the keypad are on.

The PIN Keypad can be disabled by the:

1. End key on the keypad
2. Reset key on the keyboard
3. Application program, which issues SIGNAL to switch programmable indicator 3 off before the End or Reset key is pressed.

When the PIN Pad is disabled by the End or Reset key, the indicator on the keypad, and programmable indicator 3 on the display are turned off immediately. A X'7F' is placed in the data stream signifying end of PIN data. The controller recognizes the X'7F' and turns programmable indicator 3 off.

The PIN keypad port accommodates two types of keypads: a clear and an encrypted pad. When the keypad is selected as described above, the 4704 transmits to the controller a X'7F' for the clear pad, and a X'7E' for the encrypted pad to signify "start of PIN field".

As keys are pressed on the clear keypad, their scan codes are passed to the controller. After an input error, the user can press the Erase key on the pad. This inserts a X'0B' in the data stream to tell the controller to ignore the preceding data. The entry can then be restarted.

When the PIN is complete, pressing the End key places X'7F' in the data stream to mark the end of the PIN. The PIN indicator on the display goes off, and the keypad is de-selected. The indicator on the keypad goes off. The terminal operator can use the PIN field by pressing the Reset key in lieu of pressing the End key on the pad. This results in the same action as the End key. The application program can end the PIN operation before the operator by turning off programmable indicator 3. This deselects the PIN pad.

If the keypad is encrypted, data is not transmitted from the keypad to the terminal until the End key on the keypad is pressed. In this case, the controller accepts 8 characters of encrypted data plus the X'7F' for the END key when the End key is pressed. The controller then turns off the PIN indicator and the keypad indicator, and the PIN keypad is deselected. If the operator presses the Reset key instead of the End key, no data is transmitted from the pad, and the keypad is deselected. A X'7F' is transmitted, terminating input to the controller. After an input error, the customer can press the Erase key on the pad, and reenter the PIN.

Terminal Control Characters

Control characters are embedded in the data stream and indicate operations to be performed at a terminal. The following are the control characters for the 4704:

Function:	Control Character:	Explanation:
Line Feed	X'25'	The cursor is moved down one line; the horizontal position is unchanged.
Return	X'0D'	The cursor moves to position 1 on the current display line.
New Line	X'15'	The current line is erased from the cursor to the end of the line. The cursor moves to position 1 of the next lower line.
New Page	X'0C'	The screen is completely erased; the cursor is moved to position 1 of line 1. If there is a magnetic stripe encoder (MSE) on the 4704 that is ready to use, a write to the encoder with a first byte of X'0C' in the message causes the encoder to become not ready. Reset should not be issued for a MSE that is not ready to use (not in the encode state).
Set Attribute	X'28'	This 3-byte control string defines the following field characteristics and sends one byte to the display. The second byte should be X'C1'. The third byte may be one of these values: X'40' Normal intensity X'C4' Normal intensity X'C8' Intensified X'4C' Nondisplay

Function:	Control Character:	Explanation:
Position	X'34'	This control character is followed by a flag byte and a 1-byte positioning value. The second byte is the flag byte: Bits 0-3 are ignored. Bit 4: 0 = Position is an absolute value. 1 = Position is relative to the present position. Bit 5: 0 = Position change is horizontal. 1 = Position change is vertical. Bits 6-7: 00 = Move cursor to new position without erasing. 01 = Leave cursor at the current position, but erase from the current position the number of bytes specified by the position. 10 = Move cursor to new position and erase from the old position up to, but not including, the new position. 11 = Reserved. The third byte controls position. The positioning byte indicates a line or column number. It contains an unsigned binary value ranging from 0 to 255 to be used under direction of the flag byte; a 0 results in no operation. The value specified is divided by a modulus number to determine the resulting position. The modulus division produces a change of 1 to x lines for vertical positioning, and a change of 1 to y columns for horizontal positioning, where x is the maximum number of lines on a screen, and y is the maximum number of columns in a line.
Select	X'04'	If OPTIONS=9641 is not coded on DEV4704, the 4704 skips the byte following this control character and suppresses the automatic new–line action. If OPTIONS=9641 is specified, the next byte selects direction: X'01' for backward mode, X'00' for forward mode.

Function:	Control Character:	Explanation:
Transparent Write	X'35'	A transparent write (a write without translation) begins. The format of this control sequence is: Byte 0: The control character, X'35'. Byte 1: The number of data bytes to be displayed. Byte 2-Byte n: The transparent data to be displayed. The valid positions are X'00' - X'FF'.

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or they may be restarted, or "wrapped back," to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 4704:

Keyboard

Counter:	Explanation:
1	Loop error checks
2	Terminal address card unit checks
3	Device errors (keyboard or magnetic stripe reader)
4	Translation checks
5	Segment overruns

Display

Counter:	Explanation:
1	Loop error checks
2	Terminal address card unit checks

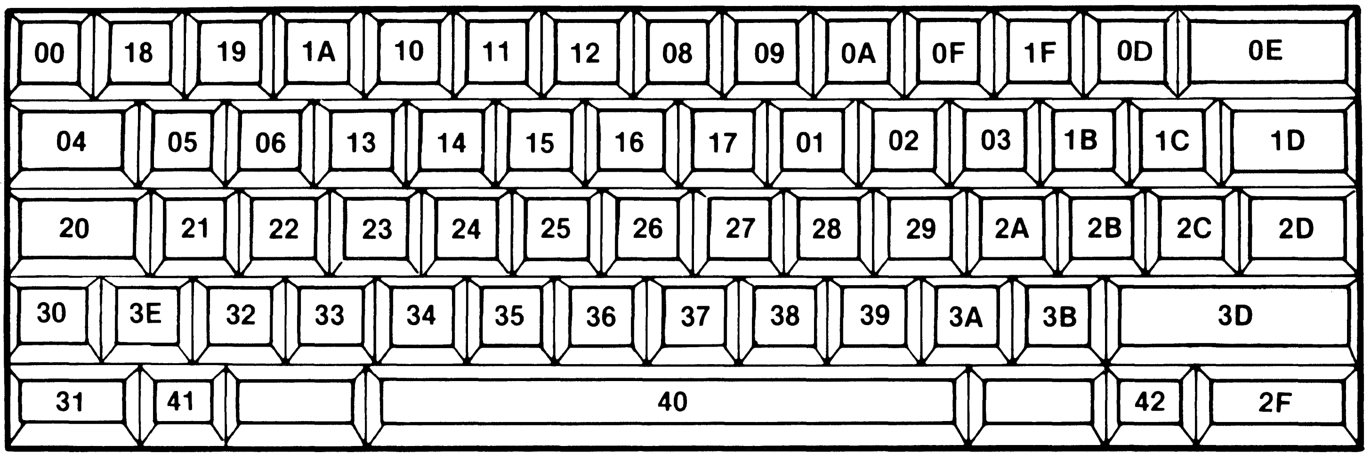
Magnetic Stripe Reader/Encoder

Counter:	Explanation:
1	Loop error checks
2	Terminal address card unit checks
3	Device error (format error or encoding failure)

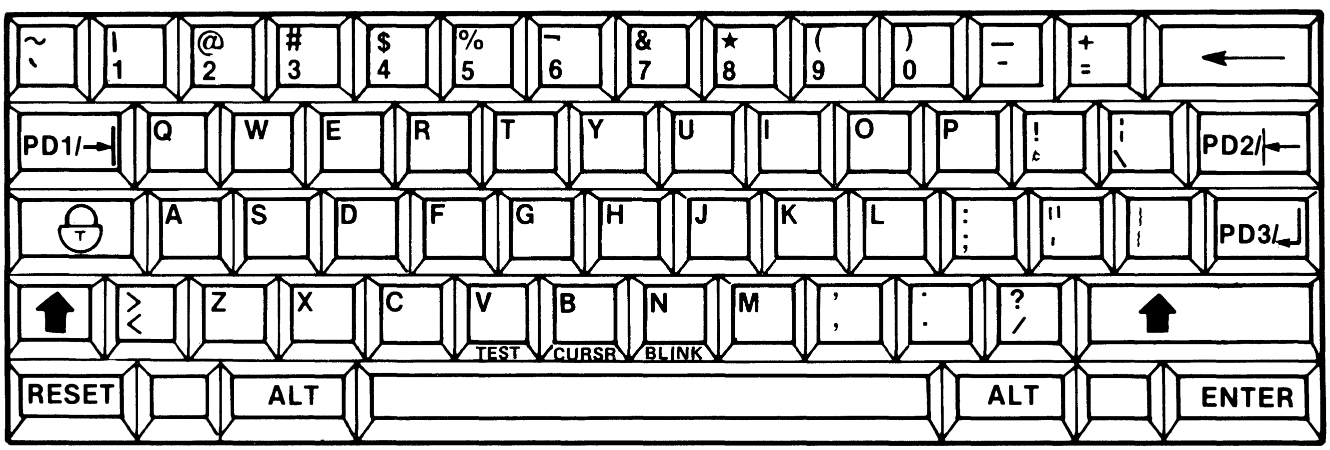
Keyboards and Translation Tables

The following pages contain the 4704 keyboards and the standard input and output translation tables for them.

4704 62-Key Keyboard (47US62)



Scan Codes

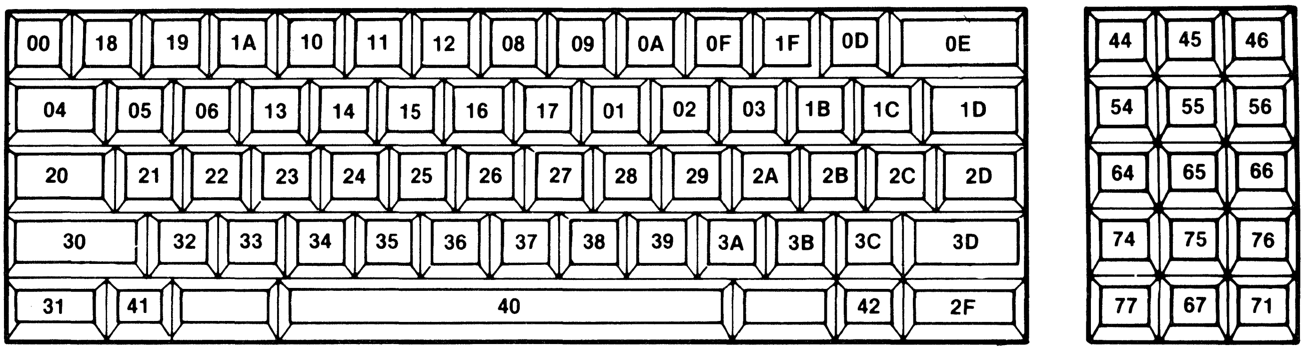


Engraved Keys

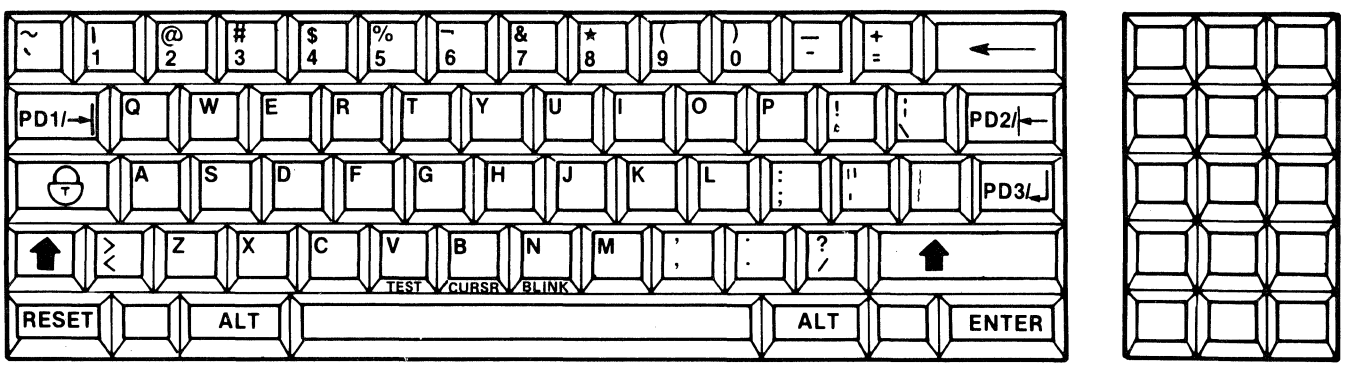
Modifying Translation Tables

The following tables show the character positions for each graphic character in the 4704 character set. For each character you can determine the applicable output position code from the matrix. Use this position code with the hexadecimal character input value to modify a translation table.

4704 77-Key Keyboard (47US77)



Scan Codes

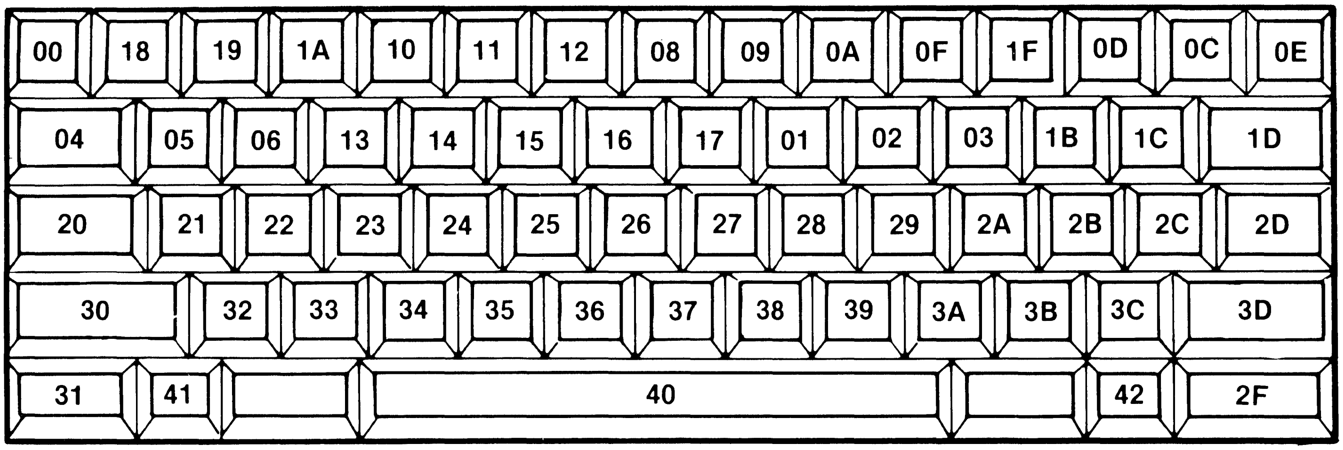


Engraved Keys

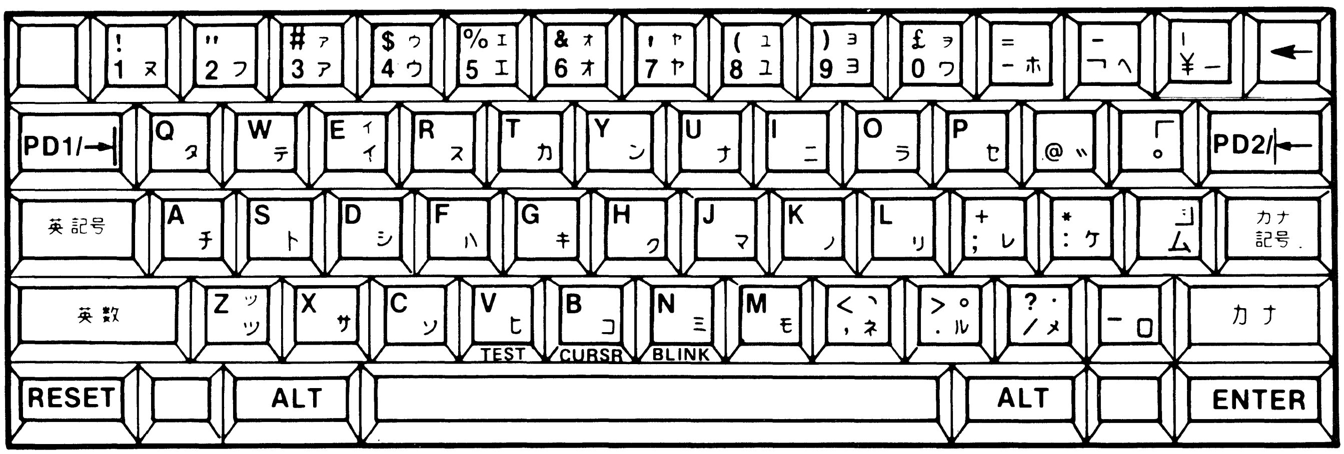
To use this value in the configuration process, convert the two-digit hexadecimal position code (X'54', the upper case T, for example) to decimal (84) for use in the OUTRTBL macro. This decimal value is used in the POS operand. Then, code the hexadecimal equivalent of the graphic character (X'E3' for the character T) in X'n' operand.

The graphics for position codes X'C0' through X'FF' depend on the character set selected, and are shown in the next matrix. Codes X'80' through X'AF' are treated as no-operation codes; X'B0' through X'BF' are for display orders.

4704 Katakana 63-Key Keyboard (47KAT62)



Scan Codes



Engraved Keys

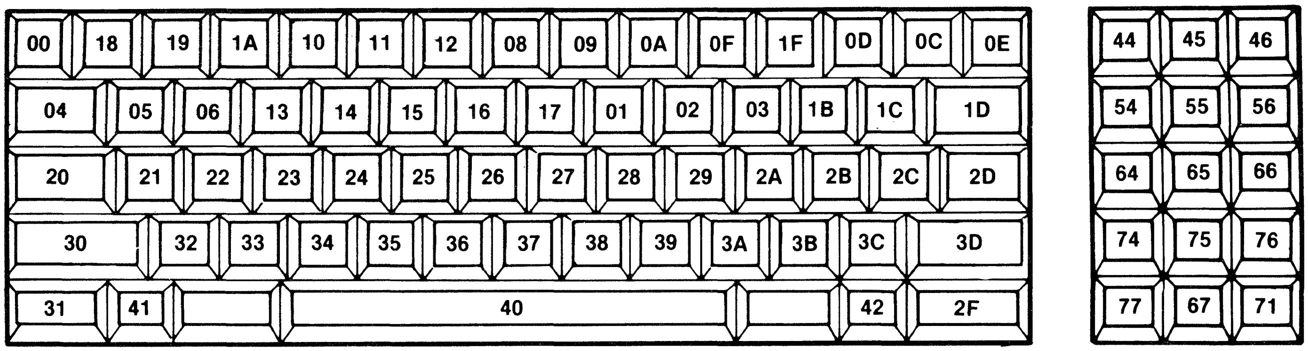
Defining The 4704-1 for Configuration

Controller configuration macro instructions are used to specify the physical and logical configuration of each 4701 and its associated terminals. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. This section contains specific notes for configuring the 4704. Refer to Volume 6 for a description and information on coding the DEV4704 configuration macro.

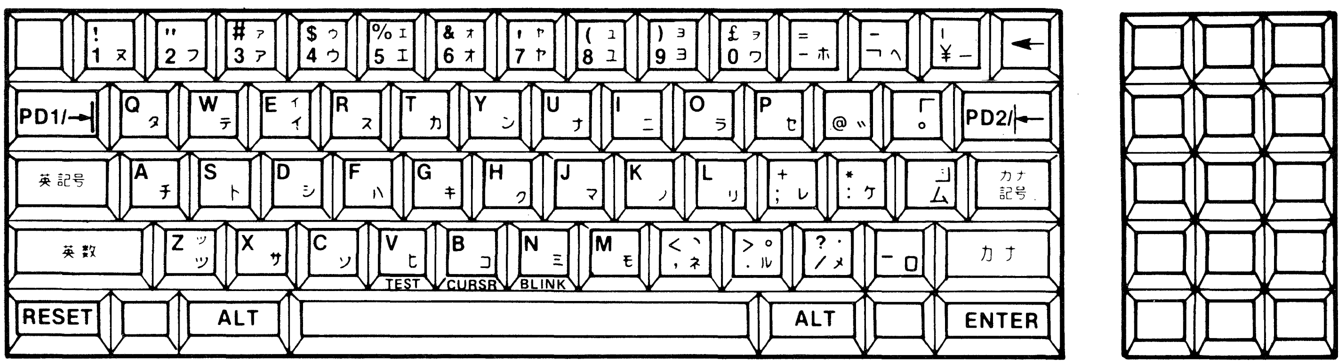
There is no MSLITE operand on DEV4704. The 4704 magnetic device indicator lights are controlled by the terminal rather than by the program.

If you are converting from a DEV3604 macro, be sure to change the TYPA and SHFT operands of the TRTBHDR macro. The 3604 scan codes are not the same.

4704 Katakana 77-Key Keyboard (47KAT77)

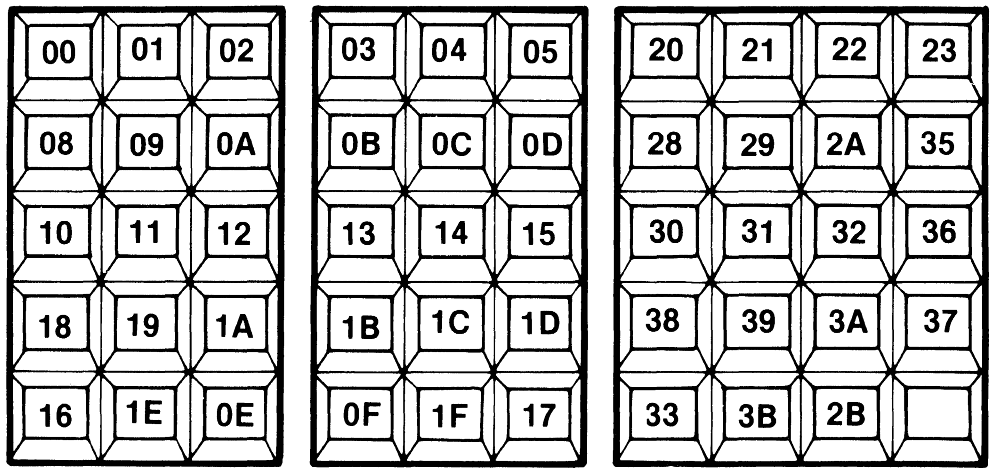


Scan Codes

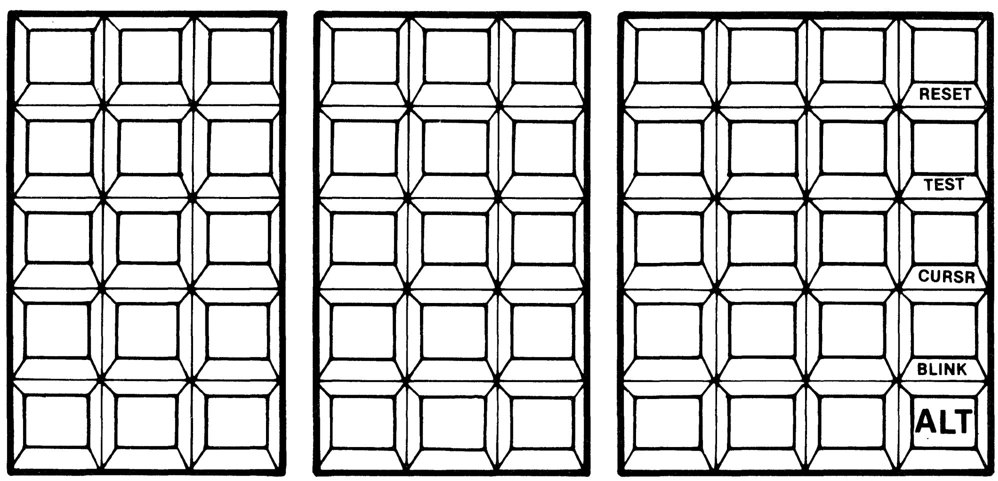


Engraved Keys

4704 50-Key Function Keyboard (470450) On Primary Port

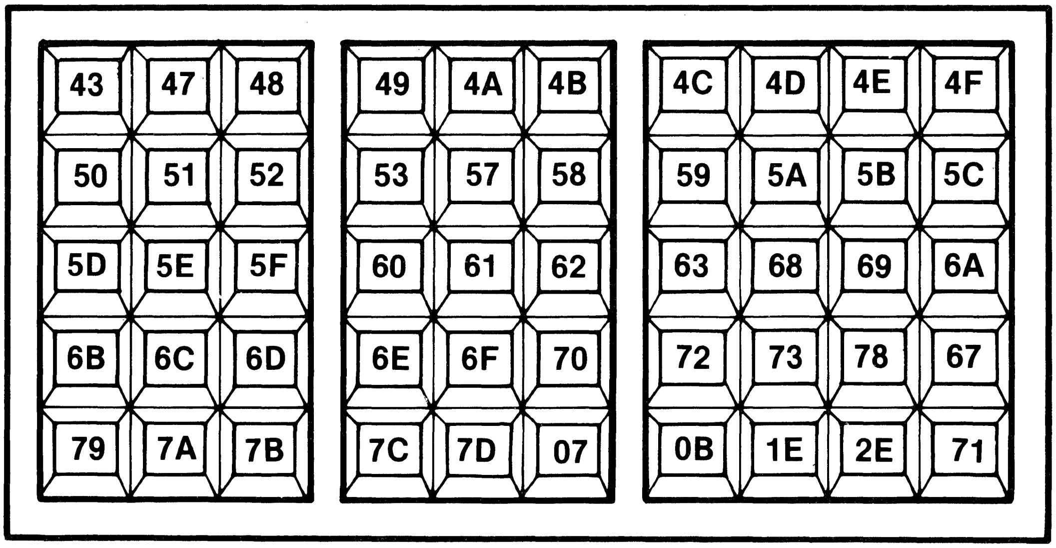


Scan Codes



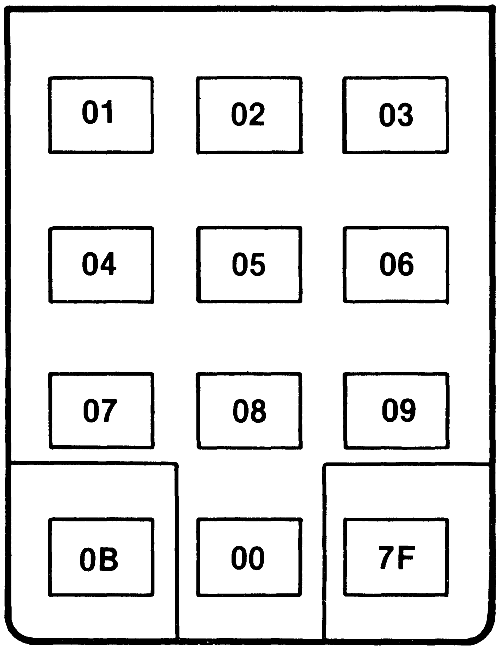
Engraved Keys

4704 50-Key Function Keyboard (470450) Attached to Secondary Port:

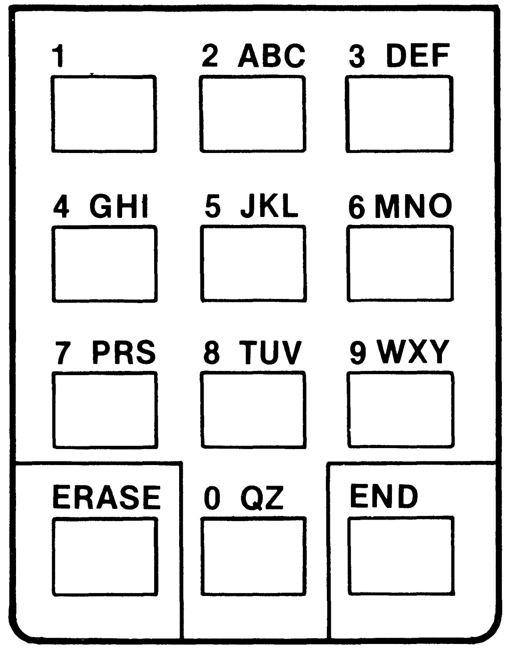


Note: You cannot define transient shift, downshift, or repeat action keys for keyboards attached to the secondary port.

4704 PIN Keypad Device

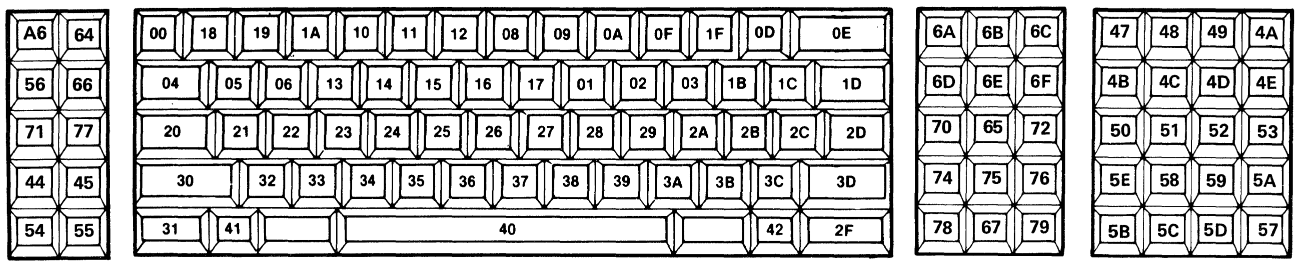


Scan Codes

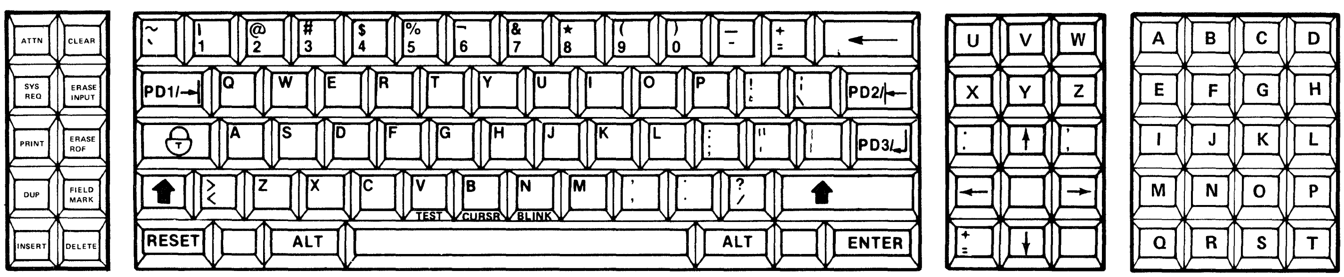


Keyboard Overlay

4704 Administrative Keyboard:



Scan Codes



Keyboard

SCAN CODE	CASE 1 HEX	OUTPUT CODE	CHAR	CASE 2 HEX	OUTPUT CODE	CHAR
00	79	1A	`	A1	1C	~
01	89	69	i	C9	49	I
02	96	6F	o	D6	4F	O
03	97	70	p	D7	50	P
04			PD1			PD1
05	98	71	q	D8	51	Q
06	A6	77	w	E6	57	W
07	NOT USED					
08	F7	37	7	50	26	&
09	F8	38	8	5C	2A	*
0A	F9	39	9	4D	2B	(
0B	NOT USED					
0C	NOT USED					
0D	7E	3D	=	4E	2B	+
0E			BS			BS
0F	F0	30	0	5D	29	)
10	F4	34	4	5B	24	$
11	F5	35	5	6C	25	%
12	F6	36	6	5F	11	¬
13	85	65	e	C5	45	E
14	99	72	r	D9	52	R
15	A3	74	t	E3	54	T
16	A8	79	y	E8	59	Y
17	A4	75	u	E4	55	U
18	F1	31	1	4F	10	|
19	F2	32	2	7C	40	@
1A	F3	33	3	7B	23	#
1B	4A	08	¢	5A	21	?
1C	E0	5C	\	6A	1E	¦
1D		PD2				PD2
1E	NOT USED					
1F	60	2D	-	6D	5F	—
20	SHIFT LOCK					
21	81	61	a	C1	41	A
22	A2	73	s	E2	53	S
23	84	64	d	C4	44	D
24	86	66	f	C6	46	F
25	87	67	g	C7	47	G
26	88	68	h	C8	48	H

4704 U.S. English Character Set

SCAN CODE	CASE 1 HEX	OUTPUT CODE	CHAR	CASE 2 HEX	OUTPUT CODE	CHAR
27	91	6A	j	D1	4A	J
28	92	6B	k	D2	48	K
29	93	6C	l	D3	4C	L
2A	5E	3B	;	7A	3A	:
2B	7D	27	'	7F	22	"
2C	C0	7B	{	D0	7D	}
2D		PD3				PD3
2E	NOT USED					
2F	ENTER					
30	SHIFT KEY					
31	RESET					
32	A9	7A	z	E9	5A	Z
33	A7	78	x	E7	58	X
34	83	63	c	C3	43	C
35	A5	76	v	E5	56	V
36	82	62	b	C2	42	B
37	95	6E	n	D5	4E	N
38	94	6D	m	D4	4D	M
39	6B	2C	,	6B	2C	,
3A	4B	2E	.	4B	2E	.
3B	61	2F	/	6F	3F	?
3C	NOT USED					
3D	SHIFT KEY					
3E	4C	3C	<	6E	3E	>
3F	ALT KEY					
40	40	01	SPACE	40	01	SPACE
41			bl			bl
42			bl			bl

4704 U.S. English Character Set

SCAN CODE	CASE 1			CASE 2			CASE 3			CASE 4		
	HEX	OUTPUT CODE	CHAR	HEX	OUTPUT CODE	CHAR	HEX	OUTPUT CODE	CHAR	HEX	OUTPUT CODE	CHAR
00	●	●	bl	●	●	bl	●	●	bl	●	●	bl
18	F1	31	1	5A	21	!	98	D6	ヌ	●	●	bl
19	F2	32	2	7F	22	"	9F	DB	フ	●	●	bl
1A	F3	33	3	7B	23	#	81	C0	ア	47	F5	ァ
10	F4	34	4	E0	24	$	83	C2	ウ	49	F7	ゥ
11	F5	35	5	6C	25	%	84	C3	エ	51	F8	ェ
12	F6	36	6	50	26	&	85	C4	オ	52	F9	ォ
08	F7	37	7	7D	27	'	A9	E3	ヤ	53	FA	ャ
09	F8	38	8	4D	28	(	AA	E4	ユ	54	FB	ュ
0A	F9	39	9	5D	29	)	AC	E5	ヨ	55	FC	ョ
0F	F0	30	0	4A	1D	£	BC	EB	ワ	46	F4	ヲ
1F	60	1D	–	7E	3D	=	A3	DD	ホ	●	●	bl
0D	5F	11	¬	A1	7F	‾	A2	DC	ヘ	●	●	bl
0C	5B	0F	¥	4F	10	|	58	FE	ー	●	●	bl
0E			← BS			← BS			← BS			← BS
04			PD1			PD1			PD1			PD1
05	D8	51	Q	●	●	bl	91	CF	タ	●	●	bl
06	E6	57	W	●	●	bl	94	D2	テ	●	●	bl
13	C5	45	E	●	●	bl	82	C1	イ	48	F6	ィ
14	D9	52	R	●	●	bl	8E	CC	ス	●	●	bl
15	E3	54	T	●	●	bl	86	C5	カ	●	●	bl
16	E8	59	Y	●	●	bl	8D	EC	ン	●	●	bl
17	E4	55	U	●	●	bl	96	D4	ナ	●	●	bl
01	C9	49	I	●	●	bl	97	D5	ニ	●	●	bl
02	D6	4F	O	●	●	bl	AD	E6	ラ	●	●	bl
03	D7	50	P	●	●	bl	8F	CD	セ	●	●	bl
1B	7C	40	@	●	●	bl	BE	ED	゛	●	●	bl
1C	●	●	bl	●	●	bl	BF	EE	゜	42	F0	「
1D			PD2			PD2			PD2			PD2
20	UPPER CASE LATIN SHIFT KEY											
21	C1	41	A	●	●	bl	92	D0	チ	●	●	bl
22	E2	53	S	●	●	bl	95	D3	ト	●	●	bl
23	C4	44	D	●	●	bl	8D	CB	シ	●	●	bl
24	C6	46	F	●	●	bl	9D	D9	ハ	●	●	bl
25	C7	47	G	●	●	bl	87	C6	キ	●	●	bl
26	C8	48	H	●	●	bl	88	C7	ク	●	●	bl
27	D1	4A	J	●	●	bl	A4	DE	マ	●	●	bl
28	D2	4B	K	●	●	bl	9A	D8	ノ	●	●	bl

SCAN CODE	CASE 1 HEX	CASE 1 OUTPUT CODE	CASE 1 CHAR	CASE 2 HEX	CASE 2 OUTPUT CODE	CASE 2 CHAR	CASE 3 HEX	CASE 3 OUTPUT CODE	CASE 3 CHAR	CASE 4 HEX	CASE 4 OUTPUT CODE	CASE 4 CHAR
29	D3	4C	L	●	●	bl	AE	E7	ﾘ	●	●	bl
2A	5E	3B	;	4E	2B	+	BA	E9	ﾚ	●	●	bl
2B	7A	3A	:	5C	2A	*	89	C8	ｹ	●	●	bl
2C	●	●	bl	●	●	bl	A6	E0	ﾑ	43	F1	｣
2D	UPPER CASE KATAKANA SHIFT KEY											
30	30 AND 3E IS LOWER CASE LATIN SHIFT KEY											
3E	NOT USED											
32	E9	5A	Z	●	●	bl	93	D1	ﾂ	56	FD	
33	E7	5B	X	●	●	bl	8C	CA	ｻ	●	●	bl
34	C3	43	C	●	●	bl	90	CE	ｿ	●	●	bl
35	E5	56	V	●	●	bl	9E	DA	ﾋ	●	●	bl
36	C2	42	B	●	●	bl	8A	C9	ｺ	●	●	bl
37	D5	4E	N	●	●	bl	A5	DF	ﾐ	●	●	bl
38	D4	40	M	●	●	bl	A8	E2	ﾓ	●	●	bl
39	6B	2C	,	4C	3C	<	99	D7	ﾈ	44	F2	､
3A	4B	2E	.	6E	3E	>	AF	E8	ﾙ	41	EF	｡
3B	61	2F	/	6F	3F	?	A7	E1	ﾒ	45	F3	･
3C	6D	5F	—	●	●	bl	BB	EA	ﾛ	●	●	bl
3D	LOWER CASE KATAKANA SHIFT KEY											
31			RESET			RESET			RESET			RESET
41	●	●	bl	●	●	bl	●	●	bl	●	●	bl
3F			ALT			ALT			ALT			ALT
40	40	01	SPACE	40	01	SPACE	40	01	SPACE	40	01	SPACE
3F			ALT			ALT			ALT			ALT
42	●	●	bl	●	●	bl	●	●	bl	●	●	bl
2F			ENTER			ENTER			ENTER			ENTER

4704EB EBCDIC Output Translation Table

This table shows the output translation table.

This table shows 4704 display output positions determined by EBCDIC characters in the user output data stream. All values are in hexadecimal. The table is in order by output positions. Positions 80 - BF are reserved. The characters displayed for positions C0-FE depend on the 'CHARSET' value coded in the DEV4704 macro, that is, EB1, EB2, EB3, or EB4.

As an example of table usage, EBCDIC X'C1' selects output position HEX 41. In decimal this is position 65; when using the 'OUTRTBL' macro all positions must be coded in decimal. The display will show a capital letter 'A'. To change the selection, code OUTRTBL (66,X'C1') ... now X'C1' selects Decimal Position 66 or hex Position 42. This is displayed as a capital 'B'.

HEX OUTPUT POSITION - | | -EBCDIC HEX

00 - ..	40 - 7C	C0 - 44
01 - 40	41 - C1	C1 - 54
02 - ..	42 - C2	C2 - 58
03 - ..	43 - C3	C3 - CD
04 - ..	44 - C4	C4 - DD
05 - ..	45 - C5	C5 - 46
06 - ..	46 - C6	C6 - CF
07 - ..	47 - C7	C7 - 43
08 - B0	48 - C8	C8 - 53
09 - ..	49 - C9	C9 - 57
0A - ..	4A - D1	CA - CC
0B - B3	4B - D2	CB - DC
0C - ..	4C - D3	CC - 42
0D - ..	4D - D4	CD - 52
0E - B8	4E - D5	CE - 56
0F - B2	4F - D6	CF - CB
10 - BB	50 - D7	D0 - 64
11 - BA	51 - D8	D1 - 74
12 - ..	52 - D9	D2 - 8C
13 - BE	53 - E2	D3 - ..
14 - ..	54 - E3	D4 - FD
15 - ..	55 - E4	D5 - 66
16 - ..	56 - E5	D6 - EF
17 - B5	57 - E6	D7 - 63
18 - AC	58 - E7	D8 - 73
19 - ..	59 - E8	D9 - 77
1A - 79	5A - E9	DA - EC
1B - 90	5B - 4A	DB - FC
1C - A1	5C - E0	DC - 62
1D - B1	5D - 5A	DD - 72
1E - 6A	5E - 5F	DE - 76
1F - BD	5F - 6D	DF - EB
20 - ..	60 - ..	E0 - DB
21 - 4F	61 - 81	E1 - 45
22 - 7F	62 - 82	E2 - 51
23 - 7B	63 - 83	E3 - 55
24 - 5B	64 - 84	E4 - CE

Output Translation Table (continued)

| -HEX OUTPUT POSITION

| -EBCDIC HEX

25 - 6C	65 - 85	E5 - DE
26 - 50	66 - 86	E6 - 49
27 - 7D	67 - 87	E7 - 47
28 - 4D	68 - 88	E8 - 48
29 - 5D	69 - 89	E9 - 70
2A - 5C	6A - 91	EA - 8D
2B - 4E	6B - 92	EB - 9C
2C - 6B	6C - 93	EC - 8E
2D - 60	6D - 94	ED - AE
2E - 4B	6E - 95	EE - 9D
2F - 61	6F - 96	EF - A0
30 - F0	70 - 97	F0 - FB
31 - F1	71 - 98	F1 - 65
32 - F2	72 - 99	F2 - 71
33 - F3	73 - A2	F3 - 75
34 - F4	74 - A3	F4 - EE
35 - F5	75 - A4	F5 - FE
36 - F6	76 - A5	F6 - 69
37 - F7	77 - A6	F7 - 67
38 - F8	78 - A7	F8 - 68
39 - F9	79 - A8	F9 - 80
3A - 7A	7A - A9	FA - AD
3B - 5E	7B - C0	FB - 9E
3C - 4C	7C - ..	FC - 9F
3D - 7E	7D - D0	FD - 59
3E - 6E	7E - ..	FE - AB
3F - 6F	7F - BC	

Output Codes Listed by Hexadecimal Value

This table shows the output position codes in order of EBCDIC hexadecimal value.

| -HEX OUTPUT POSITION
| | -EBCDIC HEX

01 - 40	1A - 79	0F - B2	.. - ED
.. - 41	3A - 7A	0B - B3	F4 - EE
CC - 42	23 - 7B	.. - B4	D6 - EF
C7 - 43	40 - 7C	17 - B5	30 - F0
C0 - 44	27 - 7D	.. - B6	31 - F1
E1 - 45	3D - 7E	.. - B7	32 - F2
C5 - 46	22 - 7F	0E - B8	33 - F3
E7 - 47	F9 - 80	.. - B9	34 - F4
E8 - 48	61 - 81	11 - BA	35 - F5
E6 - 49	62 - 82	10 - BB	36 - F6
5B - 4A	63 - 83	7F - BC	37 - F7
2E - 4B	64 - 84	1F - BD	38 - F8
3C - 4C	65 - 85	13 - BE	39 - F9
28 - 4D	66 - 86	.. - BF	.. - FA
2B - 4E	67 - 87	7B - C0	F0 - FB
21 - 4F	68 - 88	41 - C1	DB - FC
26 - 50	69 - 89	42 - C2	D4 - FD
E2 - 51	.. - 8A	43 - C3	F5 - FE
CD - 52	.. - 8B	44 - C4	.. - FF
C8 - 53	D2 - 8C	45 - C5	
C1 - 54	EA - 8D	46 - C6	
E3 - 55	EC - 8E	47 - C7	
CE - 56	.. - 8F	48 - C8	
C9 - 57	1B - 90	49 - C9	
C2 - 58	6A - 91	.. - CA	
FD - 59	6B - 92	CF - CB	
5D - 5A	6C - 93	CA - CC	
24 - 5B	6D - 94	C3 - CD	
2A - 5C	6E - 95	E4 - CE	
29 - 5D	6F - 96	C6 - CF	
3B - 5E	70 - 97	7D - D0	
5E - 5F	71 - 98	4A - D1	
2D - 60	72 - 99	4B - D2	
2F - 61	.. - 9A	4C - D3	
DC - 62	.. - 9B	4D - D4	
D7 - 63	EB - 9C	4E - D5	
D0 - 64	EE - 9D	4F - D6	
F1 - 65	FB - 9E	50 - D7	
D5 - 66	FC - 9F	51 - D8	
F7 - 67	EF - A0	52 - D9	
F8 - 68	1C - A1	.. - DA	
F6 - 69	73 - A2	E0 - DB	
1E - 6A	74 - A3	CB - DC	
2C - 6B	75 - A4	C4 - DD	
25 - 6C	76 - A5	E5 - DE	
5F - 6D	77 - A6	.. - DF	
3E - 6E	78 - A7	5C - E0	
3F - 6F	79 - A8	.. - E1	
E9 - 70	7A - A9	53 - E2	
F2 - 71	.. - AA	54 - E3	
DD - 72	FE - AB	55 - E4	
D8 - 73	18 - AC	56 - E5	
D1 - 74	FA - AD	57 - E6	
F3 - 75	ED - AE	58 - E7	
DE - 76	.. - AF	59 - E8	
D9 - 77	08 - B0	5A - E9	
.. - 78	1D - B1	.. - EA	
		DF - EB	
		DA - EC	

Display Graphics Translation Table

The following tables show the character positions for each graphic character in the 4704 character set. For each character, you can determine the applicable output position code from the matrix.

High-Order Output Position Code (columns); **Low-Order Output Position Code** (rows)

	C	D	E	F	C	D	E	F	C	D	E	F	C	D	E	F
0	à	À	û	Û	ア	チ	ム	「	א	א	נ	נ	α	σ	Ψ	P
1	è	È	á	Á	イ	ツ	メ	」	ב	ב	ס	ס	β	τ	A	Q
2	ì	ð	é	É	ウ	テ	モ	、	ג	ג	ע	ע	γ	υ	B	R
3	ò	Nul	í	Í	エ	ト	ヤ	・	ד	ד	ף	ף	δ	φ	C	S
4	ù	Ù	ó	Ó	オ	ナ	ユ	ヲ	ה	ה	פ	פ	ε	χ	D	T
5	ã	Ã	ú	Ú	カ	ニ	ヨ	ァ	ו	ו	ץ	ץ	ζ	ψ	E	U
6	õ	Õ	ñ	Ñ	キ	ヌ	ラ	ィ	ז	ז	צ	צ	η	ω	F	V
7	ä	Ä	å	Å	ク	ネ	リ	ゥ	ח	ח	ק	ק	θ	ς	G	W
8	ë	Ë	ç	Ç	ケ	ノ	ル	ェ	ט	ט	ר	ר	ι	Γ	H	X
9	ï	Ï	ø	Ø	コ	ハ	レ	ォ	י	י	ש	ש	κ	Δ	I	Y
A	ö	Ö	ý	Ý	サ	ヒ	ロ	ャ	ך	ך	ת	ת	λ	Θ	J	Z
B	ü	Ü	æ	Æ	シ	フ	ワ	ュ	כ	כ	ć	Ć	μ	Λ	K	⌒
C	â	Â	þ	¤	ス	ヘ	ン	ョ	ל	ל	č	Č	ν	Ξ	L	Nul
D	ê	Ê	Þ	ß	セ	ホ	゛	ッ	ם	ם	š	Š	ξ	Π	M	Nul
E	î	Î	ƅ	¿	ソ	マ	°	ー	מ	מ	ž	Ž	π	Σ	N	Nul
F	ô	Ô	µ	Res	タ	ミ	°	Res	ן	ן	đ	Res	ρ	Φ	O	Res
	EB1				EB2				EB3				EB4			

EBCDIC Values for 4704 Multilingual Graphics (CHARSET=EB)

High-Order Hexadecimal

Low Order Hexadecimal

Column	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0					SP	&	-	ø	Ø	°	µ	¢	{	}	\	0
1						é	/	É	a	j	~	£	A	J		1
2					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
3					ä	ë	Ä	Ë	c	l	t	Pts	C	L	T	3
4					à	è	À	È	d	m	u		D	M	U	4
5					á	í	Á	Í	e	n	v	§	E	N	V	5
6					ã	î	Ã	Î	f	o	w		F	O	W	6
7					å	ï	Å	Ï	g	p	x		G	P	X	7
8					ç	ì	Ç		h	q	y	½	H	Q	Y	8
9					ñ	ß	Ñ	`	i	r	z		I	R	Z	9
A					[	]	¦	:				¬				
B					.	$	,	#			¿	|	ô	û	Ô	Û
C					<	*	%	@	ð	æ	Đ	-	ö	ü	Ö	Ü
D					(	)	_	'	ý	¸	Ý	¨	ò	ù		Ù
E					+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
F					!	^	?	"		¤			õ		Õ	

Modifying Translation Tables

The following tables show the character positions for each graphic character in the 4704 character set. For each character you can determine the applicable output position code from the matrix. Use this position code with the hexadecimal character input value to modify a translation table.

High Order (Left-most) Digit of Output Position

Low Order (Right-most) digit of Output Position

	0	1	2	3	4	5	6	7
0	Nul	|	Sp	0	@	P	‾ *	p
1	Sp	¬	!	1	A	Q	a	q
2	Nul	Ǧ	"	2	B	R	b	r
3	Nul	'	#	3	C	S	c	s
4	Nul	'	$	4	D	T	d	t
5	Nul	Ĩ	%	5	E	U	e	u
6	Nop	ĝ	&	6	F	V	f	v
7	≠	§	'	7	G	W	g	w
8	¢	Đ	(	8	H	X	h	x
9	[illegible]	0	)	9	I	Y	i	y
A	[illegible]	`	*	:	J	Z	j	z
B	Pt	°	+	;	K	[	k	{
C	[illegible]	~	,	<	L	\	l	;̄
D	[illegible]	£	-	=	M	]	m	}
E	½	¦	.	>	N	^	n	!!
F	¥	¨	/	?	O	_	o	‾

To use this value in the configuration process, convert the two-digit hexadecimal position code (X'54', the upper case T, for example) to decimal (84) for use in the OUTRTBL macro. This decimal value is used in the POS operand. Then, code the hexadecimal equivalent of the graphic character (X'E3' for the character T) in X'n' operand.

The graphics for position codes X'C0' through X'FF' depend on the character set selected, and are shown in the next matrix. Codes X'80' through X'AF' are treated as no-operation codes; X'B0' through X'BF' are for display orders.

High Order Digit of Output Position

Low Order Digit of Output Position

CHARSET = US

	C	D	E	F
0	à	À	û	Û
1	è	È	á	Á
2	ì	ð	é	É
3	ò	Nul	í	Í
4	ù	Ù	ó	Ó
5	ã	Ã	ú	Ú
6	õ	Õ	ñ	Ñ
7	ä	Ä	å	Å
8	ë	Ë	ç	Ç
9	ï	Ï	ø	Ø
A	ö	Ö	ý	Ý
B	ü	Ü	æ	Æ
C	â	Â	þ	¤
D	ê	Ê	Þ	β
E	î	Î	›	¿
F	ô	Ô	µ	Res

CHARSET = KATAKANA

	C	D	E	F
0	ア	チ	ム	「
1	イ	ツ	メ	」
2	ウ	テ	モ	、
3	エ	ト	ヤ	・
4	オ	ナ	ユ	ヲ
5	カ	ニ	ヨ	ァ
6	キ	ヌ	ラ	ィ
7	ク	ネ	リ	ゥ
8	ケ	ノ	ル	ェ
9	コ	ハ	レ	ォ
A	サ	ヒ	ロ	ャ
B	シ	フ	ワ	ュ
C	ス	ヘ	ン	ョ
D	セ	ホ	゛	ッ
E	ソ	マ	°	ー
F	タ	ミ	•	Res

Defining the 4704-1 for Configuration

Controller configuration macro instructions are used to specify the physical and logical configuration of each 4701 and its associated terminals. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. This section contains specific notes for configuring the 4704. Refer to Volume 6 for a description and information on coding the DEV4704 configuration macro.

There is no MSLITE operand on DEV4704. The 4704 magnetic device indicator lights are controlled by the terminal rather than by the program.

If you are converting from a DEV3604 macro, be sure to change the TYPA and SHFT operands of the TRTBHDR macro. The 3604 scan codes are not the same.

Chapter 7. IBM 4704 Display, Models 2 and 3

These IBM displays (Figure 7-2) attach to the Device Cluster Adapter (DCA) and provide a choice of two display screens and either one or two of four keyboards. These 4704-2/3 displays can also include a personal identification number (PIN) keypad and a magnetic stripe device.

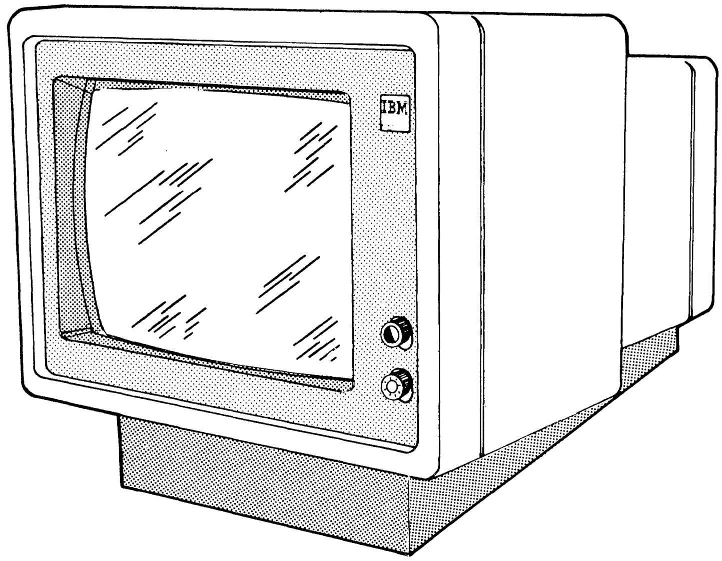


Figure 7-1. 4704-2/3 Display

Models and Features

The following figure shows the basic display and keyboard choices for the three models of the 4704:

4704 Model:	Display Size (Diagonal):	Format (rows) x (columns):	Available Keyboards:
1 (Loop)	140 mm (5.5 in)	12 x 40 standard	50-key function, 62- or 77-key alphameric, 50-/62-key combined, or administrative keyboards*
	230 mm (9.0 in)	24 x 80 standard or 12 x 40 large	
2 (DCA)	230 mm (9.0 in)	24 x 80 standard or 12 x 40 large	
3 (DCA)	300 mm (12.0 in)	24 x 80 standard or 12 x 40 large	

*Katakana keyboards (63- and 78-key) each have an extra key.

Figure 7-2. IBM 4704-2/3 Options and Features

In addition to the choices in Figure 7-2, the 4704-2/3 display can also have to following:

- Uppercase and lowercase characters
- Normal and intensified display characters
- Optional antiglare filters and support cradles

Keyboard Component

The 50-key keyboard is a numeric/function keyboard. The 62-key keyboard (63 for Katakana) is a complete alphameric keyboard. The 77-key keyboard (78 for Katakana) contains the entire 62-key keyboard along with 15 additional keys. The 107-key administrative keyboard contains the 62-key keyboard with 15 control keys and 30 programmable keys. You can also attach the 62-key keyboard to the primary port and the 50-key keyboard to the secondary port to have a 112-key keyboard.

Display Monitor Component

The 4704-2 and 4704-3 display monitors provide a choice suitable for either work station (transaction) or administrative (large volume) data operations. The 4704-2 and 4704-3 attached to the device cluster adaptor (DCA) to permit fast access to the controller. Their power/logic functions are contained within the display monitor.

Optional Display Devices

Each 4704-2/3 display station can have a PIN keypad for users to enter their personal identification numbers. Each 4704-2/3 display can also have a magnetic stripe reader or magnetic stripe reader/encoder.

Dual Intensity

The 4704-2/3 displays permit you to brighten select parts of the display using the set attribute control character (X'28C1*aa*', where *aa* is the desired intensity attribute). An attribute of X'40' or X'C4' indicates normal intensity; an attribute of X'C8' indicates high intensity. When written to the display, the set attribute control character:

- Occupies a display position
- Displays as a blank (space)
- Causes characters in display positions following the attribute's display position to be displayed with the specified intensity. Attribute control wrap occurs from the lower right-hand corner to the upper left-hand corner of the display.

For example, if the following data stream were written to the display:

```
X'0CF128C1C8F2F328C140F4'
  1 2   3    4    5   6
```

the following would be displayed at the upper left-hand corner of the display screen:

```
      Col
Row   123456
  1 | 1 23 4
  2 |
```

Notes:

1. Clears the screen (ensuring that attributes are not inadvertently left on the display).
2. Displays the character '1' in column 1 of row 1 with normal intensity.
3. Sends the high-intensity attribute byte to display column 2, row 1. Note that the attribute is displayed as a blank.
4. Displays the characters '2' and '3' in high intensity.
5. Sends the normal intensity attribute byte to display column 5, row 1. Note that the attribute is displayed as a blank. All display characters following this attribute are at normal intensity; this includes the character '1' in column 1 of row 1 (because of the attribute control wrapping).
6. Displays the character '4' in column 6 of row 1 in normal intensity.

Controls and Indicators

Power and Operating Controls and Indicators

These controls and indicators are on the display module.

Power On/Off Switch

This is a "push-pull" switch that turns the display on or off. Switching power ON applies line power to both the keyboard and display, and starts the built-in test sequence.

Test Indicator

This is a symbolic indicator on the 4704-2 and -3, and is on during the basic tests that are performed automatically when the terminal is switched on. When the tests are completed, the indicator remains on only if an error occurred.

Keyboard Alarm Control and PIN Keypad Light

Audible Alarm

This control adjusts the volume of the audible alarm. It is on the underside of the keyboard.

PIN Keypad Light

This red lamp indicates that keypad data can be entered; all other keyboard input except the Reset key is disabled.

Display Monitor Controls

Contrast Control

This control, on the display monitor, adjusts the intensity difference between the high- and the normal-intensity characters.

Brightness Control

This control on the display monitor adjusts the brightness of all displayed characters.

Operation Indicators on The Display

The indicators described in this section are symbols displayed at the bottom of the display monitor screen.

Not Operational

This indicator is displayed when one of the power-on tests fails. A two-digit error code is displayed next to this indicator.

MSR Ready to Read

A valid magnetic stripe can now be passed through the reader.

MSE Ready to Encode

Encoding data has been sent to this terminal, and the magnetic stripe encoder is ready to encode data. This indicator is removed when a magnetic-striped card is passed through the encoder.

MSR/E Check

This indicator is displayed when an error occurs at the magnetic stripe reader/encoder.

Too Much Data

The operator entered more data than the current field can hold. If this indicator is displayed, no further keys are read until the operator presses the Reset key.

PIN Keypad Enabled

This symbol indicates that the PIN keypad is enabled, and all other terminal input (except the Reset key) is disabled. Refer to "Personal Identification Number (PIN) Keypad" on page 7-9 for detailed information on how this indicator operates.

User-Programmable Indicators

Three user programmable indicators and a System Check indicator are available. These indicators are controlled by the SIGNAL instruction in the application program. If a PIN keypad is attached, indicator 3 selects the PIN keypad and lights the PIN Keypad Enabled indicator.

Shift Indicators

These indicators show which keyboard shift case is currently active.

Programming Considerations

The 4704-2/3 displays operate from either directly from a 4700 application program or through the data stream mapping (DATSM) facility. When programmed through DATSM, the displays operate in one of two modes:

- *Controller tracking mode*—This normal mode is the same as that already used by other DCA-attached display terminals on the 4700 system. The program writes data to the 4704-2/3 display with the LWRITE instruction, and reads keyboard data using LREAD. The LREAD can either reissue the keyboard data directly to the display (TRACK), or read the data without redisplaying it. Your program can process the data as an SNA character string (SCS), or through the DATSM facility.

- *Local tracking mode*—In this DATSM support mode, the program issues either the PUTFLD instruction only or a PUTFLD/GETFLD instruction sequence to transfer data through DATSM to the display. The application program issues LREAD only to determine the current display status, and issues LWRITE only to write data to a magnetic stripe encoder attached to the 4704-2/3. In local tracking mode, the 4704-2/3 rewrites, or "tracks", keyboard data directly to the display from the keyboard until the operator presses an Attention key, signalling the program to read the cursor position.

The program must issue PUTDMS to place the 4704-2/3 in local tracking mode by setting the DMSBFM local tracking flag in DMSMOD. The default for the DMSBFM flag is zero (controller tracking mode). Any attempt to set the DMSBFM flag for a terminal other than a 4704-2/3 is ignored.

PUTDMS must also place the logical device address (LDA) of the display in the DATSM machine segment field, DMSLDA. This is the LDA to which data is being sent. The default value for the LDA is '1'.

Controller Tracking Mode

The 4704-2/3 display running in controller tracking mode can be programmed identically to a 3278-2. The controller translates the display data stream containing EBCDIC data into SCS data according to the CPGEN-supplied translate tables, and then writes the data to the display. Data is read from the 4704-2/3 keyboard as scan codes, which are translated into EBCDIC and written into user storage.

Any existing application program that uses DCA-attached displays or communicates using SCS data streams and LWRITE, WRTI, LREAD, LCHECK, SIGNAL, and DEVPARM can operate on the 4704-2/3 displays in controller tracking mode without change.

Local Tracking Mode

This mode supports DATSM programs that are written to take advantage of the independent tracking operation that the 4704-2/3 is capable of performing. This reduces the I/O operations the program must perform, but means that existing programs that issue LREAD or LWRITE instructions to the DATSM display must be changed. This section discusses local tracking program operation and restrictions.

Program Operation with Local Tracking

The DATSM application program uses the PUTFLD and GETFLD instructions to perform I/O to the display. However, keyboard– to–display tracking occurs within the 4704-2/3; no keyboard data returns to the program until the operator enters end–of–message (EOM), which locks the keyboard. The program should then issue LREAD to read the AID byte and cursor address, which a subsequent GETFLD then uses to read the message data.

In local tracking mode, the DATSM instructions start I/O operations. This means that programs written for local tracking mode must manage the 4704-2/3 differently from other DATSM–driven terminals.

Each work station using DATSM can have its own DATSM image buffer, or all stations can share a system image buffer. When a work station has its own image buffer and operates in local tracking mode, the controller reads the buffer each time an AID byte requires a long read. The contents of a system image buffer, however, are transferred by each PUTFLD or GETFLD.

Local Tracking Restrictions

The following functions, supported in normal (controller tracking) DATSM operation, are not possible in local tracking mode:

1. *Character string entry*: Entering a multi-character string from a single key scan code is not supported. Each keystroke can represent only one character graphic. Accented characters, however, are supported.

2. *Different image buffers*: An image buffer created in one mode cannot be used in another mode, due to data incompatability. If the tracking mode flag is changed between the time of the first PUTFLD and a subsequent PUTFLD or GETFLD, the latter instruction fails and the controller posts status.

Controller Configuration

Controller configuration macro instructions are used to specify the physical and logical configuration of each 4701 and its associated terminals. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. This section contains specific notes for configuring the 4704-2 and 4704-3. Refer to Volume 6 for a description and information on coding the DCA4704-2/3 configuration macro.

There is no MSLITE operand on DCA4704. The 4704-2/3 magnetic device indicator lights are controlled by the terminal rather than by the program.

The DEFSBF system buffer definition macro and the NBF operand of the STATION macro define the system image buffer or buffers used for DATSM operation. Specifying NBF for the STATION macro ensures that no work station image buffer is allocated.

You cannot define transient shift, downshift, or repeat–action keys for a keyboard attached to the secondary keyboard port.

Note: 4704-2/3 display data is restricted to 0–9, A–Z, and a–z when the terminal is attached to a DCA port defined for a 3278/3279.

Magnetic Stripe Reader and Reader/Encoder

The International Standard ISO 3554 specifies the physical and magnetic characteristics for a magnetic stripe, which can have three tracks. The magnetic stripe reader and reader/encoder can read the ABA 75-bpi track 2 data written on credit or ID cards. The magnetic stripe reader/encoder or the 75/210-bpi reader reads either the 75-bpi track 2 data or 210-bpi data in accordance with the 4-bit IBM Specifications code. The magnetic stripe reader/encoder can encode the IBM 210-bpi data on track 2 of a magnetic stripe. It does not, however, encode 75-bpi ABA data.

The units have three lights to inform the operator of various conditions. Two indicators on the display screen are associated with operation of the magnetic devices. These are MSE READY to ENCODE, MSR READY, and MSR/E CHECK. One of these indicators causes an audible tone when they are set on; their operation is described below.

Read Operation

During a read operation, the indicators are:

GREEN
: Ready to read. The indicator is on whenever the magnetic device is ready to read. The indicator turns off after the magnetic medium is moved through the reader, at which time the yellow indicator comes on.

YELLOW

In process, wait. This indicator comes on after the medium is moved through the reader; magnetic data is pending transmission to the controller. After the data is transmitted, this indicator goes off and the green indicator returns. However, if an error is detected, the red indicator comes on with the green indicator.

RED

Error detected, retry. If a read error occurs (no data detected, parity or longitudinal redundancy check error), the red indicator comes on (along with the green). This indicates an error has occurred. The red indicator remains on until either a good read occurs, or the operator presses the Reset key while the Alternate key is held down.

For the data to be transmitted to the controller, the application program must issue an LREAD. After the controller has selected the device for the read, input can occur from the keyboard, PIN keypad, and magnetic device in any sequence. However, only one device can operate at a time.

Encode Operation

During encoding, the indicators are:

GREEN

Flashing -- encoder enabled; ready to encode. The indicator is normally on steadily to indicate that a read can occur. When the controller enables the encoder to encode a document, the indicator starts to flash 2 or 3 times per second. The magnetic medium may now be passed through the slot to encode. When this operation is started, the green indicator is turned off, and the yellow indicator is turned on.

YELLOW

In process; wait. This indicator is turned on immediately after a medium is passed through the encoding slot. It stays on until after the read-back check is complete, or for 0.5 second, whichever is longer. If the read-back check is good, the yellow indicator is turned off, and the steady green indicator returns. If the read-back check fails, the yellow indicator is turned off, and the red indicator is turned on and the green indicator resumes flashing.

RED

Error detected; retry. The red indicator signifies an error during the encoding operation. It stays on until one of the following events occur:

- A subsequent good encode occurs.
- Local Reset occurs (press Reset while holding Alt).
- Programmed disable (described below) is issued.

After any of these conditions occur, the green indicator returns.

Programmed Disable

The encoder can be disabled by an LWRITE of a command reset before the first encode is completed. This action disables the encoder and resets the reader.

Personal Identification Number (PIN) Keypad

The PIN keypad consists of ten data keys, two function keys, and a red "enter PIN" indicator. The keypad attaches by cable to the 4704-2 or 4704-3 display.

The PIN keypad port accommodates two types of keypads: a clear and an encrypted pad. An encrypting keypad has one additional switch to enable an authorized operator to enter an encryption key value.

The PIN keypad appears to the program as an extension of the keyboard. The program enables the PIN keypad by setting on programmable indicator 3 using the SIGNAL instruction. Enabling the PIN keypad also turns on the PIN keypad enabled indicator on the display screen and the red lamp on the keypad.

Data is not transmitted from the keypad to the controller until the operator presses the End key. This causes the encrypting keypad to send X'7E', followed by 24 characters plus the X'7F' ending character to the controller. The clear keypad sends X'7F' followed by the keyed data and the X'7F' ending character. The PIN indicator and programmable indicator 3 on the display go off, and the keypad lamp also goes off; the PIN keypad is deselected.

If the Reset key on the keyboard is pressed in lieu of the End key, no data is transmitted from the pad, and the keypad is deselected. An encrypting keypad sends X'7E7F', and a clear keypad sends X'7F7F'; either of which ends PIN pad input to the controller.

After an input error, the customer can press the Erase key on the pad, and reenter the PIN. The data can then be reentered.

The PIN Keypad can be disabled by:

1. pressing the End key on the PIN pad.
2. pressing the Reset key on the keyboard.
3. the application program turning off programmable indicator 3 before the End or Reset key is pressed.

When the PIN Pad is disabled by the End or Reset key, the keypad lamp, programmable indicator 3, and the PIN keypad enabled indicator are turned off immediately. The End or Reset key ending code is placed in the data stream to indicate the end of PIN data. The controller recognizes the code and turns programmable indicator 3 off. The application program can end the PIN operation before the operator by turning off programmable indicator 3. This deselects the PIN pad, and all indicators go off.

Terminal Control Characters

Your program must embed control characters in the SCS data stream to control data-dependent terminal operation. The following are the valid control characters for the 4704:

Function:	Control Character:	Explanation:
Line Feed	X'25'	The cursor is moved down one line; the horizontal position is unchanged.
Return	X'0D'	The cursor moves to position 1 on the current display line.
New Line	X'15'	The current line is erased from the cursor to the end of the line. The cursor moves to position 1 of the next lower line.
New Page	X'0C'	The screen is completely erased; the cursor is moved to position 1 of line 1. If there is a magnetic stripe encoder (MSE) on the 4704-2/3 that is ready to use, a write to the encoder with a first byte of X'0C' in the message causes the encoder to become not ready. Reset should not be issued for a MSE that is not ready to use (not in the encode state).
Set Attribute	X'28'	This 3-byte control string defines the following field characteristics and sends one byte to the display. The second byte should be X'C1'. The third byte may be one of these values: X'40' Normal intensity X'C4' Normal intensity X'C8' Intensified X'4C' Nondisplay

Function:	Control Character:	Explanation:
Position	X'34'	This control character is followed by a flag byte and a 1-byte positioning value. The second byte is the flag byte: Bits 0-3 are ignored. Bit 4: 0 = Position is an absolute value. 1 = Position is relative to the present position. Bit 5: 0 = Position change is horizontal. 1 = Position change is vertical. Bits 6-7: 00 = Move cursor to new position without erasing. 01 = Leave cursor at the current position, but erase from the current position the number of bytes specified by the position. 10 = Move cursor to new position and erase from the old position up to, but not including, the new position. 11 = Reserved. The third byte controls position. The positioning byte indicates a line or column number, which is an unsigned binary value ranging from 0 to 255. This value is used under control of the flag byte; a 0 results in no operation. The specified value is divided by a modulus number to determine the resulting position. The modulus division produces a change in vertical positioning of 1 to *x* lines, and a change in horizontal positioning of 1 to *y* columns where 'x' is the maximum number of lines and 'y' is the maximum number of columns.
Select	X'04'	The byte following this control character is skipped, and the automatic new-line action is suppressed.

Function:	Control Character:	Explanation:
Transparent Write	X'35'	A transparent write (a write without translation) begins. The format of this control sequence is: Byte 0: The control character, X'35'. Byte 1: The number of data bytes to be displayed. Byte 2-Byte n: The transparent data to be displayed. The valid positions are X'00' - X'FF'.

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or they may be restarted, or "wrapped back," to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 4704:

Keyboard:

Counter:	Explanation:
1	DCA adapter errors
2	Port/component/device errors
3	(reserved)
4	Translation checks
5	Segment overruns
6–15	(reserved)

Display:

Counter:	Explanation:
1	DCA adapter errors
2	Port/component/device errors
3	Device error (device did not respond in allotted time.
4	Controller error (device rejected the request as unsupported.
5	Keyboard power-on test failure
6	Magnetic device power-on test failure
7–12	(reserved)

Magnetic Stripe Reader/Encoder:

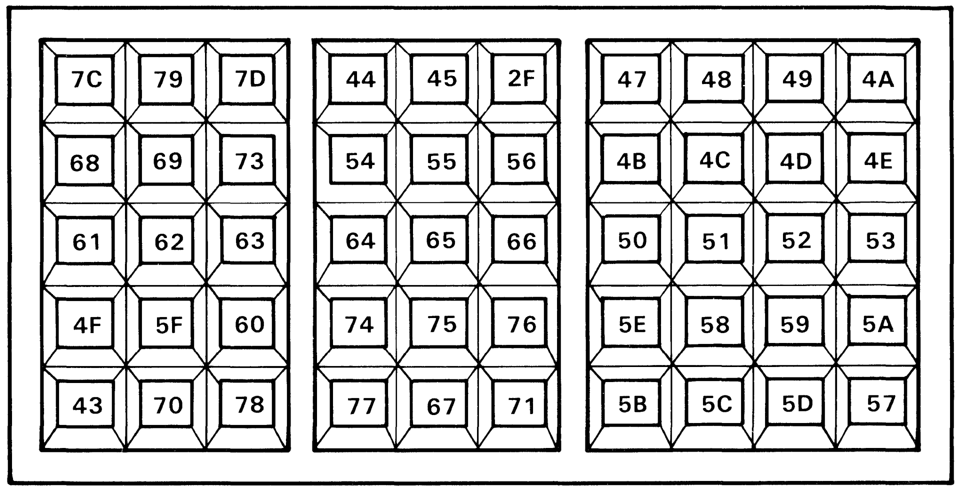
Counter:	Explanation:
1	DCA adapter errors
2	Port/component/device errors
3	Device error (format error or encoding failure)

Keyboards and Translation Tables

The keyboards for the 4704-2/3 are the same as for the 4704-1. When operating in controller tracking mode, the keyboard scan codes for the 4704-2/3 are the same as the 4704-1 except when attaching the 50-key (Model 100) keyboard to the secondary port. This is done when the 50-key keyboard is an extension to the 62-key keyboard attached to the primary keyboard port.

Alternate Scan Codes for the 50-Key Keyboard

The following is the scan code table for the 50-key keyboard when it is attached to the secondary keyboard port:



Note: You cannot define transient shift, downshift, or repeat action keys for keyboards attached to the secondary port.

Refer to Chapter 6, "IBM 4704 Display Station Model 1" for the 50-key keyboard/primary attachment description as well as the controller tracking mode descriptions of the other keyboards that attach to the 4704 Models 2 and 3. The rest of this section describes local tracking mode tables and character sets.

High Order Digit
of
Output Position

Low Order Digit of Output Position

CHARSET = US

	8	9	A	B
0	à	À	û	Û
1	è	È	á	Á
2	ì	ð	é	É
3	ò	Nul	í	Í
4	ù	Ù	ó	Ó
5	ã	Ã	ú	Ú
6	õ	Õ	ñ	Ñ
7	ä	Ä	å	Å
8	ë	Ë	ç	Ç
9	ï	Ï	ø	Ø
A	ö	Ö	ý	Ý
B	ü	Ü	æ	Æ
C	â	Â	þ	¤
D	ê	Ê	Þ	β
E	î	Î	[illegible]	¿
F	ô	Ô	µ	Res

CHARSET = KATAKANA

	8	9	A	B
0	ｱ	ﾁ	ﾑ	｢
1	ｲ	ﾂ	ﾒ	｣
2	ｳ	ﾃ	ﾓ	､
3	ｴ	ﾄ	ﾔ	･
4	ｵ	ﾅ	ﾕ	ｦ
5	ｶ	ﾆ	ﾖ	ｧ
6	ｷ	ﾇ	ﾗ	ｨ
7	ｸ	ﾈ	ﾘ	ｩ
8	ｹ	ﾉ	ﾙ	ｪ
9	ｺ	ﾊ	ﾚ	ｫ
A	ｻ	ﾋ	ﾛ	ｬ
B	ｼ	ﾌ	ﾜ	ｭ
C	ｽ	ﾍ	ﾝ	ｮ
D	ｾ	ﾎ	ﾞ	ｯ
E	ｿ	ﾏ	ﾟ	ｰ
F	ﾀ	ﾐ	•	Res

The following table lists the keyboard function scan codes when the 4704-2/3 operates in local tracking mode:

Without ALT Key		With ALT Key	
x'C0' Backspace	x'D0' Csr Up	x'C0'	x'D0' Alt Csr
x'C1' Tab Left	x'D1' Csr Dn	x'C1'	x'D1' Csr Blink
x'C2' Tab Right	x'D2' Csr L	x'C2'	x'D2' Csr Fast L
x'C3' Home	x'D3' Csr R	x'C3'	x'D3' Csr Fast R
x'C4'	x'D4'	x'C4'	x'D4'
x'C5'	x'D5'	x'C5'	x'D5'
x'C6' Dup	x'D6'	x'C6'	x'D6'
x'C7' Field Mark	x'D7'	x'C7'	x'D7'
x'C8' Insert	x'D8' Kanna Sh	x'C8'	x'D8'
x'C9' Delete	x'D9' Kanna Sh Lk	x'C9'	x'D9'
x'CA' Erase Input	x'DA' Grave DK	x'CA'	x'DA' Grave DK
x'CB' Erase EOF	x'DB' Acute DK	x'CB'	x'DB' Acute DK
x'CC' New Line	x'DC' Diaeresis DK	x'CC'	x'DC' Diaeresis DK
x'CD' Shift	x'DD' Cir'flex DK	x'CD'	x'DD' Cir'flex DK
x'CE' Shift Lock	x'DE' Tilde DK	x'CE'	x'DE' Tilde DK
x'CF'	x'DF' Cedila DK	x'CF' Test Mode	x'DF' Cedila DK

The following table shows the scan code to display character default translation for local tracking mode. You can specify other values using the DCATRTBL TYPE= configuration macro and operands.

Scan Code	Lower Case Element	Lower Case Function	Upper Case Element	Upper Case Function	Alternate Element	Alternate Function
00	1A	Grave	1C	Tilde	FF	AID
03	70	p	50	P	BF	No-op
04						
05	71	q	51	Q	DA	Grave
06	77	w	57	W	DB	Acute
08	37	7	26	&	FF	AID
09	38	8	2A	*	FF	AID
0A	39	9	28	(	FF	AID
0B	BF	No-op	BF	No-op	BF	No-op
0C	BF	No-op	BF	No-op	BF	No-op
0D	3D	=	2B	+	FF	AID
0E	C0	Bksp	C0	Bksp	BF	No-op
0F	30	0	29	)	FF	AID
10	34	4	24	$	FF	AID
11	35	5	25	%	FF	AID
12	36	6	11	–	FF	AID
13	65	e	45	E	DC	Diaeresis
14	72	r	52	R	DD	Circumflex
15	74	t	54	T	DE	Tilde
16	79	y	59	Y	DF	Cedila
17	75	u	55	U	BF	No-op
18	31	1	10	|	FF	AID
19	32	2	40	@	FF	AID
1A	33	3	23	#	FF	AID
1B	08	¢	21	!	BF	No-op
1C	5C	Rev Slash	1E	Vert Line	BF	No-op
1D	C1	Tab L	C1	Tab L	C1	No-op
1E	BF		BF		BF	No-op
1F	2D	—	5F	—	FF	AID
20	CE	Sh Lock	CE	Sh Lock	BF	No-op
21	61	a	41	A	BF	No-op
22	73	s	53	S	BF	No-op
23	64	d	44	D	BF	No-op
24	66	f	46	F	BF	No-op
25	67	g	47	G	BF	No-op
26	68	h	48	H	BF	No-op
27	6A	j	4A	J	BF	No-op
28	6B	k	4B	K	BF	No-op
29	6C	l	4C	L	BF	No-op
2A	3B	;	3A	:	BF	No-op
2B	27	'	22	''	BF	No-op
2C	7B	Opn Brace	7D	Clo Brace	BF	No-op
2D	CC	New Ln	CC	New Ln	BF	No-op
2E	69	i	49	I	BF	No-op
2F	FF	AID	FF	AID	BF	No-op
30	CD	Shift	CD	Shift	BF	No-op
31	BF	Special	BF	Special	BF	Special
32	7A	z	5A	Z	BF	No-op
33	78	x	58	X	BF	No-op

Scan Code	Lower Case Element	Lower Case Function	Upper Case Element	Upper Case Function	Alternate Element	Alternate Function
34	63	c	43	C	BF	No-op
35	76	v	56	V	CF	Test
36	62	b	42	B	D0	ALT Csr
37	6E	n	4E	N	D1	Csr Blnk
38	6D	m	4D	M	BF	No-op
39	2C	,	2C	,	BF	No-op
3A	2E	.	2E	.	BF	No-op
3B	2F	/	3F	?	BF	No-op
3C	BF	No-op	BF	No-op	BF	No-op
3D	CD	Shift	CD	Shift	BF	No-op
3E	3C	<	3E	>	BF	No-op
3F	BF	No-op	BF	No-op	BF	No-op
40	20	Space	20	Space	BF	No-op
41	BF	No-op	BF	No-op	BF	No-op
42	BF	No-op	BF	No-op	BF	No-op
43	30	0	29	)	BF	No-op
44	C6	Dup	C6	Dup	BF	No-op
45	C7	Fld Mrk	C7	Fld Mrk	BF	No-op
46	FF	AID	FF	AID	FF	AID
47	61	a	41	A	BF	No-op
48	62	b	42	B	BF	No-op
49	63	c	43	C	BF	No-op
4A	64	d	44	D	BF	No-op
4B	65	e	45	E	BF	No-op
4C	66	f	46	F	BF	No-op
4D	67	g	47	G	BF	No-op
4E	68	h	48	H	BF	No-op
4F	31	1	10	|	BF	No-op
50	69	i	49	I	BF	No-op
51	6A	j	4A	J	BF	No-op
52	6B	k	4B	K	BF	No-op
53	6C	l	4C	L	BF	No-op
54	C8	Insert	C8	Insert	FF	PA1
55	C9	Delete	C9	Delete	FF	PA2
56	FF	AID	FF	AID	BF	No-op
57	74	t	54	T	BF	No-op
58	6E	n	4E	N	BF	No-op
59	6F	o	4F	O	BF	No-op
5A	70	p	50	P	BF	No-op
5B	71	q	51	Q	BF	No-op
5C	72	r	52	R	BF	No-op
5D	73	s	53	S	BF	No-op
5E	6D	m	4D	M	BF	No-op
5F	32	2	40	@	BF	No-op
60	33	3	23	#	BF	No-op
61	34	4	24	$	BF	No-op
62	35	5	25	%	BF	No-op
63	36	6	11	¬	BF	No-op
64	BF	Special	BF	Special	BF	Special

Scan Code	Lower Case Element	Lower Case Function	Upper Case Element	Upper Case Function	Alternate Element	Alternate Function
65	D0	Csr UP	D0	Csr UP	BF	No-op
66	CA	Er Input	CA	Er Input	BF	No-op
67	D1	Csr DN	D1	Csr DN	BF	No-op
68	37	7	26	&	BF	No-op
69	38	8	2A	*	BF	No-op
6A	75	u	55	U	BF	No-op
6B	76	v	56	V	BF	No-op
6C	77	w	57	W	BF	No-op
6D	78	x	58	X	BF	No-op
6E	79	y	59	Y	BF	No-op
6F	7A	z	5A	Z	BF	No-op
70	2E	.	2E	.	BF	No-op
71	E6	Print	E6	Print	BF	No-op
72	2C	,	2C	,	BF	No-op
73	39	9	28	(	BF	No-op
74	D2	Csr L	D2	Csr L	D2	Csr Fst L
75	C3	Home	C3	Home	BF	No-op
76	D3	Csr R	D3	Csr R	D3	Csr Fst R
77	CB	Er EOF	CB	Er EOF	BF	No-op
78	3D	=	2B	+	BF	No-op
79	3C	<	3E	>	BF	No-op
7A	6F	o	4F	O	BF	No-op
7B	C2	TAB R	C2	TAB R	BF	No-op
7C	CD	SHIFT	CD	SHIFT	BF	No-op
7D	CE	SH LOCK	CE	SH LOCK	BF	No-op

Notes:

1. The keyboards do not create scan codes X'07', X'2E', X'7A', or X'7B'.
2. All keys with scan codes of X'FF' issue an EID defined for that key by the INTRTBL configuration macro.
3. Scan code X'46' is reserved for system monitor logon.

Modifying Translation Tables

The following tables show the character positions for each graphic character in the 4704-2/3 character set. For each character you can determine the applicable output position code from the matrix. Use this position code with the hexadecimal character input value to modify a translation table.

High Order (Left-most) Digit of Output Position

Low Order (Right-most) digit of Output Position

	0	1	2	3	4	5	6	7
0	Nul	|	Sp	0	@	P	‾*	p
1	Sp	¬	!	1	A	Q	a	q
2	Nul	Ǧ	"	2	B	R	b	r
3	Nul	'	#	3	C	S	c	s
4	Nul	ˈ	$	4	D	T	d	t
5	Nul	Ĩ	%	5	E	U	e	u
6	Nop	ĝ	&	6	F	V	f	v
7	≠	§	'	7	G	W	g	w
8	¢	Ð	(	8	H	X	h	x
9	ȿ	0	)	9	I	Y	i	y
A	ȿ	`	*	:	J	Z	j	z
B	₧	°	+	;	K	[	k	{
C	[illegible]	~	,	<	L	\	l	‾;
D	[illegible]	£	-	=	M	]	m	}
E	½	¦	.	>	N	^	n	"
F	¥	¨	/	?	O	_	o	‾

To use this value in the configuration process, convert the two-digit hexadecimal position code (X'54', the upper case T, for example) to decimal (84) for use in the OUTRTBL macro. This decimal value is used in the POS operand. Then, code the hexadecimal equivalent of the graphic character (X'E3' for the character T) in X'n' operand.

The graphics for position codes X'80' through X'BF' depend on the character set selected, and are shown in the next matrix. Codes X'C0' through X'FF' are processes as attributes.

Chapter 8. IBM 3604 Keyboard Display

The IBM 3604 Keyboard Display is made up of a keyboard component that can have numeric or alphameric keys and function keys, a display component that can display 120, 240, 480, 1024, or 1920 characters, depending on model, and, optionally, a magnetic stripe reader or encoder/reader. The controller uses the keyboard and display as separate components; the controller treats the magnetic stripe reader as one part of the keyboard component, but handles the magnetic stripe encoder separately.

Typical applications for the 3604 Keyboard Display are entering deposit and withdrawal transactions, making account inquiries, verifying calculations, and providing guidance to the operator.

All 3604 keyboards and translation tables are shown in this chapter under the heading "Keyboards and Translation Tables".

Models and Features

Keyboard Component

The IBM 3604 Keyboard Display Model 1 (Figure 8-1) can have a numeric keyboard or an expanded numeric keyboard. The numeric keyboard has 30 keys, made up of two sets of 15 keys. One set has 12 keys engraved with numerics and three nonengraved keys; the other set has 15 nonengraved keys. The expanded numeric keyboard has 45 keys, made up of three sets of 15 keys. One set contains 12 keys engraved with numerics and three nonengraved keys; the other two sets contain 15 nonengraved keys each.

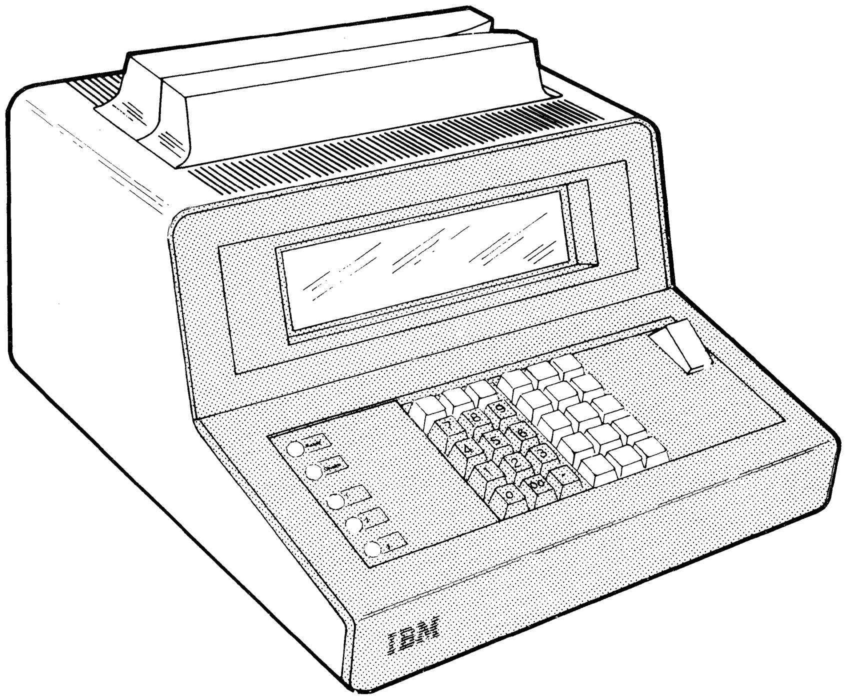


Figure 8-1. IBM 3604 Keyboard Display Model 1

The 3604 Keyboard Display Models 2 (Figure 8-2), 3, and 4 can have a numeric or expanded numeric keyboard (described above) or an alphameric or expanded alphameric keyboard. The alphameric keyboard has 74 keys, made up of three sets: a set of 54 keys engraved with alphamerics and control markings, a set of 12 keys engraved with numerics and three nonengraved keys, and a set of five nonengraved keys. The expanded alphameric keyboard has 94 keys, made up of a set of 54 keys engraved with alphamerics and control markings, a set of 12 keys engraved with numerics and three nonengraved keys, and a set of 25 nonengraved keys. (The alphameric keyboard for the Japanese 3604 keyboard display has 77 keys: a set of 62 keys engraved with alphamerics and a set of 12 keys engraved with numerics and three nonengraved keys. The Japanese expanded alphameric keyboard has 92 keys: a set of 62 keys engraved with alphamerics, a set of 12 keys engraved with numerics and three nonengraved keys, and a set of 15 nonengraved keys.)

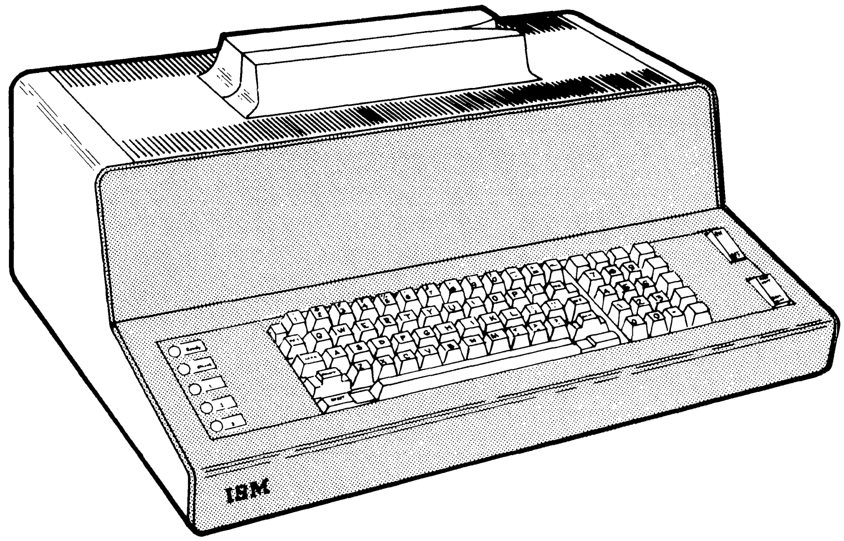


Figure 8-2. IBM 3604 Keyboard Display Model 2

The 3604 Keyboard Display Models 5 and 6 (Figure 8-3) have a 45-key keyboard (Figure 8-4) made up of three sets of 15 keys. Ten of the keys in one of the sets are labeled as numeric keys on an overlay; the other keys can be labeled as desired. A blank plastic overlay that allows all keys to be labeled as desired is also available.

Translation tables defined by the installation determine what character a key represents to a controller application program. If a key translates to a character other than the one engraved on it, that character should be placed on the key by means of an adhesive label. Sets of printed and blank adhesive keytop labels are supplied with the 3604 keyboard display.

If a 3604 keyboard display Models 2, 3, or 4 is the first terminal on a remote subloop, either a 1200-bps loop integrated modem or an external modem such as an IBM 3603 is a prerequisite. (If a 3614/3624 Consumer Transaction Facility is on the same subloop as a 3604 terminal, the 3614/3624 terminal should contain the loop integrated modem. This prevents halting the loop if the 3604 is powered off.)

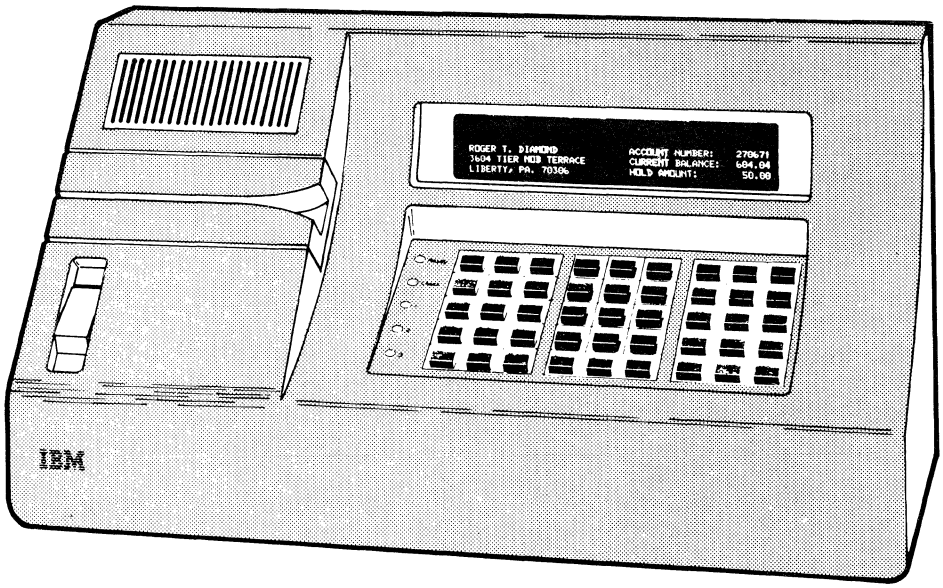
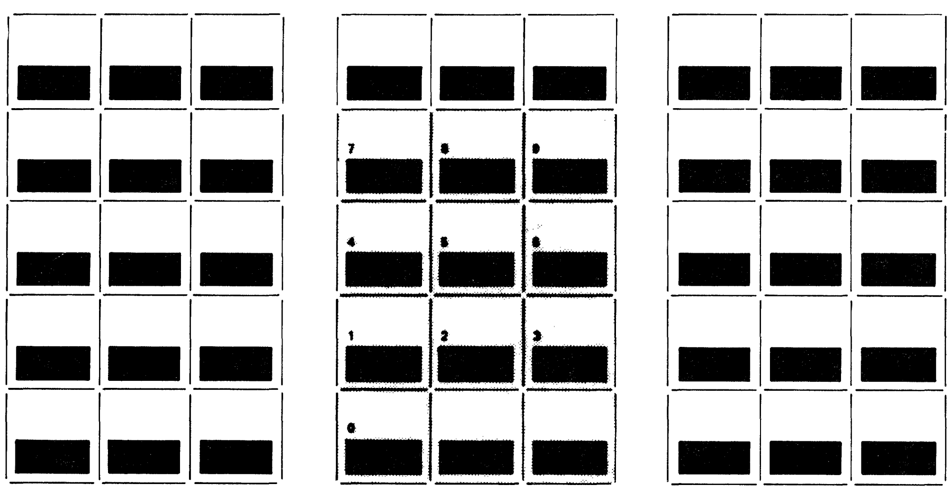


Figure 8-3. IBM 3604 Keyboard Display Models 5 and 6



Note: An overlay may be used to identify the numeric keys in center 15-key set as shown, or in left or right 15-key sets.

Figure 8-4. IBM 3604 Models 5 and 6, 45-Key Alphameric Keyboard

The 3604 Administrative Keyboard Display Model 7 (Figure 8-5) has either a 99-key or 92-key keyboard that is cable-attached to the display enclosure. The 99-key keyboard, available in all countries except Japan, is divided into four clusters: a typewriter arrangement, an adding machine arrangement, and two groups of 15 function keys. The 92-key keyboard, available only in Japan, is divided into three clusters: a typewriter arrangement, an adding machine arrangement, and one group of 15 function keys.

For all five keyboards, the three keys in the bottom row of the numeric keys can be: zero, double zero, and decimal point; or zero, double zero, and triple zero; or zero, triple zero, and decimal point. For the numeric, expanded numeric, and expanded alphameric keyboards, a motor bar, which is raised so that it can be pressed with a roll of the hand, can take the place of the lower three keys in the left column of the function keys.

Sets of preprinted and blank adhesive labels are supplied with the 3604 keyboard display. Protective caps are provided for nonengraved function keys on the Model 7.

No repeat-action capability is available with Models 5, 6, and 7. Scan codes for the 3604 Models 5 and 6 are compatible with the scan code for the 3604 Model 1 expanded numeric keyboard.

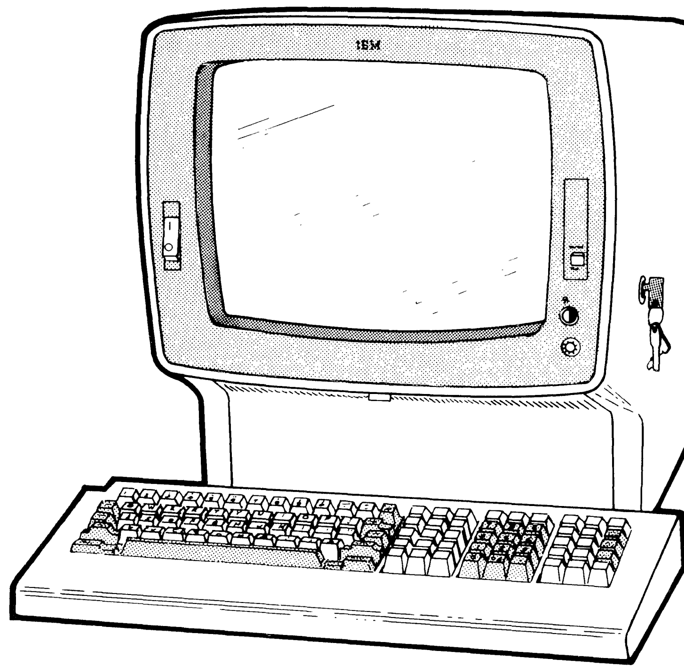


Figure 8-5. IBM 3604 Administrative Keyboard Display, Model 7

Display Component

The display component for Models 1-6 contains a gas discharge panel that displays alphameric characters as follows:

- Models 1 and 2 can display 240 characters in a 6-line by 40-column format.
- Model 3 can display 480 characters in a 12-line by 40-column format.
- Model 4 can display 1024 characters in a 16-line by 24-column format.
- Model 5 can display 120 characters in a 3-line by 40-column format.
- Model 6 can display 240 characters in a 6-line by 40-column format.

These components contain a character generator that can generate up to 153 unique characters. The Model 7 has a TV-type picture tube display that can display 1920 characters in a 24-line by 80-column format.

Special Features and Options

The 3604 keyboard display can have a magnetic stripe reader or a magnetic stripe encoder/reader (Models 1-4 only) as a special feature. The International Standard ISO 3554 specifies the physical and magnetic characteristics for a magnetic stripe. The magnetic stripe has three tracks.

The magnetic reader can read data from track 2 encoded at 75 bits per inch (bpi) in accordance with the American Banking Association (ABA) standard code and at 210 bpi in accordance with the 4-bit IBM Specifications code. The reader/encoder read the same as the reader, and encodes the IBM 210 bpi data on track 2 of a magnetic stripe, but might destroy the data already on tracks 1 and 3. The magnetic stripe can contain data such as a logon identification or an account number, and reading it may reduce the amount of data that has to be entered from the keyboard. (Magnetic stripe identification cards and adhesive magnetic stripe labels are available from IBM).

Models 5, 6, and 7 can have an audible alarm as a special feature. Depending on the setting of four internal switches (set by a customer engineer as required), the alarm can be tied to any of the indicators: 1, 2, 3, and Check. When tied to an indicator, the alarm is set on when the indicator is set on, and set off when the indicator is set off.

The 3604 Models 1-7 have language and power options. The language, which affects the alphamerics engraved on the keys, can be Austrian/German, Belgian, Brazilian/Portuguese, Danish/Norwegian, American/Canadian English, United Kingdom English, Finnish, French, Italian, Japanese (Katakana), Spanish, Spanish-speaking, or Swedish. The power can be 60 hertz (at 100 or 115 volts) or 50 hertz (at 100, 110, 123.5, 220, or 235 volts).

The 3604 Model 7 can have a Security Keylock, on the display enclosure, that limits operation to authorized persons only.

When the keylock is turned off, and no data transmission is in process, the display screen is blank except for the indicator row and the present cursor position if the cursor was visible. If keyboard clicking was previously active, clicking is stopped and data transmission inhibited but the terminal remains available for the reception of messages from the control unit. When the keylock is turned on (unsecured), the screen returns to its original display unless an intervening message from the control unit has altered the original display message. Keyboard clicking is restored, and operations return to normal.

Turning the keylock off when the control unit is sending a message to the terminal also causes the screen to be erased except for the indicator row and the cursor. The message to the terminal continues until normal termination but additional operator input is blocked. When the keylock is turned on, the entire message is displayed unless another message from the control unit has altered it.

Controls and Indicators (3604 Models 1-6)

Power On/Off Switch

This two-position rocker switch with graphic symbols turns power on and off. Use the switch with care, because turning power on and off at one terminal may cause errors at other terminals on the same loop. (If, because of some abnormal condition caused by a loop or other hardware error, the 3604 keyboard display stops responding to the operator's actions, turning the power off, waiting approximately 20 seconds, and turning the power back on should start the keyboard display responding again.)

Dsbl Unit/Test Unit Switch

This three-position rocker switch, which is on only the first 3604 keyboard display on a remote subloop, tests the loop or disables the keyboard display. Its normal (off) position is the center position. The Dsbl Unit/Test Unit switch should be used only as directed by the problem recovery procedures.

Ready Indicator

This light indicates the status of communications on the loop between the 3604 Keyboard Display and the controller. If the Ready indicator is on, communications are normal; if it is off or flashing off and on, there is a communications problem. When the indicator is off, signals from the controller are not arriving at the 3604 terminal. When it is flashing off and on, signals from the terminal are not reaching the controller; the problem is usually at the first terminal that is not flashing (the ready indicator is off). Check for incorrect settings of address and speed switches on all terminals on the loop.

Check Indicator

This light indicates that the keyboard component is in purge mode.

The controller can put the keyboard component into purge mode and light the Check indicator to notify the operator of a keyboard or magnetic stripe reader error. The controller application program, by means of the SIGNAL instructions, can also put the keyboard component into purge mode and light the Check indicator to notify the operator of an error discovered by the program. Press the Reset key to get the keyboard component out of purge mode and turn off the Check indicator.

Indicators 1, 2, and 3

These lights are defined by the installation and can be set on and off by the controller application program by means of the SIGNAL instruction issued for the display component or the magnetic stripe encoder.

On a 3604 Keyboard Display with a magnetic stripe encoder/reader, one of these lights (specified by means of the MSLITE operand of the DEV3604 configuration macro instruction) indicates that the magnetic stripe encoder is ready for encoding. (If, during a write operation to the magnetic stripe encoder, an error occurs before the encoder is ready for encoding, this light may flicker once.)

Log Message Indicator

An indicator for an unsolicited log message can be specified using the MSGLITE operand of the STARTGEN configuration macro instruction. The indicator applies only to the 3604 used during controller load as the control operator's console. Any of the three indicator lights or the Check indicator light may be used. If none is specified, the Check indicator is used. The unsolicited message indicator is reset (turned off) when the control operator reads the log or, if the Check indicator is being used, when the Reset key is pressed after data has been entered.

Controls and Indicators (3604 Model 7)

The display unit has discrete indicators and screen indicators. Controls and discrete indicators are on the display unit housing and the screen indicators are displayed at the bottom of the display screen.

Controls

Normal/Test

This switch logically disconnects the terminal from the system and initiates the test mode sequence.

Audible Alarm Volume Control (Special Feature)

This adjustable control allows the operator to set the loudness of the alarm.

Intensity

This adjustable control allows the operator to set the overall brightness of data displayed on the screen.

Intensity Override

This switch, an integral part of the intensity control, when turned fully clockwise allows the operator to override logic control of the brightness.

Contrast

This adjustable control allows the operator to establish the difference in brightness between high and normal intensity data fields.

Security Keylock (Special Feature)

This key-operated switch allows only authorized persons to operate the terminal.

Power On/Off

This two-position rocker switch, with graphic symbols, turns power on and off. Use the switch with care, because turning power on and off at one terminal may cause errors at other terminals on the same loop.

Indicator Lights

Ready

This indicator, working in conjunction with the display screen ready indicator, indicates loop synchronization.

Light 1

This light indicates operation of the 'sweep' circuits that provide character display on the screen.

Light 2

This light indicates the presence of voltage required to illuminate the display screen.

Light 3

This light indicates the presence of voltages required to operate the display terminal logic.

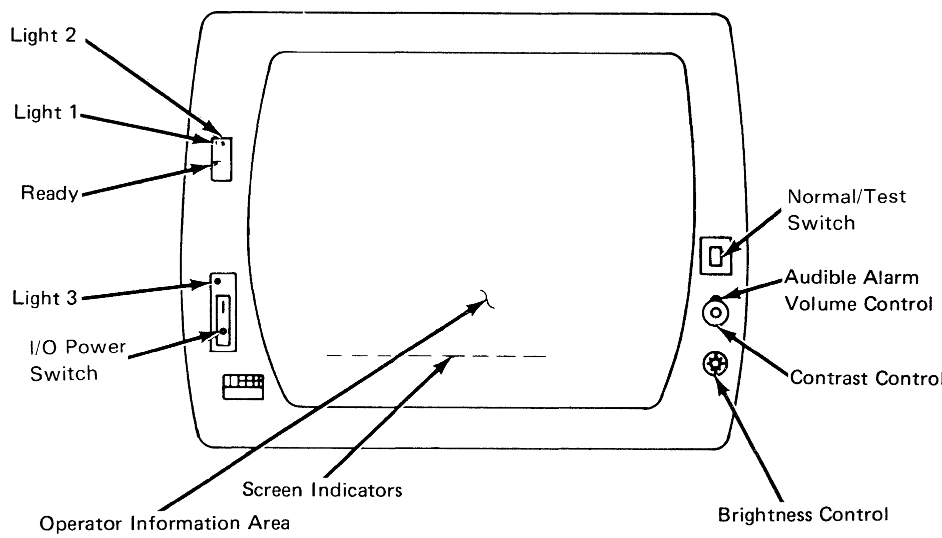


Figure 8-6. 3604-7 Operator's Panel

Screen Indicators

Two nonprogrammable and four programmable indicators appear at the bottom of the display screen. These indicators do not appear when the display is in Test mode.

An indicator that works in conjunction with READY on the display housing indicates loop synchronization.

A programmable indicator corresponds to the 3604 Models 1-6 CHECK indicator. When this indicator is turned on, the functional state of the keyboard clicker is reversed. This indicator can be turned on by a number of conditions:

- By the controller if a keyboard error is detected. This causes the keyboard to be in purge mode.
- By the user application program to stop keyboard input when errors or other conditions are detected by the program. For example: an attempt to enter data into a protected field, a host or subsystem is not ready to accept input, or in combination with other conditions. This causes the keyboard to be in purge mode.
- When the Security Keylock (feature) is in the secured position.
- By the user program to indicate a predefined condition that requires operator action.
- By the controller if the CHECK light is defaulted as the message light indicator (in the STARTGEN macro) signifying that a message has been written into the system log. This applies to the control station only.

A nonprogrammable indicator shows that the Security Keylock is in the secured (off) position.

Programmable indicators 1, 2, and 3 are defined by the user according to specific requirements.

Controller Configuration Macro Instructions

Controller configuration macro instructions are used to specify the physical and logical configuration of each 4701 and its associated terminals. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. Refer to Volume 6 for a coding information and a description of the DEV3604 and PINTBL macros.

Terminal Control Characters

Control characters are embedded in the data stream and indicate operations to be performed at a terminal. The following are the control characters for the 3604:

Function	Control Character	Explanation
Line Feed	X'25'	The cursor is moved down one line; the horizontal position is unchanged.
Return	X'0D'	The cursor moves to position 1 on the current display line.
New Line	X'15'	The current line is erased from the cursor to the end of the line. The cursor moves to position 1 of the next lower line.
New Page	X'0C'	The screen is completely erased; the cursor is then set to position 1 of of line 1. If a magnetic stripe encoder on the 3604 is ready to use, a write to the encoder with a first byte of X'0C' in the message causes the encoder to become not ready. Reset should not be issued for an encoder that is not ready to use (not in the encode state).
Set Attribute	X'28'	This 3-byte control string sends one byte to the display. The second byte should be X'C1'. The third byte may be any of the following values: X'40' Normal intensity X'C4' Normal intensity X'C8' Intensified X'4C' Nondisplay For 3604 Model 7, the byte sent to the display is the attribute byte (which is displayed as a space) and assigns the given characteristics to all characters on the display between this attribute byte and the next. The next attribute may be found via display wrap from the last character of the last line (excluding the indicator row) to the first character of the first line, and continuing the search for the attribute. For 3604 Models 1-6, the byte sent to the display is position 32 (which displays as a blank). This byte has no effect on the characteristics of the display characters that follow.

Function	Control Character	Explanation
Position	X'34'	This control character is followed by a flag byte and a 1-byte positioning value. Flag byte: Bits 0-3 are ignored. Bit 4: 0 = Position is an absolute value. 1 = Position is relative to the present postion. Bit 5: 0 = Move is horizontal (column). 1 = Move is vertical (line). Bits 6-7: 00 = Move cursor to new position without erasing. 01 = Leave cursor at the current position, but erase from the current position the number of bytes specified by the position. 10 = Move cursor to new position and erase from the old position up to, but not including, the new position. 11 = Reserved. The positioning byte indicates a line or column number. It contains an unsigned binary value ranging from 0 to 255 to be used under direction of the flag byte; a 0 results in no operation. The value specified is divided by a modulus number to determine the resulting position. The modulus division produces a change of 1 to x lines for vertical positioning, and a change of 1 to y columns for horizontal positioning, where x is the maximum number of lines on a screen, and y is the maximum number of columns in a line.
Select	X'04'	The automatic new line action is suppressed. The next byte selects direction: X'00' selects a forward, left to right direction; X'01' selects backward, right to left direction. Use X'0401' with the backward-mode RPQ.

Function	Control Character	Explanation
Transparent Write	X'35'	A transparent write (a write without translation) begins. The format of this control sequence is: Byte 0: The control character, X'35'. Byte 1: The number of display positions to be displayed. Byte 2-Byte n: The position to be displayed. The valid positions are X'00' - X'AF'.

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or they may be restarted, or "wrapped back," to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 3604:

Keyboard

Counter	Explanation
1	Loop error checks
2	Terminal address card unit checks
3	Device errors (keyboard or magnetic stripe reader)
4	Translation checks
5	Segment overruns

Display

Counter	Explanation
1	Loop error checks
2	Terminal address card unit checks

Magnetic Stripe Encoder/Reader

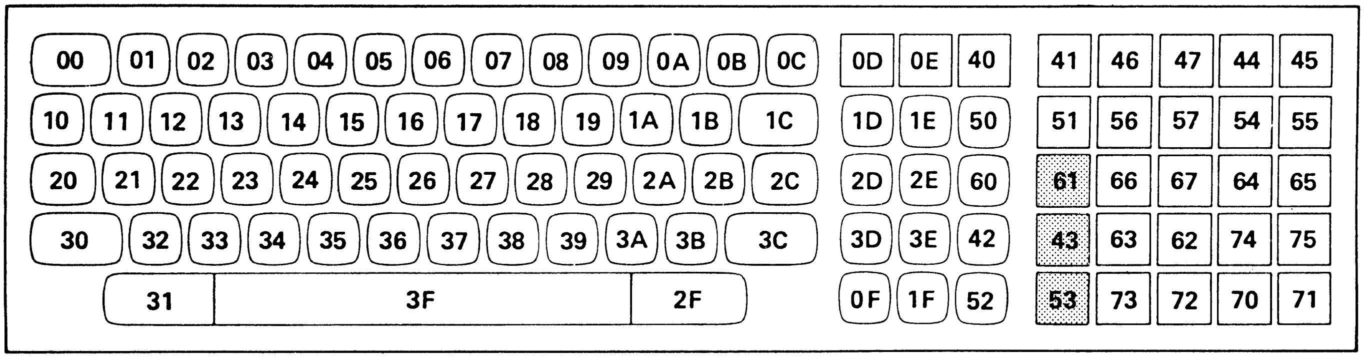
Counter	Explanation
1	Loop error checks
2	Terminal address card unit checks
3	Device error (format error or encoding failure)

Keyboards and Translation Tables

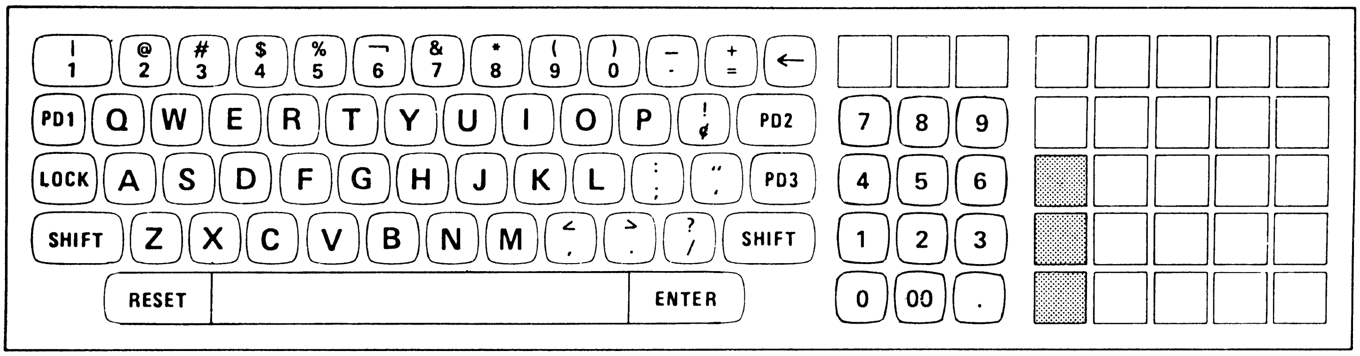
The next few pages contain the 3604 keyboards and the standard input translation tables for them. The keyboards show the key scan-code scheme for the 3600 keyboard devices and the engraved keys for keyboards 4661, 4662, 4663, and 4664.

The keyboards are:

- 4664 -- 94-key
- 4663 -- 45-key
- 4661 -- 30-key
- 4662 -- 74-key
- 4664J-- 94-key Katakana
- 4662J-- 77-key Katakana
- 36047-- 99-key Model 7
- 3604K-- 99-key



Scan Codes



Engraved Keys

Notes: Three of the keys are engraved with: zero, double zero, and decimal point (as shown); or zero, double zero, and triple zero; or zero, triple zero, and decimal point.

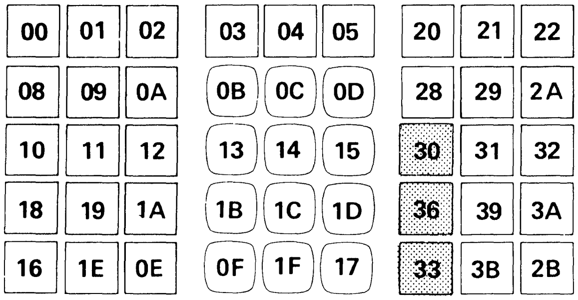
For the expanded alphanumeric keyboard, a motor bar (indicated by the shading) can take the place of three of the nonengraved keys.

94-Key Keyboard (4664)

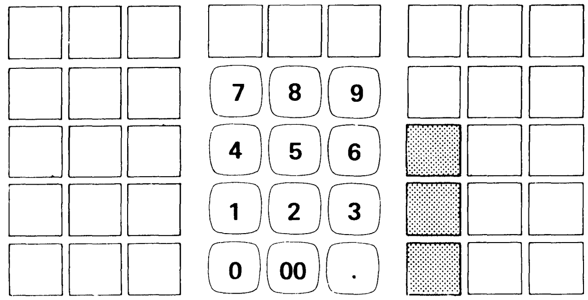
4664/4774 94-Key Keyboard

Scan Code	*Case 1*	*Scan Code*	*Case 1*
00	C'1'	2F	EOM
01	C'2'	30	NOOP
02	C'3'	31	NOOP
03	C'4'	32	C'Z'
04	C'5'	33	C'X'
05	C'6'	34	C'C'
06	C'7'	35	C'V'
07	C'8'	36	C'B'
08	C'9'	37	C'N'
09	C'0'	38	C'M'
0A	X'60'	39	X'6B'
0B	X'7E'	3A	X'4B'
0C	BKSP	3B	X'61'
0D	NOOP	3C	NOOP
0E	NOOP	3D	C'1'
0F	C'0'	3E	C'2'
10	NOOP	3F	X'40'
11	C'Q'	40	NOOP
12	C'W'	41	NOOP
13	C'E'	42	C'3'
14	C'R'	43	NOOP
15	C'T'	44	
16	C'Y'	45	
17	C'U'	46	
18	C'I'	47	NOOP
19	C'O'	50	C'9'
1A	C'P'	51	NOOP
1B	X'4A	52	NOOP
1C	NOOP	53	EOM
1D	C'7'	54	NOOP
1E	C'8'	55	
1F	NOOP	56	
20	NOOP	57	NOOP
21	C'A'	60	C'6'
22	C'S'	61	NOOP
23	C'D'	62	
24	C'F'	63	
25	C'G'	64	
26	C'H'	65	
27	C'J'	66	
28	C'K'	67	
29	C'L'	70	
2A*	X'5E'	71	
2B	X'7D'	72	
2C	NOOP	73	
2D	C'4'	74	
2E	C'5'	75	NOOP

Note: Case 2 for the 94-key keyboard is the same as case 2 for the 74-key keyboard; the additional 20 keys of the 94-key keyboard are set to NOOP.

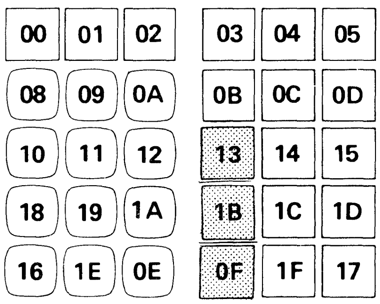


Scan Codes

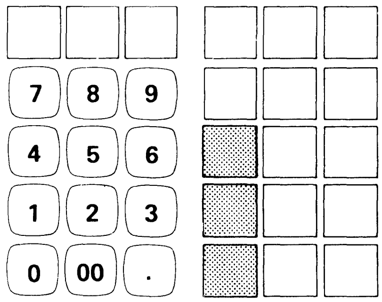


Engraved Keys

45-Key Keyboard (4663)



Scan Codes



Engraved Keys

30-Key Keyboard (4661)

Notes: Three of the keys are engraved with: zero, double zero, and decimal point (as shown); or zero, double zero, and triple zero; or zero, triple zero, and decimal point.

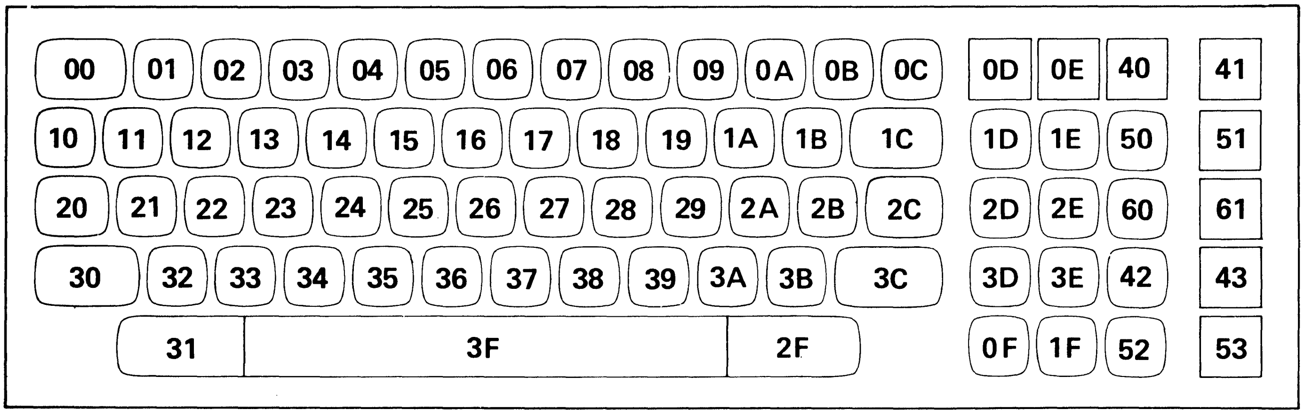
A motor bar (indicated by the shading) can take the place of three of the nonengraved keys.

4661/4771 30-Key Keyboard

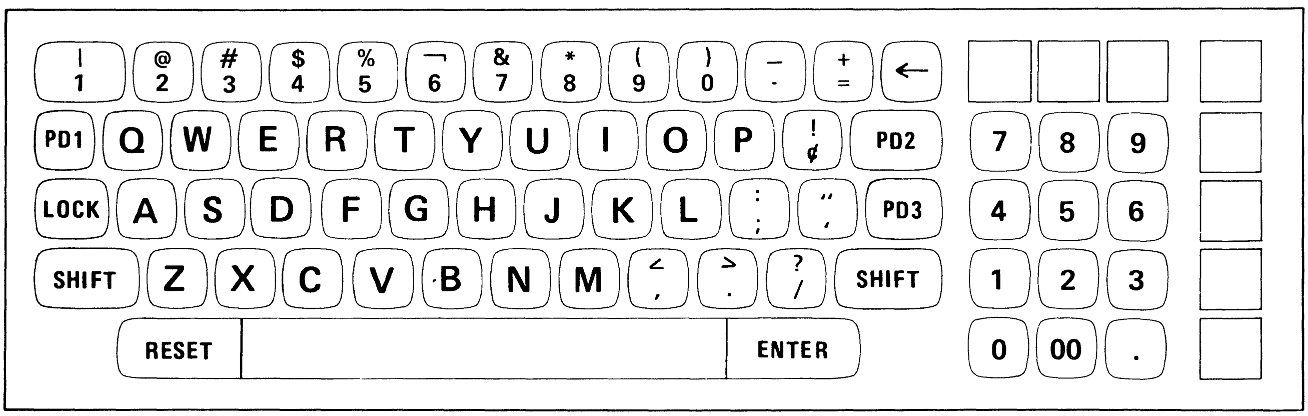
Scan Code	*Case 1*
00	NOOP
01	
02	
03	
04	
05	NOOP
08	X'F7'
09	X'F8'
0A	X'F9'
0B	NOOP
0C	
0D	
0E	
0F	NOOP
10	X'F4'
11	X'F5'
12	X'F6'
13	NOOP
14	
15	NOOP
16	X'F0'
17	EOM
18	X'F1'
19	X'F2'
1A	X'F3'
1B	NOOP
1C	
1D	
1E	
1F	NOOP

4663/4773 45-Key Keyboard

Scan Code	*Case 1*
00	NOOP
01	
02	
03	
04	
05	
08	
09	
0A	NOOP
0B	C'7'
0C	C'8'
0D	C'9'
0E	NOOP
0F	C'0'
10	NOOP
11	
12	NOOP
13	C'4'
14	C'5'
15	C'6'
16	NOOP
17	
18	
19	
1A	NOOP
1B	C'1'
1C	C'2'
1D	C'3'
1E	NOOP
1F	
20	
21	
22	
28	
29	
2A	NOOP
2B	EOM
30	NOOP
31	
32	
33	
38	
39	
3A	
3B	NOOP



Scan Codes

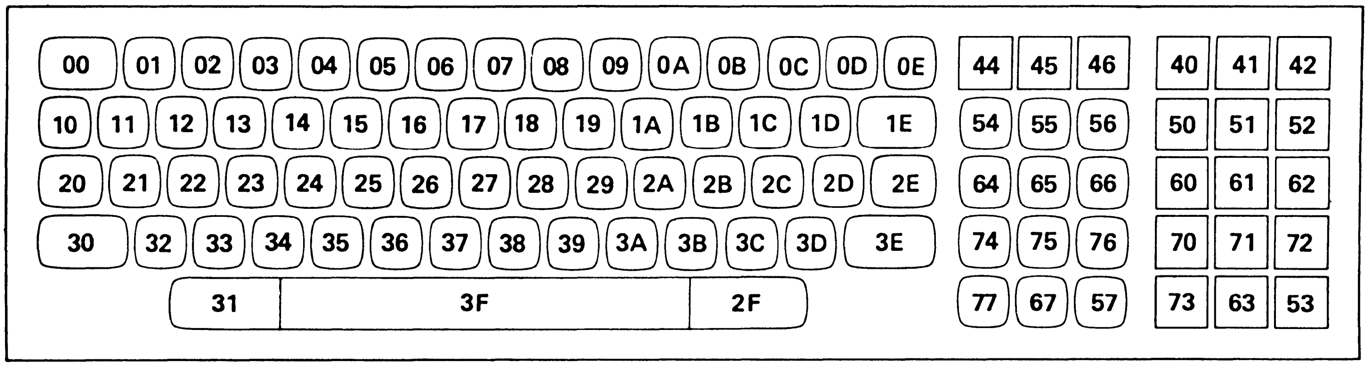


Engraved Keys

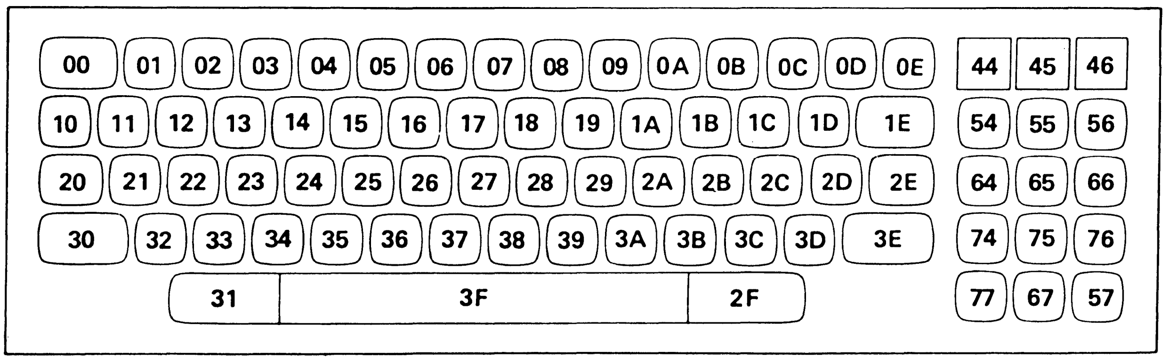
74-Key Keyboard (4662)

4662/4772 74-Key Keyboard

Scan Code	*Case 1*	*Case 2*	*Scan Code*	*Case 1*	*Case 2*
00	C'1'	X'4F'	25	C'G'	C'G'
01	C'2'	X'7C'	26	C'H'	C'H'
02	C'3'	X'7B'	27	C'J'	C'J'
03	C'4'	X'5B'	28	C'K'	C'K'
04	C'5'	X'6C'	29	C'L'	C'L'
05	C'6'	X'5F'	2A	X'5E'	X'7A'
06	C'7'	X'50'	2B	X'7D'	X'7F'
07	C'8'	X'5C'	2C	NOOP	NOOP
08	C'9'	X'4D'	2D	C'4'	C'4'
09	C'0'	X'5D'	2E	C'5'	C'5'
0A	X'60'	X'6D'	2F	EOM	EOM
0B	X'7E'	X'4E'	30	NOOP	NOOP
0C	BKSP	BKSP	31	NOOP	NOOP
0D	NOOP	NOOP	32	C'Z'	C'Z'
0E	NOOP	NOOP	33	C'X'	C'X'
0F	C'0'	C'0'	34	C'C'	C'C'
10	NOOP	NOOP	35	C'V'	C'V'
11	C'Q'	C'Q'	36	C'B'	C'B'
12	C'W'	C'W'	37	C'N'	C'N'
13	C'E'	C'E'	38	C'M'	C'M'
14	C'R'	C'R'	39	X'6B'	X'4C'
15	C'T'	C'T'	3A	X'4B'	X'6E'
16	C'Y'	C'Y'	3B	X'61'	X'6F'
17	C'U'	C'U'	3C	NOOP	NOOP
18	C'I'	C'I'	3D	C'1'	C'1'
19	C'O'	C'O'	3E	C'2'	C'2'
1A	C'P'	C'P'	3F	X'40'	X'40'
1B	X'4A'	X'5A'	40	NOOP	NOOP
1C	NOOP	NOOP	41	NOOP	NOOP
1D	C'7'	C'7'	42	C'3'	C'3'
1E	C'8'	C'8'	43	NOOP	NOOP
1F	NOOP	NOOP	50	C'9'	C'9'
20	NOOP	NOOP	51	NOOP	NOOP
21	C'A'	C'A'	52	NOOP	NOOP
22	C'S'	C'S'	53	EOM	EOM
23	C'D'	C'D'	60	C'6'	C'6'
24	C'F'	C'F'	61	NOOP	NOOP



92-Key Keyboard (4664J)



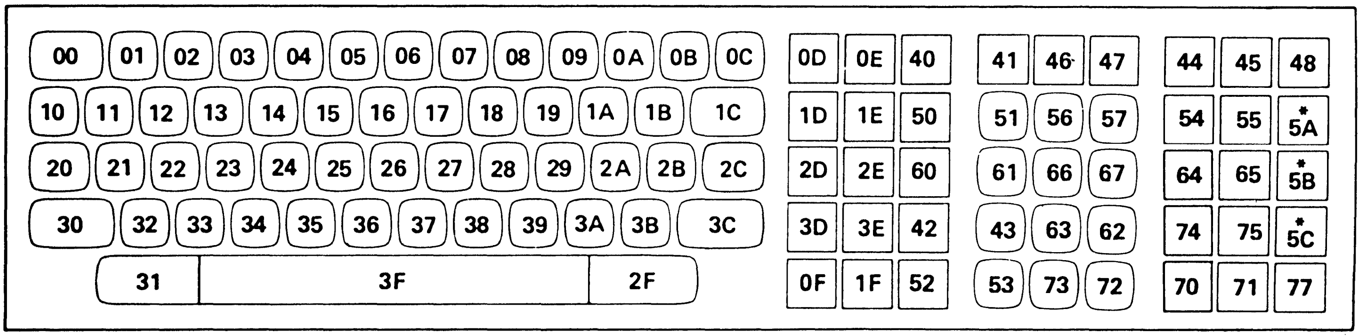
77-Key Keyboard (4662J)

4662J 77-Key Keyboard

Scan Code	*Case 1*	*Case 2*	*Case 3*	*Case 4*
00	C'1'	X'5A'	X'98'	NOOP
01	C'2'	X'7F'	X'88'	NOOP
02	C'3'	X'7B'	X'81'	X'47'
03	C'4'	X'5B'	X'83'	X'49'
04	C'5'	X'6C'	X'84'	X'51'
05	C'6'	X'50'	X'85'	X'52'
06	C'7'	X'7D'	X'A9'	X'53'
07	C'8'	X'4D'	X'AA'	X'54'
08	C'9'	X'5D'	X'AC'	X'55'
09	C'0'	X'5B'	X'BC'	NOOP
0A	X'60'	X'7E'	X'A3'	↓
0B	X'5F'	X'A1'	X'A2'	↓
0C	X'5B'	X'4F'	X'58'	NOOP
0D	BKSP	BKSP	BKSP	BKSP
0E	NOOP	NOOP	NOOP	NOOP
10	NOOP	↓	NOOP	↓
11	C'Q'	↓	X'91'	↓
12	C'W'	↓	X'94'	NOOP
13	C'E'	↓	X'82'	X'48'
14	C'R'	↓	X'8E'	NOOP
15	C'T'	↓	X'86'	↓
16	C'Y'	↓	X'BD'	↓
17	C'U'	↓	X'96'	↓
18	C'I'	↓	X'97'	↓
19	C'O'	↓	X'AD'	↓
1A	C'P'	↓	X'8F'	↓
1B	X'7C'	↓	X'BE'	NOOP
1C	NOOP	↓	X'BF'	X'42'
1D	↓	↓	NOOP	NOOP
1E	↓	↓	↓	↓
20	NOOP	↓	NOOP	↓
21	C'A'	↓	X'92'	↓
22	C'S'	↓	X'95'	↓
23	C'D'	↓	X'8D'	↓
24	C'F'	↓	X'9D'	↓
25	C'G'	↓	X'87'	↓
26	C'H'	↓	X'88'	↓
27	C'J'	↓	X'A4'	↓
28	C'K'	NOOP	X'9A'	NOOP

Scan Code	*Case 1*	*Case 2*	*Case 3*	*Case 4*
29	C'L'	NOOP	X'AE'	NOOP
2A	X'5E'	X'4E'	X'BA'	↓
2B	X'7A'	X'5C'	X'89'	NOOP
2C	NOOP	NOOP	X'A6'	X'43'
2D	↓	↓	NOOP	NOOP
2E	NOOP	NOOP	NOOP	NOOP
2F	EOM	EOM	EOM	EOM
30	NOOP	NOOP	NOOP	NOOP
31	NOOP	↓	NOOP	↓
32	C'Z'	↓	X'93'	↓
33	C'X'	↓	X'8C'	↓
34	C'C'	↓	X'90'	↓
35	C'V'	↓	X'9E'	↓
36	C'B'	↓	X'8A'	↓
37	C'N'	NOOP	X'A5'	NOOP
38	C'M'	NOOP	X'A8'	NOOP
39	X'6B'	X'4C'	X'99'	X'44'
3A	X'4B'	X'6E'	X'AF'	X'41'
3B	X'61'	X'6F'	X'A7'	X'45'
3C	NOOP	X'6D'	X'BB'	NOOP
3D	↓	NOOP	NOOP	↓
3E	NOOP	NOOP	NOOP	NOOP
3F	X'40'	X'40'	X'40'	X'40'
44	NOOP	NOOP	NOOP	NOOP
45	↓	↓	↓	↓
46	↓	↓	↓	↓
54	↓	↓	↓	↓
55	↓	↓	↓	↓
56	↓	↓	↓	↓
57	↓	↓	↓	↓
64	↓	↓	↓	↓
65	↓	↓	↓	↓
66	↓	↓	↓	↓
67	↓	↓	↓	↓
74	↓	↓	↓	↓
75	↓	↓	↓	↓
76	↓	↓	↓	↓
77	NOOP	NOOP	NOOP	NOOP

Note: The key assignments for the 92-key keyboard (4664J) are the same as those for the 77-key keyboard with the addition of the keys with scan codes 40-42, 50-52, 60-62, 70-73, 63 and 53, which are set to NOOP in all cases.



Scan Codes

*Scan Codes 5A, 5B, and 5C are used to control internal device function, and being fixed in their definition are not available to the programmer for redefinition.

99-Key Keyboard (36047 and 36047K)

36047 Keyboard

Scan Code	*Case 1*	*Case 2*	*Scan Code*	*Case 1*	*Case 2*
00	C'1'	X'4F'	30	NOOP	NOOP
01	C'2'	X'7C'	31	NOOP	NOOP
02	C'3'	X'7B'	32	C'z'	C'Z'
03	C'4'	X'5B'	33	C'x'	C'X'
04	C'5'	X'6C'	34	C'c'	C'C'
05	C'6'	X'5F'	35	C'v'	C'V'
06	C'7'	X'50'	36	C'b'	C'B'
07	C'8'	X'5C'	37	C'n'	C'N'
08	C'9'	X'4D'	38	C'm'	C'M'
09	C'0'	X'5D'	39	X'6B'	X'4C'
0A	X'60'	X'6D'	3A	X'4B'	X'6E'
0B	X'7E'	X'4E'	3B	X'61'	X'6F'
0C	BKSP	BKSP	3C	NOOP	NOOP
0D	NOOP	NOOP	3D	NOOP	NOOP
0E	NOOP	NOOP	3E	NOOP	NOOP
0F	NOOP	NOOP	3F	X'40'	X'40'
10	NOOP	NOOP	40	NOOP	NOOP
11	C'q'	C'Q'	41	NOOP	NOOP
12	C'w'	C'W'	42	NOOP	NOOP
13	C'e'	C'E'	43	C'1'	C'1'
14	C'r'	C'R'	44	NOOP	NOOP
15	C't'	C'T'	45	NOOP	NOOP
16	C'y'	C'Y'	46	NOOP	NOOP
17	C'u'	C'U'	47	NOOP	NOOP
18	C'i'	C'I'	48	NOOP	NOOP
19	C'o'	C'O'	50	NOOP	NOOP
1A	C'p'	C'P'	51	C'7'	C'7'
1B	X'4A'	X'5A'	52	NOOP	NOOP
1C	NOOP	NOOP	53	C'0'	C'0'
1D	NOOP	NOOP	54	NOOP	NOOP
1E	NOOP	NOOP	55	NOOP	NOOP
1F	NOOP	NOOP	56	C'8'	C'8'
20	NOOP	NOOP	57	C'9'	C'9'
21	C'a'	C'A'	60	NOOP	NOOP
22	C's'	C'S'	61	C'4'	C'4'
23	C'd'	C'D'	62	C'3'	C'3'
24	C'f'	C'F'	63	C'2'	C'2'
25	C'g'	C'G'	64	NOOP	NOOP
26	C'h'	C'H'	65	NOOP	NOOP
27	C'j'	C'J'	66	C'5'	C'5'
28	C'k'	C'K'	67	C'6'	C'6'
29	C'l'	C'L'	70	NOOP	NOOP
2A	X'5E'	X'7A'	71	NOOP	NOOP
2B	X'7D'	X'7F'	72	NOOP	NOOP
2C	NOOP	NOOP	73	NOOP	NOOP
2D	NOOP	NOOP	74	NOOP	NOOP
2E	NOOP	NOOP	75	NOOP	NOOP
2F	EOM	EOM	77	NOOP	NOOP

36047K Keyboard

Scan Code	*Case 1*	*Case 2*	*Case 3*	*Case 4*
00	C'1'	X'5A'	X'98'	NOOP
01	C'2'	X'7F'	X'88'	NOOP
02	C'3'	X'7B'	X'81'	X'47'
03	C'4'	X'E0'	X'83'	X'49'
04	C'5'	X'6C'	X'84'	X'51'
05	C'6'	X'50'	X'85'	X'52'
06	C'7'	X'7D'	X'A9'	X'53'
07	C'8'	X'4D'	X'AA'	X'54'
08	C'9'	X'5D'	X'AC'	X'55'
09	C'0'	X'4A'	X'BC'	X'46'
0A	X'60'	X'7E'	X'A3'	NOOP
0B	X'5F'	X'A1'	X'A2'	NOOP
0C	X'5B'	X'4F'	X'58'	NOOP
0D	BKSP	BKSP	BKSP	BKSP
0E	NOOP	NOOP	NOOP	NOOP
10	NOOP	NOOP	NOOP	NOOP
11	C'Q'	NOOP	X'91'	NOOP
12	C'W'	NOOP	X'94'	NOOP
13	C'E'	NOOP	X'82'	X'48'
14	C'R'	NOOP	X'8E'	NOOP
15	C'T'	NOOP	X'86'	NOOP
16	C'Y'	NOOP	X'BD'	NOOP
17	C'U'	NOOP	X'96'	NOOP
18	C'I'	NOOP	X'97'	NOOP
19	C'O'	NOOP	X'AD'	NOOP
1A	C'P'	NOOP	X'8F'	NOOP
1B	X'7C'	NOOP	X'BE'	NOOP
1C	NOOP	NOOP	X'BF'	X'42'
1D	NOOP	NOOP	NOOP	NOOP
1E	NOOP	NOOP	NOOP	NOOP
20	NOOP	NOOP	NOOP	NOOP
21	C'A'	NOOP	X'92'	NOOP
22	C'S'	NOOP	X'95'	NOOP
23	C'D'	NOOP	X'8D'	NOOP
24	C'F'	NOOP	X'9D'	NOOP
25	C'G'	NOOP	X'87'	NOOP
26	C'H'	NOOP	X'88'	NOOP
27	C'J'	NOOP	X'A4'	NOOP
28	C'K'	NOOP	X'9A'	NOOP
29	C'L'	NOOP	X'AE'	NOOP
2A	X'5E'	X'4E'	X'BA'	NOOP
2B	X'7A'	X'5C'	X'89'	NOOP
2C	NOOP	NOOP	X'A6'	X'43'
2D	NOOP	NOOP	NOOP	NOOP
2E	NOOP	NOOP	NOOP	NOOP
2F	EOM	EOM	EOM	EOM

Scan Code	*Case 1*	*Case 2*	*Case 3*	*Case 4*
30	NOOP	NOOP	NOOP	NOOP
31	NOOP	NOOP	NOOP	NOOP
32	C'Z'	NOOP	X'93'	X'56'
33	C'X'	NOOP	X'8C'	NOOP
34	C'C'	NOOP	X'90'	NOOP
35	C'V'	NOOP	X'9E'	NOOP
36	C'B'	NOOP	X'8A'	NOOP
37	C'N'	NOOP	X'A5'	NOOP
38	C'M'	NOOP	X'A8'	NOOP
39	X'6B'	X'4C'	X'99'	X'44'
3A	X'4B'	X'6E'	X'AF'	X'41'
3B	X'61'	X'6F'	X'A7'	X'45'
3C	NOOP	X'6D'	X'BB'	NOOP
3D	NOOP	NOOP	NOOP	NOOP
3E	NOOP	NOOP	NOOP	NOOP
3F	X'40'	X'40'	X'40'	X'40'
44	NOOP	NOOP	NOOP	NOOP
45	NOOP	NOOP	NOOP	NOOP
46	NOOP	NOOP	NOOP	NOOP
50	NOOP	NOOP	NOOP	NOOP
51	NOOP	NOOP	NOOP	NOOP
52	NOOP	NOOP	NOOP	NOOP
53	NOOP	NOOP	NOOP	NOOP
54	C'7'	C'7'	C'7'	C'7'
55	C'8'	C'8'	C'8'	C'8'
56	C'9'	C'9'	C'9'	C'9'
57	NOOP	NOOP	NOOP	NOOP
60	NOOP	NOOP	NOOP	NOOP
61	NOOP	NOOP	NOOP	NOOP
62	NOOP	NOOP	NOOP	NOOP
63	NOOP	NOOP	NOOP	NOOP
64	C'4'	C'4'	C'4'	C'4'
65	C'5'	C'5'	C'5'	C'5'
66	C'6'	C'6'	C'6'	C'6'
67	NOOP	NOOP	NOOP	NOOP
70	NOOP	NOOP	NOOP	NOOP
71	NOOP	NOOP	NOOP	NOOP
72	NOOP	NOOP	NOOP	NOOP
73	NOOP	NOOP	NOOP	NOOP
74	C'1'	C'1'	C'1'	C'1'
75	C'2'	C'2'	C'2'	C'2'
76	C'3'	C'3'	C'3'	C'3'
77	C'0'	C'0'	C'0'	C'0'

The tables below are the standard output translation tables for the 3600 displays and printers. Characters that do not have corresponding hexadecimal values in these tables are nonstandard. To refer to these characters, you must either give each of them a unique hexadecimal value when it is specified in the OUTRTBL macro, or use a transparent write operation.

For Displays, 175-Character Set

Position	*Hex Value*	*Character*	
0	4A	¢	cent sign
1	4F	|	logical OR
2	5F	¬	logical NOT
3		£	pound sign
4		Ä	A-umlaut
5		Ö	O-umlaut
6		Ü	U-umlaut
7		Ã	A-tilde
8		Ñ	N-tilde
9		Õ	O-tilde
10		Ç	C-cedilla
11		Pt	peseta (PTS)
12		Å	angstrom
13		/⊏	dipthong
14		∅	
15		¥	yen sign
16	A1	‾	overline
17		{	opening brace
18		}	closing brace
19	41	。	Kana period
20	42	「	Kana left bracket
21	43	」	Kana right bracket
22	44	、	Kana comma
23	45	・	center period
24	BE	゛	Kana sonant
25	BF	゜	Kana explosive
26	81	ア	Kana (a)
27	82	イ	Kana (i)
28	83	ウ	Kana (u)
29	84	エ	Kana (e)
30	85	オ	Kana (o)
31	86	カ	Kana (ka)
32	40		space (blank)
33	5A	!	exclamation mark
34	7F	"	quotation marks
35	7B	#	number sign
36	5B	$	dollar sign
37	6C	%	percent sign
38	50	&	ampersand (and sign)
39	7D	'	apostrophe
40	4D	(	left parenthesis
41	5D	)	right parenthesis
42	5C	*	asterisk
43	4E	+	plus sign
44	6B	,	comma
45	60	-	minus sign
46	4B	.	period
47	61	/	divide sign (slash)
48	F0	0	number-zero
49	F1	1	number-one
50	F2	2	number-two
51	F3	3	number-three
52	F4	4	number-four
53	F5	5	number-five
54	F6	6	number-six
55	F7	7	number-seven
56	F8	8	number-eight
57	F9	9	number-nine
58	7A	:	colon
59	5E	;	semi-colon
60	4C	<	less-than sign
61	7E	=	equal sign
62	6E	>	greater-than sign
63	6F	?	question mark
64	7C	@	at sign
65	C1	A	Roman characters
66	C2	B	
67	C3	C	
68	C4	D	
69	C5	E	
70	C6	F	
71	C7	G	
72	C8	H	
73	C9	I	
74	D1	J	
75	D2	K	
76	D3	L	
77	D4	M	
78	D5	N	
79	D6	O	
80	D7	P	
81	D8	Q	
82	D9	R	
83	E2	S	
84	E3	T	
85	E4	U	
86	E5	V	
87	E6	W	
88	E7	X	
89	E8	Y	
90	E9	Z	Roman characters
91	C0	[	left bracket
92	E0	\	reverse slant
93	D0	]	right bracket
94		^	upper-arrow sign
95	6D	_	underscore
96	87	キ	Kana (ki)
97	88	ク	Kana (ku)
98	89	ケ	Kana (ke)
99	8A	コ	Kana (ko)

For Displays, 175-Character Set (continued)

Position	*Hex Value*	*Character*
100	8C	サ Kana (sa)
101	8D	シ Kana (si)
102	8E	ス Kana (su)
103	8F	セ Kana (se)
104	90	ソ Kana (so)
105	91	タ Kana (ta)
106	92	チ Kana (chi)
107	93	ツ Kana (tsu)
108	94	テ Kana (te)
109	95	ト Kana (to)
110	96	ナ Kana (na)
111		CHK-1 Check character-1
112		Unused code
113		
114		
115		
116		
117		
118		
119		
120		
121		
122		
123		
124		
125		Unused code
126		CHK-2 Check character-2
127		Unused code
128	97	ニ Kana (ni)
129	98	ヌ Kana (nu)
130	99	ネ Kana (ne)
131	9A	ノ Kana (no)
132	9D	ハ Kana (ha)
133	9E	ヒ Kana (hi)
134	9F	フ Kana (fu)
135		Unused code
136	A2	ヘ Kana (he)
137	A3	ホ Kana (ho)
138	A4	マ Kana (ma)
139	A5	ミ Kana (mi)
140	A6	ム Kana (mu)
141	A7	メ Kana (me)
142	A8	モ Kana (mo)
143		Unused code
144	A9	ヤ Kana (ya)
145	AA	ユ Kana (yu)
146	AC	ヨ Kana (yo)
147	AD	ラ Kana (ra)
148	AE	リ Kana (ri)
149	AF	ル Kana (ru)
150	BA	レ Kana (re)
151		Unused code
152	BB	ロ Kana (ro)
153	BC	ワ Kana (wa)

Position	*Hex Value*	*Character*	
154	46	ヲ Kana (wo)	
155	BD	ン Kana (n)	
156	47	ァ Kana (Sm. a)	
157	48	ィ Kana (Sm. i)	
158	49	ゥ Kana (Sm. u)	
159		Unused code	
160	51	ェ Kana (Sm. e)	Unused for Model 4
161	52	ォ Kana (Sm. o)	
162	53	ャ Kana (Sm. ya)	
163	54	ュ Kana (Sm. yu)	
164	55	ョ Kana (Sm. yo)	
165	56	ッ Kana (Sm. tsu)	
166		— Kana long vowel	
167		Unused code	
168			
169		Unused code	
170		f lower case F	
171		l lower case L	
172		1 European one	
173		7 European seven	
174		¿ Spanish question mark	
175		Unused code	

For Displays, 16-Character Set

Position	*Hex Value*	*Character*
0	F0	0
1	F1	1
2	F2	2
3	F3	3
4	F4	4
5	F5	5
6	F6	6
7	F7	7
8	F8	8
9	F9	9
A	C1	
B	C2	
C	C3	
D	C4	
E	C5	
F	C6	blank

For 3604 Model 7 228E-Character Set

Position	*Hex (Char)*	*Character*	*Position*	*Hex (Char)*	*Character*
0	4A	¢	50	F2	2
1	4F	|	41	F3	3
2	5F	¬	52	F4	4
3	B1	£	53	F5	5
4		Ä	54	F6	6
5		Ö	55	F7	7
6		Ü	56	F8	8
7		Ã	57	F9	9
8		Ñ	58	7A	:
9		Õ	59	5E	;
10	68	Ç	60	4C	<
11	B3	Pt	61	7E	=
12	67	Å	62	6E	>
13		unused	63	6F	?
14	80	ϕ	64	7C	@
15	B2	¥	65	C1	A
16	A1	‾	66	C2	B
17		{	67	C3	C
18		}	68	C4	D
19		C	69	C5	E
20	63	Ä	70	C6	F
21	73	Ë	71	C7	G
22	77	Ï	72	C8	H
23	EC	Ö	73	C9	I
24		U	74	D1	J
25		Y	75	D2	K
26		à	76	D3	L
27		è	77	D4	M
28		ì	78	D5	N
29		ò	79	D6	O
30		ù	80	D7	P
31	46	ã	81	D8	Q
32	40	space	82	D9	R
33	5A	!	83	E2	S
34	7F	"	84	E3	T
35	7B	#	85	E4	U
36	5B	$	86	E5	V
37	6C	%	87	E6	W
38	50	&	88	E7	X
39	7D	'	89	E8	Y
40	4D	(	90	E9	Z
41	5D	)	91	C0	[
42	5C	*	92	E0	\
43	4E	+	93	D0	]
44	6B	,	94	BA	^
45	60	—	95	6D	—
46	4B	.	96	CF	õ
47	61	/	97	DF	ÿ
48	F0	0	98	44	à
49	F1	1	99	54	è

For 3604 Model 7 228E-Character Set (continued)

Position	*Hex (Char)*	*Character*	*Position*	*Hex (Char)*	*Character*
100	51	é	151	99	r
101	58	ì	152		E
102	CD	ò	153		I
103	DD	ù	154	FC	Ü
104	DC	ü	155		O
105	48	ç	156	62	Â
106	43	ä	157	72	Ê
107	51	ë	158	76	Î
108	57	ï	159	A2	s
109	CC	ö	160	EB	Ô
110		ü	161	FB	Û
111		unused	162	65	Á
112	81	a	163	71	É
113	82	b	164	75	Í
114	83	c	165	EE	Ó
115	84	d	166	FE	Ú
116	85	e	167	A3	t
117	87	g	168	A4	u
118	88	h	169	A5	v
119	89	i	170	86	f
120	91	j	171	93	l
121	92	k	172		unused
122		null	173		unused
123	94	m	174		unused
124	95	n	175	9D	ç
125	96	o	176		reserved
126		unused	177		reserved
127	97	p	178		reserved
128	42	â	179		reserved
129	52	ê	180		reserved
130	56	î	181		reserved
131	CB	ô	182		reserved
132	DB	û	183		reserved
133	45	á	184		reserved
134		é	185		reserved
135	59	ß	186		reserved
136	55	í	187		reserved
137	CE	ó	188		reserved
138	DE	ú	189		reserved
139	49	ñ	190		reserved
140	64	Á	191		reserved
141	74	È	192		reserved
142	78	Í	193		reserved
143	98	q́	194		reserved
144	ED	ó	195		reserved
145	FD	Ù	196		reserved
146	66	Ã	197		reserved
147	´EF	Õ	198		reserved
148		Y	199		reserved
149		A	200		reserved
150		E	201		reserved

For 3604 Model 7 228E-Character Set (continued)

Position	*Hex (Char)*	*Character*
202		reserved
203		reserved
204		reserved
205		reserved
206		reserved
207		reserved
208	6A	¦
209	79	'
210		v
211	A6	w
212	A7	x
213	A8	y
214	A9	z
215		$\overline{;}$
216		$\overline{*}$
217	B5	§
218		~
219	BE	/
220	9F	¤
221	70	ϕ
222	9E	₧
223	9C	α
224	47	å
225	90	°
226	BD	¨
227	69	Ñ
228		ç

For 3604 Model 7 228K-Character Set

Position	*Hex (Char)*	*Character*
0		¢
1	4F	|
2	5F	¬
3	4A	£
4		┌
5		•
6		ヨ
7		ヨ
8		¨
9		ヲ
10		ç
11		Pt
12		Å
13		unused
14		Ø
15	5B	¥
16	A1	—
17		{
18		}
19	41	.
20	42	┌
21	43	┘
22	44	,
23	45	•
24	BE	≤
25	BF	o
26	81	ア
27	82	イ
28	83	ウ
29	84	エ
30	85	オ
31	86	カ
32	40	space
33	5A	!
34	7F	"
35	7B	#
36	E0	$
37	6C	%
38	50	&
39	7D	'
40	4D	(
41	5D	)
42	5C	*
43	4E	+
44	6B	,
45	60	—
46	4B	.
47	61	/
48	F0	0
49	F1	1
50	F2	2
51	F3	3
52	F4	4
53	F5	5
54	F6	6
55	F7	7
56	F8	8
57	F9	9
58	7A	:
59	5E	;
60	4C	<
61	7E	=
62	6E	>
63	6F	?
64	7C	@
65	C1	A
66	C2	B

For 3604 Model 7 228K-Character Set (continued)

Position	*Hex (Char)*	*Character*	*Position*	*Hex (Char)*	*Character*
67	C3	C	116		e
68	C4	D	117		g
69	C5	E	118		h
70	C6	F	119		i
71	C7	G	120		j
72	C8	H	121		k
73	C9	I	122		null
74	D1	J	123		m
75	D2	K	124		n
76	D3	L	125		o
77	D4	M	126		unused
78	D5	N	127		p
79	D6	O	128	97	ﾆ
80	D7	P	129	98	ﾇ
81	D8	Q	130	99	ﾈ
82	D9	R	131	9A	ﾉ
83	E2	S	132	9D	ﾊ
84	E3	T	133	9E	ﾋ
85	E4	U	134	9F	ﾌ
86	E5	V	135		ß
87	E6	W	136	A2	ﾍ
88	E7	X	137	A3	ﾎ
89	E8	Y	138	A4	ﾏ
90	E9	Z	139	A5	ﾐ
91		[	140	A6	ﾑ
92		\	141	A7	ﾒ
93		]	142	A8	ﾓ
94		^	143		q
95	6D	_	144	A9	ﾔ
96	87	ｷ	145	AA	ﾕ
97	88	ｸ	146	AC	ﾖ
98	89	ｹ	147	AD	ﾗ
99	8A	ｺ	148	AE	ﾘ
100	8C	ｻ	149	AF	ﾙ
101	8D	ｼ	150	BA	ﾚ
102	8E	ｽ	151		r
103	8F	ｾ	152	BB	ﾛ
104	90	ｿ	153	BC	ﾜ
105	91	ﾀ	154	46	ｦ
106	92	ﾁ	155	BD	ﾝ
107	93	ﾂ	156	47	ｧ
108	94	ﾃ	157	48	ｨ
109	95	ﾄ	158	49	ｩ
110	96	ﾅ	159		s
111		unused	160	51	ｪ
112		a	161	52	ｫ
113		b	162	53	ｬ
114		c	163	54	ｭ
115		d	164	55	ｮ

For 3604 Model 7 228K-Character Set (continued)

Position	*Hex (Char)*	*Character*
165	56	"7
166	58	—
167		t
168		u
169		v
170		f
171		l
172		unused
173		unused
174		unused
175		ç
176		reserved
177		reserved
178		reserved
179		reserved
180		
181		
182		
183		
184		
185		
186		
187		
188		
189		
190		
191		
192		
193		
194		
195		
196		
197		reserved
198		reserved
199		
200		
201		
202		
203		
204		
205		
206		
207		reserved
208		' :
209		`
210		v
211		w
212		x
213		y
214		z
215		-;
216		*
217		§
218		~
219		/
220		¤
221		ø
222		Æ
223		æ
224		å
225		°
226		..
227		"
228		ç

Chapter 9. IBM 3278 Display Station Model 2

The IBM 3278 Display Station Model 2 (Figure 9-1) is made up of a 75-key or 87-key keyboard and a display component that can display 1920 characters on a cathode-ray tube (CRT).

The display station is attached to the controller via the Device Cluster Adapter (DCA) feature. The application program controls the 3278 as it does the IBM 3604 keyboard display.

Functions and Features

The following list summarizes 3278-2 features and functions, some of which differ from a corresponding 3604 (for details see the remainder of this chapter and the 3604 Keyboard Display chapter):

- You may specify which keys will control the 'cursor blink', 'alternate cursor', and 'keyboard clicker' functions (the INTRTBL configuration macro).
- By specifying the APL feature on the DCA3278 or DCA3279 device configuration macro, you can use the extended data stream support described in Chapter 4, "Processing 3270 Data Streams" to support APL characters.
- The magnetic slot reader special feature allows your application to read the American Banking Association (ABA) ten-character codes.
- The audible alarm special feature sounds a single short tone when indicators X, 1, 2, or 3 are turned on.
- 3278 typematic and downshift function keys are fixed. You may not assign these functions to keys via the TRTBHDR macro.
- There is no significant difference between EOM and EOF keys for 3278s; neither ends a physical read.

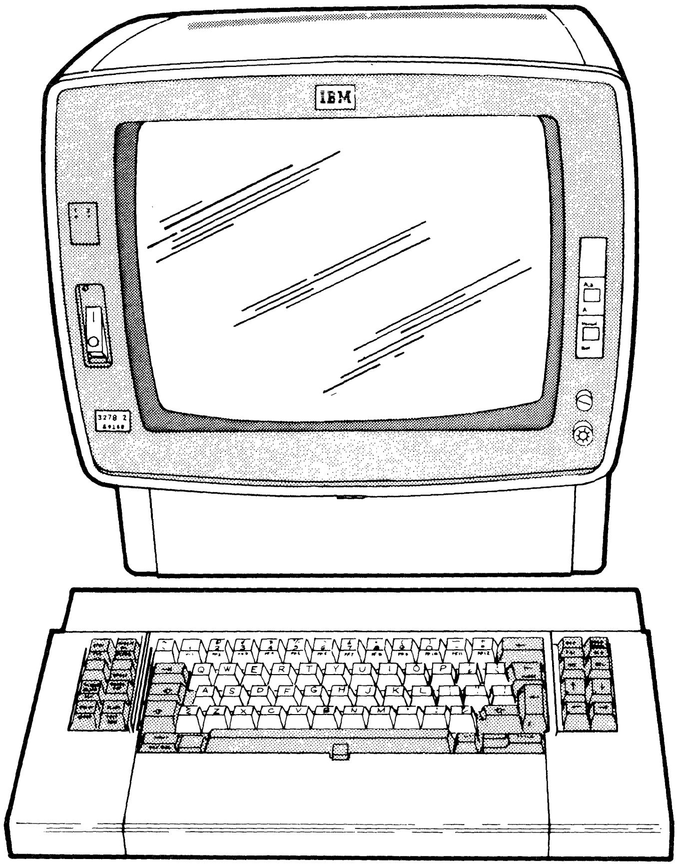


Figure 9-1. IBM 3278 Display Station, Model 2

3278-2 Keyboard Component

Four keyboards are available for 3278-2: a 75-key typewriter, 75-key data entry, a 75-key data entry keypunch layout, and an 87-key typewriter keyboard.

Cursor blink, alternate cursor, and keyboard clicker functions may be associated with keys via the INTRTBL configuration macro. Each of these functions is either active or inactive; pressing a corresponding key causes the function to be activated (if inactive) or inactivated (if active). When the controller or 3278 is reset, the cursor blink is inactive, the alternate cursor is inactive, and keyboard clicker is active.

Cursor Blink

Cursor Blink is activated by pressing a cursor blink function key. This causes the cursor to blink approximately 3 times per second until the cursor blink key is again pressed.

Alternate Cursor

Alternate cursor is activated by pressing an alternate cursor function key. This causes the cursor to change from an underline to a rectangular cursor (reverse video). When the configured Alternate Cursor key is again pressed, the cursor reverts to the underline.

Keyboard Clicker

Keyboard clicking is deactivated by pressing a keyboard clicker function key. This disables the clicker (assuming that it was enabled). When the keyboard clicker key is again pressed, the clicker is enabled.

The clicker state also changes when the check indicator changes. If clicking is active and an input-inhibited condition occurs (for example, purge mode is active, or position of the security keylock is changed), then clicking is stopped. If clicking had been inactive, the input inhibited condition would activate clicking.

Display Component

The display component is cathode-ray tube (CRT) that can display 1920 characters in a 24-line by 80 column format. The display also has operator controls and indicators and a location for the security keylock on the display housing. The display screen has a 25th line that provides the operator with symbols to assist in operating procedures.

Dual Intensity

Dual intensity is supported through the set attribute (X'28C1aa') display control character (where 'aa' is the desired attribute). An attribute of X'40' or X'C4' indicates normal intensity; an attribute of X'C8' indicates high intensity. The attribute, when written to the display by the set attribute control character:

- Occupies a display position.
- Displays as a blank.
- Causes characters in display positions which follow the attribute's display position to be displayed with the specified intensity. Attribute control wrap occurs from the lower right-hand corner to the upper left-hand corner of the display.

For example, if the following data stream were written to the display:

$$X'\underbrace{0C}_{1}\underbrace{F1}_{2}\underbrace{28C1C8}_{3}\underbrace{F2F3}_{4}\underbrace{28C140}_{5}\underbrace{F4}_{6}'$$

the following would be displayed (assuming an appropriate output translation table) at the upper left-hand corner of the display screen.

Row \ Col	1	2	3	4	5	6
1	1		**2**	**3**		4
2						

Notes:

1. Causes the screen to be cleared (ensuring that attributes are not inadvertently left on the display).

2. Causes the character 1 to be displayed in column 1 of row 1 with normal intensity.

3. Causes the high intensity attribute byte to be sent to the display in column 2 of row 1. Note that the attribute is displayed as a blank.

4. Causes the characters 2 and 3 to be displayed in high intensity.

5. Causes the normal intensity attribute byte to be sent to the display in column 5 of row 1. Note that the attribute is displayed as a blank. All display positions following this attribute will be generated at normal intensity; this includes the character 1 in column 1 of row 1 (because of the attribute control wrapping).

6. Causes the character 4 to be displayed in column 6 of row 1 in normal intensity.

Special Features and Options

The 3278-2 offers the following special features and options:

Magnetic Slot Reader

The Magnetic Slot Reader special feature (9441) and its accompanying Magnetic Reader Control feature (4999 or RPQ ZC7299) allow the 3278-2 to read American Banking Association (ABA) ten-character codes with an *F* end of stripe character. The method for reading the magnetic stripe is compatible with the method used for reading the 3604/4704 magnetic stripe.

The points to consider when programming the 3278-2 magnetic slot reader vary, depending on whether the display operates through the 4700 Data Stream Mapping (DATSM) facility, or not.

Existing non-DATSM application programs written for the 3604 or 4704 magnetic stripe reader (MSR) run without modification with the 3278 magnetic slot reader, which is treated as part of the keyboard. However, to ensure compatible operation, use the MSTRTBL macro to define the character "F" as the EOM character having a corresponding output value of X'7C'. The program then interprets X'7C' as the end of stripe *and* EOM character, just as it does with the character "C" for the 3604 and 4704.

Your program reads data from the magnetic slot reader by issuing LREAD to read from the keyboard. Data from both the MSR and the keyboard can be in the same or in separate messages. When LREAD completes successfully, SMSIML contains the total message length.

When used with DATSM, the 4700 allows the character "B" (start-of-stripe) character to be an EOM/EID indicator. DATSM recognizes this EID, generates a protected, nondisplay field at the current cursor position, and then notifies the program to issue LREAD NOTRACK to read the magnetic stripe data. The DATSM image buffer is updated with this stripe data, and any fields affected by this data are marked as "modified".

If the current cursor position is in either an attribute location or a protected field when DATSM recognizes the EID, the magnetic stripe data field does not update the DATSM image buffer.

Note: The 3278 magnetic slot reader cannot read a stripe containing fewer than six data characters. An attempt to read such a stripe lights the red MSR indicator, and the data is lost.

Security Keylock

The 3278 can have a security keylock, on the right side of the display unit housing, to improve data processing security by allowing operation by authorized persons only.

When the keylock is turned to off (secured) and no data transmission is in process, the display screen is blank (except for the indicator row and the cursor), the keyboard clicker state is changed, and keyboard data transmission is inhibited. However, the terminal remains available for messages from the application program. When the keylock is turned on (unsecured), the screen returns to its original display provided that an intervening message from the application program has not altered the original display message, keyboard clicking state is changed, and operations return to normal.

Turning the keylock off when a message is being sent to the terminal also erases the screen except for the indicator row and the cursor. The message to the terminal continues until normal termination but additional operator input is blocked. When the keylock is turned on, the entire message is displayed provided that another has not altered it.

Audible Alarm

The 3278 can have an audible alarm as a special feature. The alarm is sounded when any indicator 1, 2, 3, or Check is turned on. The alarm is a single short tone. The tone can be used to inform the operator that an indicator has just been displayed (NOTE: The Check indicator when set by the operator turning the security key off does not cause the alarm to sound). The loudness of the alarm can be adjusted by the operator.

Controls and Indicators

The 3278-2 display has operator controls, indicator lights, and screen indicators. Operator controls and indicator lights are on the display unit housing and the screen indicators are in row 25 of the display screen.

Operator Controls

Normal/Test

This two-position rocker switch when placed in the "Test" position logically disconnects the terminal from the system and initiates the test mode sequence.

Audible Alarm Volume Control (Special Feature)

This adjustable control enables the operator to set the loudness of the audible alarm feature.

Intensity

This adjustable control enables the operator to set the overall brightness of data displayed on the screen.

Intensity Override

This switch, an integral part of the intensity control, when turned clockwise enables the operator to override control of the brightness.

Contrast

This adjustable control enables the operator to establish the difference in brightness between High and Normal intensity data fields.

Security Keylock (Special Feature)

This key-operated switch provides additional security to allow only authorized persons to operate the terminal.

Power On/Off

This two position rocker switch, with graphic symbols, turns the display station power on and off.

A,a/A

This two position rocker switch, when in the "A,a" position, enables characters to be displayed as they are written to the display (upper or lowercase). When in the "A" position, all characters are displayed in their uppercase form regardless of the form in which they were written to the display.

Lamp Indicators

Light 1

This light indicates operation of the "sweep" circuits that provide character display on the screen.

Light 2

This light indicates the presence of voltage required to illuminate the display screen.

Light 3

This light indicates the presence of voltages required to operate the display station logic.

Indicators for the Magnetic Slot Reader (Special Feature)

This feature has three indicators—green, red, and yellow. Only one of these indicators is on at a time. Their meanings are:

1. Green–MSR power is on, the test switch is set to "normal", and/or the controller read the MSR data successfully.

2. Yellow–The slot reader read data successfully and is sending that data to the controller.

3. Red–Either the controller or slot reader did not read data successfully, the data was invalid, or a required program module has not been installed in the controller configuration.

Screen Indicators

Nonprogrammable and programmable indicators appear at the bottom of the display screen. These indicators do not appear when the display is in Test mode.

4700

This indicator shows that this display is attached to a 4700 controller. The indicator appears when communication with the controller is established. This indicator may fail to display for any of the following reasons:

- Display station not powered on.

- Display station not connected to a DCA port that was configured for a 3278.

- The operator suppressed loading of the DCA or 3278 optional modules during startup.

- A hardware error exists in the 3278, the coaxial cable, the DCA, or the controller.

X

This indicator shows that the keyboard component is in purge mode. When X is turned on, the functional state of the keyboard clicker is reversed.

This indicator may show that the keyboard component was placed into purge mode by the operator turning the security keylock (feature) to the secured position.

1, 2, 3

Define programmable indicators 1, 2, and 3 according to your specific requirements.

↑1, ↓2, ↑3, ↓4

You can select these optional shift case indicators using the IND=Y operand of the INTRTBL macro.

Configuring the 3278

The 3278 display must be defined with configuration macro instructions to describe the unit's physical characteristics, and its relationship to the controller. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the desired hardware and programming configuration.

For coding information and descriptions of the DCA3278 macro, refer to Volume 6. Refer to "Programming DATSM for APL and Color Displays" on page 4-45 in Chapter 4, "Processing 3270 Data Streams" for other points to consider when configuring a 3278 display.

The following is a sample CPGEN including APL and color (3278/3279) terminals:

```
DSP3278   DCA3278   MODEL=8,OUTRTBL=OUTPSS,TRTBHDR=HDRPSS,FEATURE=APL   00000010
DSP3279   DCA3279   MODEL=8B,OUTRTBL=OUTPSS,TRTBHDR=HDRPSS,FEATURE=APL  00000020
HDRPSS    PSSLST    HDR,APLHDR                                           00000030
OUTPSS    PSSLST    OUT,APLOUT                                           00000040
HDR       TRTBHDR   CANCEL=31,INTRTBL=(IN3278C1, IN3278C2, IN327983),   *00000050
                    SHFT=(30,3C)                                         00000060
IN3278C1  INTRTBL   (30,,CAS2),(20,,CAS2),(3C,,CAS2),(56,,CAS3),        *00000070
                    (1D,X'12',REVERSE,EOM,X'FF',EID),   REVERSE   41F2  *00000080
                    (1E,X'31',PS1,EOM,X'FF',EID),       PSA       43F1  *00000090
                    (50,X'22',RED,EOM,X'FF',EID,)       RED       42F2  *00000100
                    (2D,X'11',BLINK,EOM,X'FF',EID),     BLINK     41F1  *00000110
                    (2E,X'32',PS2,EOM,X'FF',EID),       PSB       43F2  *00000120
                    (60,X'24',GREEN,EOM,X'FF',EID),     GREEN     42F4  *00000130
                    (3D,X'13',UNDRSCR,EOM,X'FF,EID),    UNDERSCR  41F3  *00000140
                    (3E,X'33',PS3,EOM,X'FF',EID),       PSC       43F3  *00000150
                    (42,X'21',BLUE,EOM,X'FF',EID),      BLUE      42F1  *00000160
                    (52,X'27',WHITE,EOM,X'FF',EID),     WHITE     42F7  *00000170
                    (0E,,BKSP),(15,,ADVAN),KYBD=3278A,CASE=1             00000180
IN3278C2  INTRTBL   (20,,CAS1),(56,,CAS3),                              *00000190
                    (0E,,BKSP),(15,,ADVAN),KYBD=3278A,CASE=2             00000200
IN3278C3  INTRTBL   (1E,X'34',PS4,EOM,X'FF',EID),       PSD       43F4  *00000210
                    (50,X'23',PINK,EOM,X'FF',EID),      MAGENTA   42F3  *00000220
                    (2E,X'35',PS5,EOM,X'FF',EID),       PSE       43F5  *00000230
                    (60,X'26',YELLOW,EOM,X'FF',EID),    YELLOW    42F6  *00000240
                    (3E,X'36',PS6,EOM,X'FF',EID),       PSF       43F6  *00000250
                    (42,X'25',TURQS,EOM,X'FF',EID),     TURQUOISE 42F5  *00000260
                    (0F,X'10',DEFHI,EOM,X'FF',EID), CANCEL HILITE 41F0  *00000270
                    (51,X'30',DEFPS,EOM,X'FF',EID), CANCEL PS     43F0  *00000280
                    (52,X'20',DEFCOL,EOM,X'FF',EID), CANCEL COLOR 42F0  *00000290
                    KYBD=3278A,CASE=3                                    00000300
APLHDR    TRTBHDR   CANCEL=31,INTRTBL=(IN3278A1,IN3278A2,IN3278A3),     *00000310
                    SHFT=(30,3C)                                         00000320
IN3278A1  INTRTBL   (30,,CAS2),(20,,CAS2),(3C,,CAS2),(56,,CAS3),        *00000330
                    (1D,X'12',REVERSE,EOM,X'FF',EID),   REVERSE   41F2  *00000340
                    (1E,X'31',PS1,EOM,X'FF',EID),       PSA       43F1  *00000350
                    (50,X'22',RED,EOM,X'FF',EID),       RED       42F2  *00000360
                    (2D,X'11',BLINK,EOM,X'FF',EID),     BLINK     41F1  *00000370
                    (2E,X'32',PS2,EOM,X'FF',EID),       PSB       43F2  *00000380
                    (60,X'24', GREEN, EOM,X'FF',EID),   GREEN     42F4  *00000390
                    (3D,X'13',UNDRSCR,EOM,X'FF',EID),   UNDRSCR   41F3  *00000400
                    (3E,X'33',PS3,EOM,X'FF',EID),       PSC       43F3  *00000410
                    (42,X'21',BLUE,EOM,X'FF',EID)       BLUE      42F1  *00000420
                    (52,X'27',WHITE,EOM,X'FF',EID),     WHITE     42F7  *00000430
                    (0E,,BKSP),(15,,ADVAN),KYBD=APL,CASE=1               00000440
```

Figure 9-2 (Part 1 of 2). Sample 3278/3279 Configuration

```
IN3278A2  INTRTBL   (20,,CAS1),(56,,CAS3),                                   *00000450
                    (0E,,BKSP),(15,,ADVAN),KYBD=APL,CASE=2                    00000460
IN3278A3  INTRTBL   (1E,X'34',PS4,EOM,X'FF',EID),       PSD           43F4   *00000470
                    (50,X'23',PINK,EOM,X'FF',EID),      MAGENTA       42F3   *00000480
                    (2E,X'35',PS5,EOM,X'FF',EID),       PSE           43F5   *00000490
                    (60,X'26',YELLOW,EOM,X'FF',EID),    YELLOW        42F6   *00000500
                    (3E,X'36',PS6,EOM,X'FF',EID),       PSF           43F6   *00000510
                    (42,X'25',TURQS,EOM,X'FF',EID),     TURQUOISE     42F5   *00000520
                    (0F,X'10',DEFHI,EOM,X'FF',EID), CANCEL HILITE     41F0   *00000530
                    (51,X'30',DEFPS,EOM,X'FF',EID), CANCEL PS         43F0   *00000540
                    (52,X'20',DEFCOL,EOM,X'FF',EID), CANCEL COLOR     42F0   *00000550
                    KYBD=APL,CASE=3                                           00000560
OUT       OUTRTBL   DEVICE=3278,CHARSET=192E
APLOUT    OUTRTBL   DEVICE=3278,CHARSET=APL
```

Figure 9-2 (Part 2 of 2). Sample 3278/3279 Configuration

3278/3279 Substitution

You can substitute the 4700-compatible 3278s and 3279s, subject to the following constraints:

- If a port is defined as a 3278, but a 3279 is attached, the 3279 operates as though it were a 3278. The convergence feature is not recognized, the power-on convergence routine is not invoked, and the combination of the ALT key with the Clicker/TEST key acts as specified in the user's translation tables. If no specification is given for this combination, the keystrokes are ignored.

- If a port is defined as a 3279, but a 3278 is attached, the 3278 operates as though a 3278 had been defined. The combination of the ALT key with the Clicker/TEST key acts as specified in the user's translation tables. If no specification is given for this combination, the keystrokes are ignored.

- If a port is defined as a 4704-2/3, but a 3278 or 3279 is attached, you can read and write only the characters 0 – 9, A – Z, and a – z..

Terminal Control Characters

Control characters are embedded in the data stream and indicate operations to be performed at a terminal. The following are the control characters for the 3278:

Function:	Control Character:	Explanation:
Line Feed	X'25'	The cursor moves down one line; the horizontal position is unchanged.
Return	X'OD'	The cursor moves to position 1 on the current display line.
New Line	X'15'	The current line is erased from the cursor to the end of the line. The cursor is set to position 1 of the next lower line.
New Page	X'OC'	The screen is completely erased; the cursor moves to position 1 of line 1.
Set Attribute	X'28'	This 3-byte control string sends one byte to the display. The second byte should be X'C1'. The third byte may be any valid 3270 attribute; only the following values should be used: X'40' Normal intensity X'C4' Normal intensity; selector pen not supported X'C8' Intensified; selector pen not supported X'4C' Nondisplay For 3278 Model 2, the byte sent to the display is the attribute byte (which is displayed as a space) and assigns the given characteristics to all characters on the display between this attribute byte and the next. The next attribute may be found via display wrap from the last character of the last line (excluding the indicator row) to the first character of the first line, and continuing the search for the attribute.

Function:	Control Character:	Explanation:
Position	X'34'	This control character is followed by a flag byte and a 1-byte positioning value. Flag byte: Bits 0-3 are ignored. Bit 4: 0 = Position is an absolute value. 1 = Position is relative to the present position. Bit 5: 0 = Position change is horizontal. 1 = Position change is vertical. Bits 6-7: 00 = Move cursor to new position without erasing. 01 = Leave cursor at the current position, but erase from the current position the number of bytes specified by the position. 10 = Move cursor to new position and erase from the old position up to, but not including, the new position. 11 = Reserved. The positioning byte indicates a line or column number. It contains an unsigned binary value ranging from 0 to 255 to be used under direction of the flag byte; a 0 results in no operation. The value specified is divided by a modulus number to determine the resulting position. The modulus division produces a change of 1 to x lines for vertical positioning, and a change of 1 to y columns for horizontal positioning, where x is the maximum number of lines on a screen, and y is the maximum number of columns in a line.
Select	X'04'	The byte following this control character is skipped (no operation takes place). Automatic new line action is suppressed.

Function:	Control Character:	Explanation:
Transparent Write	X'35'	A transparent write (a write without translation) begins. The format of this control sequence is: Byte 0: The control character, X'35'. Byte 1: The number of display positions to be displayed. Byte 2-Byte n: The positions to be displayed. The valid positions are X'00'-X'AF'

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or they may be restarted, or "wrapped back," to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 3278:

Keyboard

Counter	Explanation
1	DCA error
2	Port/component/terminal error
3	Reserved
4	Translation checks
5	Segment overruns
6-15	Reserved

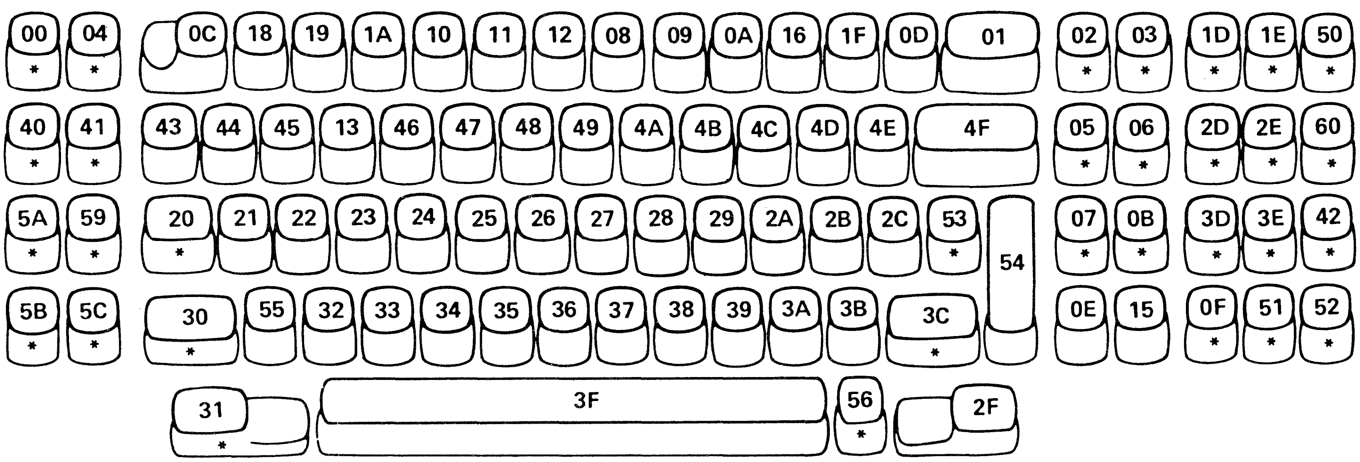
Display

Counter	Explanation
1	DCA error
2	Port/component/terminal error
6-12	Reserved

Function Keys, Keyboards, and Translation Tables

The next few pages contain the 3278 keyboard and the standard input translation tables for it.

Keyboard for 3278A, and 3278K



*Non-typematic key

Notes:

1. X'53' key is available only on Katakana keyboard.
2. Key that produces downshift when released has scan code of X'56'.
3. Keys that may be selected in TRTBHDR as downshift keys have scan codes X:20', X'30', X'3C', and X'53'.

3278A Keyboard

Scan Code	*Case 1*	*Case 2*	*Scan Code*	*Case 1*	*Case 2*
00	NOOP	NOOP	30	NOOP	NOOP
01	BKSP	BKSP	31	NOOP	NOOP
02	NOOP	NOOP	32	C'z'	C'Z'
03	NOOP	NOOP	33	C'x'	C'X'
04	NOOP	NOOP	34	C'c'	C'C'
05	NOOP	NOOP	35	C'v'	C'V'
06	NOOP	NOOP	36	C'b'	C'B'
07	NOOP	NOOP	37	C'n'	C'N'
08	C'7'	X'50'	38	C'm'	C'M'
09	C'8'	X'5C'	39	X'6B'	X'6B'
0A	X'9'	X'4D'	3A	X'4B'	X'4B'
0B	NOOP	NOOP	3B	X'61'	X'6F'
0C	X'79'	NOOP	3C	NOOP	NOOP
0D	X'7E'	X'4E'	3D	NOOP	NOOP
0E	NOOP	NOOP	3E	NOOP	NOOP
0F	NOOP	NOOP	3F	X'40'	X'40'
10	C'4'	X'5B'	40	NOOP	NOOP
11	C'5'	X'6C'	41	NOOP	NOOP
12	C'6'	X'5F'	42	NOOP	NOOP
13	C'e'	C'E'	43	NOOP	NOOP
15	NOOP	NOOP	44	C'q'	C'Q'
16	C'0'	X'5D'	45	C'w'	C'W'
18	C'1'	X'4F'	46	C'r'	C'R'
19	C'2'	X'7C'	47	C't'	C'T'
1A	C'3'	X'7B'	48	C'y'	C'Y'
1D	NOOP	NOOP	49	C'u'	C'U'
1E	NOOP	NOOP	4A	C'i'	C'I'
1F	X'60'	X'6D'	4B	C'o'	C'O'
20	NOOP	NOOP	4C	C'p'	C'P'
21	C'a'	C'A'	4D	X'4A'	X'5A'
22	C's'	C'S'	4E	X'E0'	X'6A'
23	C'd'	C'D'	4F	NOOP	NOOP
24	C'f'	C'F'	50	NOOP	NOOP
25	C'g'	C'G'	51	NOOP	NOOP
26	C'h'	C'H'	52	NOOP	NOOP
27	C'j'	C'J'	54	NOOP	NOOP
28	C'k'	C'K'	55	X'4C'	X'6E'
29	C'l'	C'L'	56	NOOP	NOOP
2A	X'5E'	X'7A'	59	NOOP	NOOP
2B	X'7D'	X'7F'	5A	BLNKCUR	BLNKCUR
2C	NOOP	NOOP	5B	ALTCUR	ALTCUR
2D	NOOP	NOOP	5C	CLICKER	CLICKER
2E	NOOP	NOOP	60	NOOP	NOOP
2F	EOM	EOM			

3278A Data Entry Keyboard

Scan Code	*Case 1*	*Case 2*
00	NOOP	NOOP
01	BKSP	BKSP
02	NOOP	NOOP
03	NOOP	NOOP
04	NOOP	NOOP
05	NOOP	NOOP
06	NOOP	NOOP
07	NOOP	NOOP
08	NOOP	NOOP
09	X'60'	X'60'
0A	X'61'	C'0'
0B	NOOP	NOOP
0C	NOOP	NOOP
0D	NOOP	NOOP
0E	NOOP	NOOP
10	X'5C'	X'5B'
11	X'4C'	X'4B'
12	NOOP	NOOP
13	C'E'	X'5D'
15	NOOP	NOOP
16	NOOP	NOOP
18	NOOP	NOOP
19	X'7C'	X'7B'
1A	X'6C'	X'6B'
1F	NOOP	NOOP
20	NOOP	NOOP
21	C'A'	NOOP
22	C'S'	X'6E'
23	C'D'	X'7A'
24	C'F'	X'5E'
25	C'G'	X'5F'
26	C'H'	X'7D'
27	C'J'	C'4'
28	C'K'	C'5'
29	C'L'	C'6'
2A	NOOP	NOOP
2B	NOOP	NOOP
2C	NOOP	NOOP
2F	EOM	EOM
30	NOOP	NOOP
31	NOOP	NOOP
32	C'Z'	NOOP
33	C'X'	X'6F'
34	C'C'	X'7F'
35	C'V'	X'7E'
36	C'B'	X'5A'
37	C'N'	X'4D'
38	C'M'	C'7'
39	X'6B'	C'8'
3A	X'4B'	C'9'
3B	NOOP	NOOP
3C	NOOP	NOOP
3F	NOOP	NOOP
40	NOOP	NOOP
41	NOOP	NOOP
43	NOOP	NOOP
44	C'Q'	X'4E'
45	C'W'	X'6D'
46	C'R'	X'4A'
47	C'T'	X'E0'
48	C'Y'	X'4F'
49	C'U'	C'1'
4A	C'I'	C'2'
4B	C'O'	C'3'
4C	C'P'	X'50'
4D	NOOP	NOOP
4E	NOOP	NOOP
4F	NOOP	NOOP
54	NOOP	NOOP
55	NOOP	NOOP
56	NOOP	NOOP
59	NOOP	NOOP
5A	BLNKCUR	BLNKCUR
5B	ALTCUR	ALTCUR
5C	CLICKER	CLICKER

3278A Data Entry Keypunch Keyboard

Scan Code	*Case 1*	*Case 2*	*Scan Code*	*Case 1*	*Case 2*
00	NOOP	NOOP	2F	NOOP	NOOP
01	NOOP	NOOP	30	NOOP	NOOP
02	NOOP	NOOP	31	NOOP	NOOP
03	NOOP	NOOP	32	C'Z'	NOOP
04	NOOP	NOOP	33	C'X'	X'6F'
05	NOOP	NOOP	34	C'C'	X'7F'
06	NOOP	NOOP	35	C'V'	X'7E'
07	NOOP	NOOP	36	C'B'	X'5A'
08	NOOP	NOOP	37	C'N'	X'4D'
09	X'60'	X'60'	38	C'M'	C'7'
0A	X'61'	C'0'	39	X'6B'	C'8'
0B	NOOP	NOOP	3A	X'4B'	C'9'
0C	NOOP	NOOP	3B	BKSP	BKSP
0D	NOOP	NOOP	3C	NOOP	NOOP
0E	NOOP	NOOP	3F	NOOP	NOOP
10	X'5C'	X'5B'	40	NOOP	NOOP
11	X'4C'	X'4B'	41	NOOP	NOOP
12	NOOP	NOOP	43	NOOP	NOOP
13	C,E'	X'5D'	44	C'Q'	X'4E'
15	NOOP	NOOP	45	C'W'	X'6D'
16	NOOP	NOOP	46	C'R'	X'4A'
18	NOOP	NOOP	47	C'T'	X'E0'
19	X'7C'	X'7B'	48	C'Y'	X'4F'
1A	X'6C'	X'6B'	49	C'U'	C'1'
1F	NOOP	NOOP	4A	C'I'	C'2'
20	NOOP	NOOP	4B	C'O'	C'3'
21	C'A'	NOOP	4C	C'P'	X'50'
22	C'S'	X'6E'	4D	EOM	EOM
23	C'D'	X'7A'	4E	NOOP	NOOP
24	C'F'	X'5E'	4F	NOOP	NOOP
25	C'G'	X'5F'	54	NOOP	NOOP
26	C'H'	X'7D'	55	NOOP	NOOP
27	C'J'	C'4'	56	NOOP	NOOP
28	C'K'	C'5'	59	NOOP	NOOP
29	C'L'	C'6'	5A	BLNKCUR	BLNKCUR
2A	NOOP	NOOP	5B	ALTCUR	ALTCUR
2B	NOOP	NOOP	5C	CLICKER	CLICKER
2C	NOOP	NOOP			

3278K Keyboard

Scan Code	*Case 1*	*Case 2*	*Case 3*	*Case 4*
00	NOOP	NOOP	NOOP	NOOP
01	BKSP	BKSP	BKSP	BKSP
02	NOOP	NOOP	NOOP	NOOP
03	NOOP	NOOP	NOOP	NOOP
04	NOOP	NOOP	NOOP	NOOP
05	NOOP	NOOP	NOOP	NOOP
06	NOOP	NOOP	NOOP	NOOP
07	NOOP	NOOP	NOOP	NOOP
08	C'8'	X'4D'	X'AA'	X'54'
09	C'9'	X'5D'	X'AC'	X'55'
0A	C'0'	X'4A'	X'BC'	X'46'
0B	NOOP	NOOP	NOOP	NOOP
0C	C'1'	X'5A'	X'98'	NOOP
0D	X'5B'	X'4F'	X'58'	NOOP
0E	NOOP	NOOP	NOOP	NOOP
0F	NOOP	NOOP	NOOP	NOOP
10	C'5'	X'6C'	X'84'	X'51'
11	C'6'	X'50'	X'85'	X'52'
12	C'7'	X'7D'	X'A9'	X'53'
13	C'E'	NOOP	X'82'	X'48'
15	NOOP	NOOP	NOOP	NOOP
16	X'60'	X'7E'	X'A3'	NOOP
18	C'2'	X'7F'	X'9F'	NOOP
19	C'3'	X'7B'	X'81'	X'47'
1A	C'4'	X'E0'	X'83'	X'49'
1D	NOOP	NOOP	NOOP	NOOP
1E	NOOP	NOOP	NOOP	NOOP
1F	X'5F'	X'A1'	X'A2'	NOOP
20	NOOP	NOOP	NOOP	NOOP
21	C'A'	NOOP	X'92'	NOOP
22	C'S'	NOOP	X'95'	NOOP
23	C'D'	NOOP	X'8D'	NOOP
24	C'F'	NOOP	X'9D'	NOOP
25	C'G'	NOOP	X'87'	NOOP
26	C'H'	NOOP	X'88'	NOOP
27	C'J'	NOOP	X'A4'	NOOP
28	C'K'	NOOP	X'9A'	NOOP
29	C'L'	NOOP	X'AE'	NOOP
2A	X'5E'	X'4E'	X'BA'	NOOP
2B	X'7A'	X'5C'	X'89'	NOOP
2C	NOOP	NOOP	X'A6'	X'4B'
2D	NOOP	NOOP	NOOP	NOOP
2E	NOOP	NOOP	NOOP	NOOP
2F	EOM	EOM	EOM	EOM

Scan Code	*Case 1*	*Case 2*	*Case 3*	*Case 4*
30	NOOP	NOOP	NOOP	NOOP
31	NOOP	NOOP	NOOP	NOOP
32	C'X'	NOOP	X'8C'	NOOP
33	C'C'	NOOP	X'90'	NOOP
34	C'V"	NOOP	X'9E'	NOOP
35	C'B'	NOOP	X'8A'	NOOP
36	C'N'	NOOP	X'A5'	NOOP
37	C'M'	NOOP	X'A8'	NOOP
38	X'6B'	X'4C'	X'99'	X'44'
39	X'4B'	X'6E'	X'AF'	X'41'
3A	X'61'	X'6F'	X'A7'	X'45'
3B	NOOP	X'6D'	X'BB'	NOOP
3C	NOOP	NOOP	NOOP	NOOP
3D	NOOP	NOOP	NOOP	NOOP
3E	NOOP	NOOP	NOOP	NOOP
3F	X'40'	X'40'	X'40'	X'40'
40	NOOP	NOOP	NOOP	NOOP
41	NOOP	NOOP	NOOP	NOOP
42	NOOP	NOOP	NOOP	NOOP
43	NOOP	NOOP	NOOP	NOOP
44	C'Q'	NOOP	X'91'	NOOP
45	C'W'	NOOP	X'94'	NOOP
46	C'R'	NOOP	X'8E'	NOOP
47	C'T'	NOOP	X'86'	NOOP
48	C'Y'	NOOP	X'BD'	NOOP
49	C'U'	NOOP	X'96'	NOOP
4A	C'I'	NOOP	X'97'	NOOP
4B	C'O'	NOOP	X'AD'	NOOP
4C	C'P'	NOOP	X'8F'	NOOP
4D	X'7C'	NOOP	X'BE'	NOOP
4E	NOOP	NOOP	X'BF'	X'42'
4F	NOOP	NOOP	NOOP	NOOP
50	NOOP	NOOP	NOOP	NOOP
51	NOOP	NOOP	NOOP	NOOP
52	NOOP	NOOP	NOOP	NOOP
53	NOOP	NOOP	NOOP	NOOP
54	NOOP	NOOP	NOOP	NOOP
55	C'Z'	NOOP	X'93'	X'56'
56	NOOP	NOOP	NOOP	NOOP
59	NOOP	NOOP	NOOP	NOOP
5A	BLNKCUR	BLNKCUR	BLNKCUR	BLNKCUR
5B	ALTCUR	ALTCUR	ALTCUR	ALTCUR
5C	CLICKER	CLICKER	CLICKER	CLICKER
60	NOOP	NOOP	NOOP	NOOP

For 3278, 192E-Character Set

Position	*Hex (Char)*	*Character*	*Position*	*Hex (Char)*	*Character*
0		null	48	50	&
1		unused	49	60	—
2		unused	50	4B	.
3		unused	51	6B	,
4		unused	52	7A	:
5		unused	53	4E	+
6		unused	54	5F	¬
7		unused	55	A1	‾
8	6E	>	56	90	°
9	4C	<	57		∨
10	C0	[	58	BA	∧
11	D0	]	59		~
12	5D	)	60	BD	¨
13	4D	(	61	79	`
14		}	62	BE	´
15		{	63	9D	ç
16	40	space	64		à
17	7E	=	65		è
18	7D	'	66		ì
19	7F	"	67		ò
20	61	/	68		ù
21	E0	\	69	46	ã
22	4F	¦	70	CF	õ
23	6A	¡	71	DF	ÿ
24	6F	?	72	44	à
25	5A	!	73	54	è
26	5B	$	74	51	é
72	4A	¢	75	58	ì
28	B1	£	76	CD	ò
29	B2	¥	77	DD	ù
30	B3	Pt	78	DC	ü
31	9F	¤	79	48	ç
32	F0	0	80	43	ä
33	F1	1	81	53	ë
34	F2	2	82	57	ï
35	F3	3	83	CC	ö
36	F4	4	84		ü
37	F5	5	85	42	â
38	F6	6	86	52	ê
39	F7	7	87	56	î
40	F8	8	88	CB	ô
41	F9	9	89	DB	û
42	59	ß	90	45	á
43	B5	§	91		é
44	7B	#	92	55	í
45	7C	@	93	CE	ó
46	6C	%	94	DE	ú
47	6D	—	95	49	ñ

For 3278, 192E-Character Set (continued)

Position	Hex (Char)	Character	Position	Hex (Char)	Character
96	64	À	144	98	q
97	74	È	145	99	r
98	78	Ì	146	A2	s
99	ED	Ò	147	A3	t
100	FD	Ù	148	A4	u
101	66	Ã	149	A5	v
102	EF	Õ	150	A6	w
103		Y	151	A7	x
104		A	152	A8	y
105		E	153	A9	z
106		E	154	9C	æ
107		I	155	70	ø
108		O	156	47	å
109		U	157		ç
110		Y	158		$\overline{;}$
111		C	159		$\overline{*}$
112	63	Ä	160	C1	A
113	73	Ë	161	C2	B
114	77	Ï	162	C3	C
115	EC	Ö	163	C4	D
116	FC	Ü	164	C5	E
117	62	Â	165	C6	F
118	72	Ê	166	C7	G
119	76	Î	167	C8	H
120	EB	Ô	168	C9	I
121	FB	Û	169	D1	J
122	65	Á	170	D2	K
123	71	É	171	D3	L
124	75	Í	172	D4	M
125	EE	Ó	173	D5	N
126	FE	Ú	174	D6	O
127	69	Ñ	175	D7	P
128	81	a	176	D8	Q
129	82	b	177	D9	R
130	83	c	178	E2	S
131	84	d	179	E3	T
132	85	e	180	E4	U
133	86	f	181	E5	V
134	87	g	182	E6	W
135	88	h	183	E7	X
136	89	i	184	E8	Y
137	91	j	185	E9	Z
138	92	k	186	9E	Æ
139	93	l	187	80	Ø
140	94	m	188	67	Å
141	95	n	189	68	Ç
142	96	o	190	5E	;
143	97	p	191	5C	*

For 3278, 192K-Character Set

Position	*Hex (Char)*	*Character*	*Position*	*Hex (Char)*	*Character*
0		null	48	50	&
1		unused	49	60	–
2		unused	50	4B	.
3		unused	51	6B	,
4		unused	52	7A	:
5		unused	53	4E	+
6		unused	54	5F	¬
7		unused	55	A1	°
8	6E	>	56	90	
9	4C	<	57		∨
10		[	58		∧
11		]	59		~
12	5D	)	60		¨
13	4D	(	61		`
14		}	62		´
15		{	63		ç
16	40	space	64	81	ア
17	7E	=	65	82	イ
18	7D	'	66	83	ウ
19	7F	"	67	84	エ
20	61	/	68	85	オ
21		\	69	86	カ
22	4F	|	70	87	キ
23		¦	71	88	ク
24	6F	?	72	89	ケ
25	5A	!	73	8A	コ
26	E0	$	74	8C	サ
72		¢	75	8D	シ
28	4A	£	76	8E	ス
29	5B	¥	77	8F	セ
30		Pt	78	90	ソ
31		¤	79	91	タ
32	F0	0	80	92	チ
33	F1	1	81	93	ツ
34	F2	2	82	94	テ
35	F3	3	83	95	ト
36	F4	4	84	96	ナ
37	F5	5	85	97	ニ
38	F6	6	86	98	ヌ
39	F7	7	87	99	ネ
40	F8	8	88	9A	ノ
41	F9	9	89	9D	ハ
42	59	ß	90	9E	ヒ
43	B5	§	91	9F	フ
44	7B	#	92	A2	ヘ
45	7C	@	93	A3	ホ
46	6C	%	94	A4	マ
47	6D	_	95	A5	ミ

For 3278, 192K-Character Set (continued)

Position	Hex (Char)	Character	Position	Hex (Char)	Char
96	A6	ム	144		q
97	A7	メ	145		r
98	A8	モ	146		s
99	A9	ヤ	147		t
100	AA	ユ	148		u
101	AC	ヨ	149		v
102	AD	ラ	150		w
103	AE	リ	151		x
104	AF	ル	152		y
105	BA	レ	153		z
106	BB	ロ	154		æ
107	BC	ワ	155		ø
108	BD	ン	156		å
109	BE	゛	157		ç
110	BF	●	158		$\overline{;}$
111	41	•	159		$\overline{*}$
112	42	「	160	C1	A
113	43	」	161	C2	B
114	44	'	162	C3	C
115	45	·	163	C4	D
116	46	ヲ	164	C5	E
117	47	ァ	165	C6	F
118	48	ィ	166	C7	G
119	49	ゥ	167	C8	H
120	51	ェ	168	C9	I
121	52	ォ	169	D1	J
122	53	ャ	170	D2	K
123	54	ュ	171	D3	L
124	55	ョ	172	D4	M
125	56	ッ	173	D5	N
126	58	ー	174	D6	O
127		"	175	D7	P
128		a	176	D8	Q
129		b	177	D9	R
130		c	178	E2	S
131		d	179	E3	T
132		e	180	E4	U
133		f	181	E5	V
134		g	182	E6	W
135		h	183	E7	X
136		i	184	E8	Y
137		j	185	E9	Z
138		k	186	9E	Æ
139		l	187	80	Ø
140		m	188	67	Å
141		n	189	68	Ç
142		o	190	5E	;
143		p	191	5C	*

Output Translation Tables

The tables below are the *output* translation tables for the 3278 and 3279. Characters that do not have corresponding hexadecimal values in these tables are nonstandard. To refer to these characters, you must either give each of them a unique hexadecimal value when it is specified in the OUTRTBL macro, or use a transparent write operation.

3278 APL Character Set (Output Translation Table)

Position	Hex (Char)	Character	Position	Hex (Char)	Character
0		null	44		unused
1		unused	45		unused
2		unused	46		unused
3		unused	47		unused
4		unused	48	81	‖
5		unused	49		—
6		unused	50	83	|
7		unused	51	84	|
8		unused	52	85	|
9		unused	53	91	▌
10		unused	54	92	▐
11		unused	55	93	▄
12		unused	56	94	▀
13		unused	57	95	█
14		unused	58	82	=
15		unused	59	A3	•
16	40	space	60		unused
17		unused	61		unused
18		unused	62		unused
19		unused	63		unused
20		unused	64		unused
21		unused	65	8A	↑
22		unused	66	8C	≤
23		unused	67	8D	⌈
24		unused	68	8E	⌊
25		unused	69	8F	→
26		unused	70	90	⎕
27		unused	71	9A	⊃
28		unused	72	9B	⊂
29		unused	73	9D	○
30		unused	74	9F	←
31		unused	75	A0	¯
32		unused	76	AA	∩
33		unused	77	AB	∪
34		unused	78	AC	⊥
35		unused	79	AD	[
36		unused	80	AE	≥
37		unused	81	AF	∘
38		unused	82	B0	⍺
39		unused	83	B1	∊
40		unused	84	B2	⍳
41		unused	85	B3	⍴
42		unused	86	B4	⍵
43		unused	87	B6	X

3278 APL Character Set (continued)

Position	Hex (Char)	Character	Position	Hex (Char)	Character
88	B7	\	140	C5	┌
89	B8	÷	141	C6	├
90	BA	∇	142	D6	┤
91	BB	∆	143		
92	BC	⊤	144	F0	^{0} (superscript)
93	BD	]	145	F1	^{1} (superscript)
94	BE	≠	146	F2	^{2} (superscript)
95	BF	|	147	F3	^{3} (superscript)
96	D0	}	148	F4	^{4} (superscript)
97	41	A̲	149	F5	^{5} (superscript)
98	42	B̲	150	F6	^{6} (superscript)
99	43	C̲	151	F7	^{7} (superscript)
100	44	D̲	152	F8	^{8} (superscript)
101	45	E̲	153	F9	^{9} (superscript)
102	46	F̲	154	A4	∩
103	47	G̲	155	D4	┘
104	48	H̲	156	D5	┐
105	49	I̲	157	D7	⊤
106	E1	$_{1}$ (subscript)	158		⍮
107		unused	159		*
108	D2	- (superscript)	160		unused
109	C1	((superscript)	161	C8	§
110	C2	+ (superscript)	162	8B	↓
111	D3	†	163	72	¨
112	C0	{	164	ED	⊖
113	51	J̲	165	EE	⌹
114	52	K̲	166	FB	⍫
115	53	L̲	167	FD	⍟
116	54	M̲	168	EF	⍕
117	55	N̲	169	FE	⍎
118	56	O̲	170	CA	⍲
119	57	P̲	171	CB	⍱
120	58	Q̲	172	71	∧
121	59	R̲	173	78	∨
122	E2	$_{2}$ (subscript	174	CD	⌽
123	9C	⌑	175	CF	⍉
124		unused	176		unused
125	D1	) (superscript)	177		unused
126	9E	±	178	DA	⌶
127	C3	■	179	DB	!
128	FC	⍙	180		unused
129	A1	∘	181		unused
130	62	S̲	182	DC	⍒
131	63	T̲	183	DD	⍋
132	64	U̲	184	C7	⊥
133	65	V̲	185	A2	¯ (superscript)
134	66	W̲	186	DE	⍞
135	67	X̲	187	DF	⍝
136	68	Y̲	188	D8	¶
137	69	Z̲	189	80	∼
138	E3	$_{3}$ (subscript)	190	EA	⌿
139	C4	L	191	EB	⍀

3278A Keyboard Translation Table

Scan Code	Case 1	Case 2	Case 3	Scan Code	Case 1	Case 2	Case 3
00	NOOP	NOOP	NOOP	30	NOOP	NOOP	NOOP
01	BKSP	BKSP	APL OFF	31	NOOP	NOOP	NOOP
02	NOOP	NOOP	NOOP	32	C'Z'	X'089B'	X'0869'
03	NOOP	NOOP	NOOP	33	C'X'	X'089A'	X'0867'
04	NOOP	NOOP	NOOP	34	C'C'	X'08AA'	X'0843'
05	NOOP	NOOP	NOOP	35	C'V'	X'08AB'	X'0865'
06	NOOP	NOOP	NOOP	36	C'B'	X'08AC'	X'0842'
07	NOOP	NOOP	NOOP	37	C'N'	X'08BC'	X'0855'
08	C'7'	X'6E'	X'08ED'	38	C'M'	X'08BF'	X'0854'
09	C'8'	X'08BE'	X'08FD'	39	X'6B'	X'5E'	X'08DF'
0A	C'9'	X'0878'	X'08CB'	3A	X'4B'	X'7A'	X'08EB'
0B	NOOP	NOOP	NOOP	3B	X'61'	X'08B7'	X'08EA'
0C	NOOP	NOOP	NOOP	3C	NOOP	NOOP	NOOP
0D	X'08B6'	X'08B8'	X'08EE'	3D	NOOP	NOOP	NOOP
0E	NOOP	NOOP	NOOP	3E	NOOP	NOOP	NOOP
0F	NOOP	NOOP	NOOP	3F	X'40'	X'40'	NOOP
10	C'4'	X'088C'	X'08DD'	40	NOOP	NOOP	NOOP
11	C'5'	X'7E'	X'08CD'	41	NOOP	NOOP	NOOP
12	C'6'	X'08AE'	X'08CF'	42	NOOP	NOOP	NOOP
13	C'E'	X'08B1'	X'0845'	43	NOOP	NOOP	NOOP
15	NOOP	NOOP	NOOP	44	C'Q'	X'6F'	X'0858'
16	C'0'	X'0871'	X'08CA'	45	C'W'	X'08B4'	X'0866'
18	C'1'	X'0872'	X'08DA'	46	C'R'	X'08B3'	X'0859'
19	C'2'	X'08A0'	X'08FB'	47	C'T'	X'0880'	X'0863'
1A	C'3'	X'4C'	X'08DC'	48	C'Y'	X'088A'	X'0868'
1D	NOOP	NOOP	NOOP	49	C'U'	X'088B'	X'0864'
1E	NOOP	NOOP	NOOP	4A	C'I'	X'08B2'	X'0849'
1F	X'4E'	X'60'	X'08DB'	4B	C'O'	X'089D'	X'0856'
20	NOOP	NOOP	NOOP	4C	C'P'	X'5C'	X'0857'
21	C'A'	NOOP	X'0841'	4D	X'089F'	X'088F'	X'08DE'
22	C'S'	X'08C5'	X'0862'	4E	NOOP	NOOP	X'08FC'
23	C'D'	X'08C4'	X'0844'	4F	NOOP	NOOP	NOOP
24	C'F'	X'6D'	X'0846'	50	NOOP	NOOP	NOOP
25	C'G'	X'08BA'	X'0847'	51	NOOP	NOOP	NOOP
26	C'H'	X'08BB'	X'0848'	52	NOOP	NOOP	NOOP
27	C'J'	X'08A1'	X'0851'	54	NOOP	NOOP	NOOP
28	C'K'	X'7D'	X'0852'	55	NOOP	NOOP	NOOP
29	C'L'	X'0890'	X'0853'	56	CAS3	CAS3	NOOP
2A	X'08AD'	X'4D	X'08FE'	59	NOOP	NOOP	NOOP
2B	X'08BD'	X'5D'	X'08EF'	5A	BLNKCUR	BLNKCUR	NOOP
2C	NOOP	NOOP	NOOP	5B	ALTCUR	ALTCUR	NOOP
2D	NOOP	NOOP	NOOP	5C	CLICKER	CLICKER	NOOP
2E	NOOP	NOOP	NOOP	60	NOOP	NOOP	NOOP
2F	EOM	EOM	NOOP				

3278A Keyboard Translation Table (continued)

Scan Code	Case 1	Case 2	Case 3
00	NOOP	NOOP	NOOP
01	BKSP	BKSP	APL ON
02	NOOP	NOOP	NOOP
03	NOOP	NOOP	NOOP
04	NOOP	NOOP	NOOP
05	NOOP	NOOP	NOOP
06	NOOP	NOOP	NOOP
07	NOOP	NOOP	NOOP
08	C'7'	X'50'	NOOP
09	C'8'	X'5C'	NOOP
0A	C'9'	X'4D'	NOOP
0B	NOOP	NOOP	NOOP
0C	X'79'	NOOP	NOOP
0D	X'7E'	X'4E'	NOOP
0E	NOOP	NOOP	NOOP
0F	NOOP	NOOP	NOOP
10	C'4'	X'5B'	NOOP
11	C'5'	X'6C'	NOOP
12	C'6'	X'5F'	NOOP
13	C'e'	C'E'	NOOP
15	NOOP	NOOP	NOOP
16	C'0'	X'5D'	NOOP
18	C'1'	X'4F'	NOOP
19	C'2'	X'7C'	NOOP
1A	C'3'	X'7B'	NOOP
1D	NOOP	NOOP	NOOP
1E	NOOP	NOOP	NOOP
1F	X'60'	X'6D'	NOOP
20	NOOP	NOOP	NOOP
21	C'a'	C'A'	NOOP
22	C's'	C'S'	NOOP
23	C'd'	C'D'	NOOP
24	C'f'	C'F'	NOOP
25	C'g'	C'G'	NOOP
26	C'h'	C'H'	NOOP
27	C'j'	C'J'	NOOP
28	C'k'	C'K'	NOOP
29	C'l'	C'L'	NOOP
2A	X'5E'	X'7A'	NOOP
2B	X'7D'	X'7F'	NOOP
2C	NOOP	NOOP	NOOP
2D	NOOP	NOOP	NOOP
2E	NOOP	NOOP	NOOP
2F	EOM	EOM	NOOP

Scan Code	Case 1	Case 2	Case 3
30	NOOP	NOOP	NOOP
31	NOOP	NOOP	NOOP
32	C'z'	C'Z'	NOOP
33	C'x'	C'X'	NOOP
34	C'c'	C'C'	NOOP
35	C'v'	C'V'	NOOP
36	C'b'	C'B'	NOOP
37	C'n'	C'N'	NOOP
38	C'm'	C'M'	NOOP
39	X'6B'	X'6B'	NOOP
3A	X'4B'	X'4B'	NOOP
3B	X'61'	X'6F'	NOOP
3C	NOOP	NOOP	NOOP
3D	NOOP	NOOP	NOOP
3E	NOOP	NOOP	NOOP
3F	X'40'	X'40'	NOOP
40	NOOP	NOOP	NOOP
41	NOOP	NOOP	NOOP
42	NOOP	NOOP	NOOP
43	NOOP	NOOP	NOOP
44	C'q'	C'Q'	NOOP
45	C'w'	C'W'	NOOP
46	C'r'	C'R'	NOOP
47	C't'	C'T'	NOOP
48	C'y'	C'Y'	NOOP
49	C'u'	C'U'	NOOP
4A	C'i'	C'I'	NOOP
4B	C'o'	C'O'	NOOP
4C	C'p'	C'P'	NOOP
4D	X'4A'	X'5A'	NOOP
4E	X'E0'	X'6A'	NOOP
4F	NOOP	NOOP	NOOP
50	NOOP	NOOP	NOOP
51	NOOP	NOOP	NOOP
52	NOOP	NOOP	NOOP
54	NOOP	NOOP	NOOP
55	X'4C'	X'6E'	NOOP
56	NOOP	NOOP	NOOP
59	NOOP	NOOP	NOOP
5A	BLNKCUR	BLNKCUR	NOOP
5B	ALTCUR	ALTCUR	NOOP
5C	CLICKER	CLICKER	NOOP
60	NOOP	NOOP	NOOP

Function Key Assignment

Mnemonic	Function
CAS1	Shift to Case 1
CAS2	Shift to Case 2
CAS3	Shift to Case 3
CAS4	Shift to Case 4
BKSP	Backspace
BKSP2	Backspace two positions
ADVAN	Advance
ADVAN2	Advance to positions
NOOP	Null key

These are only for the 3278/3279 displays:

ALTCUR	Alternate cursor
BLNKCUR	Blinking cursor
CLICKER	Clicker key
DEFHI	Default highlighting
BLINK	Blinking
REVERSE	Reverse video
UNDRSCR	Underscore
DEFCOL	Default color
BLUE	Blue characters
RED	Red characters
PINK	Pink characters
GREEN	Green characters
TURQS	Tourquoise characters
YELLOW	Yellow characters
WHITE	White characters
DEFPS	Default program storage
PS1	Program storage 1
PSSLST0	PSSLST EBCDIC table
PSSLST1	PSSLST APL table

Chapter 10. IBM 3279 Display Station Model 2

The 4700 system supports the IBM 3279 Display Stations Models 2A and 2B. The Model 2A is a 4-color display with a 1920-character screen and either the 3279-A data-entry keyboard or the 3279-K standard keyboard. The Model 2B is a 7-color display with a 1920-character screen and either the 3279-A or 3279-K keyboard.

Operating and Application Programming Characteristics

The 3279-2A and 2B have the same components (keyboard and display), controls, and indicators as the 3278-2. Therefore, this chapter describes the unique aspects of the 3279-2A/2B. For information about the components, controls, and indicators these 3279 displays have in common with the 3278-2, refer to that component chapter.

Color Selection

The color selection on a 4-color display is by field, and is controlled by the protection and intensity attributes of the field. The displayed colors relate to the field attributes as follows:

Field Attributes:	*Color:*
Unprotected, Normal Intensity	Green
Unprotected, Intensified	Red
Protected, Normal Intensity	Blue
Protected, Intensified	White

Color selection on a 7-color display is on a field and character basis, as described in Chapter 4, "Processing 3270 Data Streams" under "Programming DATSM for APL and Color Displays."

Convergence Feature

Convergence is the technique of bringing three beams (red, green, and blue) together so that they meet at the same part of the screen at the same time. This is necessary to produce a precise picture in a secondary color or in white. The 3279 convergence is accomplished by the device's convergence feature. From time to time you will use the convergence utility program to adjust this feature.

Compatibility with 327x Controller Utilities

1. In the 327x implementation, you press the Clicker/TEST key while holding the ALT key to invoke the Test Function Router. This program clears the screen, displays the words 4700 TEST in the indicator row, displays the cursor in the top left corner of the screen and waits for a request. To invoke the color convergence function, the operator presses the / key, the 7 key and the Enter key. (A service representative might press /0 to invoke Test 0, or a /8 to invoke Test 8.) These tests display character patterns for diagnosis and maintenance.

In the 4700 implementation there is no Test Function Router. Press the Clicker/TEST key while holding the ALT key to invoke the utility directly. The system clears the screen, displays the characters TEST in the indicator row, and displays a panel of diagnostic and maintenance information. This panel contains the Test 0 character pattern in the top left corner of the screen and, for a 7-color terminal, the Test 8 pattern superimposed in the center of the panel. The system then displays several messages noting the difference between the 327x and the 4700 implementations. The last of these messages is:

```
PRESS SPACE BAR FOR CONVERGENCE TEST 7
```

If you press the Space bar now, the convergence program presents a convergence test pattern. Press Clicker/TEST and hold ALT to clear the screen and return to normal operation. The system ignores any other input.

2. In the 327x implementation, the utility can be invoked by only one display operator at a time. If the utility is executing on behalf of one display operator, and a second operator enters a "/7" request to the Test Function Router, the second operator is given an input-inhibited indication with a "-F" message. The operator can reset this condition and re-try later.

   In the 4700 implementation, the utility can be invoked by any number of display operators concurrently.

3. In the 327x implementation, selected keys are recognized as Reset keys to accommodate various keyboards. Specifically, on the Data Entry Keypunch keyboard, the engraved Reset key is the sixth key from the right on the top row. This key can be used to clear an input-inhibited condition.

   In the 4700 implementation, the only key recognized as a Reset key on any keyboard is the key immediately to the left of the space bar. This key presents a scan code of X'34' when pressed. Note that this fixed definition applies only while the utility is running. During normal use, the Reset key is the one specified during CPGEN.

4. In the 327x implementation, the Clear key ends the utility but retains the Test Function Router, permitting another test to be invoked.

   In the 4700 implementation, the Clear key is not honored while the utility is running. Instead, it causes an input-inhibited condition.

5. In the 327x implementation, the clicker state can be changed via the keyboard while the utility is running.

   In the 4700 implementation, the Clicker key by itself (without ALT) is not honored while the utility is running. Instead, it causes an input-inhibited condition. You can change the state of the clicker before entering the utility by pressing the key that was defined for that purpose during CPGEN. When the utility is running, the clicker state is inverted whenever a Reset key is needed to clear an input-inhibited condition, and re-inverted when the Reset key is pressed.

Configuration

The DCA3279 macro allows specification of a 3279 display. The operands are the same as those on a DCA3278 macro, except for the MODEL operand. On the 3279, Model 2A is coded as MODEL=8A; model 2B is coded as MODEL=8B.

Notes:

1. If the translation table named on the INTRTBL operand defines a function or character for the combination of the ALT key and Clicker/TEST key, no error is declared but the function or character is not honored. Instead, pressing this combination of keys invokes the convergence utility.

2. When attached to a DCA port defined for a 4704-2/3, the 3279 is restricted to accepting and displaying only 0–9, A–Z, and a–z.

Refer to "Programming DATSM for APL and Color Displays" on page 4-45 in Chapter 4, "Processing 3270 Data Streams" for other points to consider when configuring a 3278 or 3279 display.

3278/3279 Substitution

You can substitute the 4700-compatible 3278s and 3279s, subject to the following constraints:

- If a port is defined as a 3278, but a 3279 is attached, the 3279 operates as though it were a 3278. The convergence feature is not recognized, the power-on convergence routine is not invoked, and the combination of the ALT key with the Clicker/TEST key behaves as specified in the user's translation tables. If no specification is given for this combination, the keystrokes are ignored.

- If a port is defined as a 3279, but a 3278 is attached, the 3278 operates as though a 3278 had been defined. The combination of the ALT key with the Clicker/TEST key behaves as specified in the user's translation tables. If no specification is given for this combination, the keystrokes are ignored.

Translation Tables

Refer to the device chapter describing the 3278 display for the translation tables used by the 3279-2A and 2B.

Chapter 11. IBM 3606 and 3608 Financial Services Terminals

This chapter describes the IBM 3606 and 3608 Financial Services Terminals. Typical applications for the 3606 and 3608 are in data capture and account processing at points-of-sale, usually merchant locations. Figure 11-1 and Figure 11-2 show the two terminals. The 3606 and 3608 terminals have a 16-key keyboard, an 8-position numeric display, 9 indicator lights, and a magnetic stripe reader. The 3608 terminal incorporates a 3-line printer. In this chapter, discussions of the keyboard, magnetic stripe reader, display, and indicator lights apply to both the 3606 and 3608.

The controller treats the keyboard, magnetic stripe reader, display, and indicators as one terminal component, and the 3608 printing unit as another terminal component. The 3608 is thus a multicomponent terminal.

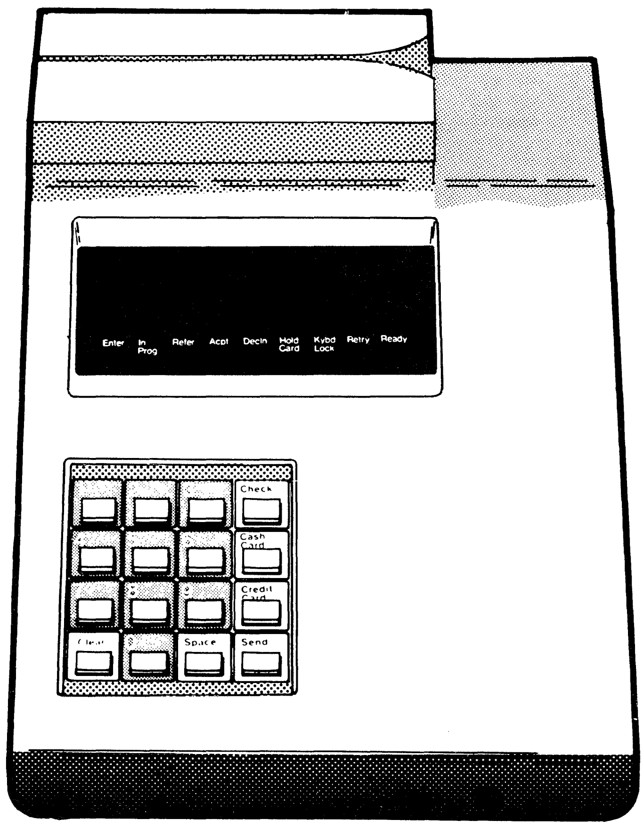


Figure 11-1. IBM 3606 Financial Services Terminal

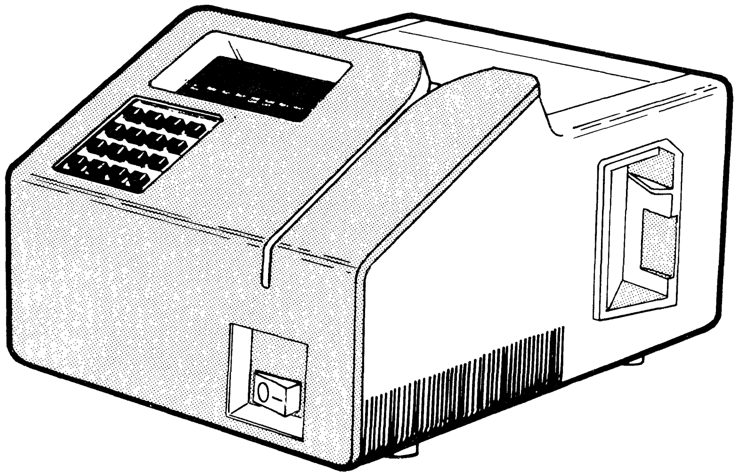


Figure 11-2. IBM 3608 Printing Financial Services Terminal

Use the DEV3606, DEV3608, and DEFADDR configuration macros to describe the terminals to the controller.

Loop Speed and Address Switches: The loop speed and address switches are next to the display and indicator lights under the screen. Four loop speed switches provide for setting the desired rate: 600, 1200, 2400, or 4800 bps. (The 4700 controller does not support 600 bps.)

The 3606 and 3608 terminals are designed for sharing a common loop (base) address and using a subaddress for identifying individual devices. Fifteen 3606 terminals or seven 3608 terminals can share a common loop address.

Eight address switches provide for setting a loop address (1-16) and a subaddress (1-15) for each device. Use these values in the DEFADDR configuration macro when describing the terminal to the controller. The 3608 terminal uses only even subaddresses.

Configurations: Model 1 financial services terminals can be connected to local loops or remote subloops, and can be mixed with other terminals on the same loop. The IBM 3603 Terminal Attachment Unit is used to connect subloops of 3606 Model 1 and 3608 Model 1 terminals to telecommunication lines if no 3604 Model 2, 3, or 4, or 3614 with an integrated modem is present on the subloop.

Model 2 financial services terminals each contain an integrated modem for single unit attachment directly to a remote loop.

Keyboard and Magnetic Stripe Reader

The keyboard and the magnetic stripe reader provide the data input capability of the financial services terminals. The keyboard provides function and numeric keys for identifying transactions and entering data; the magnetic stripe reader reads magnetic stripes encoded to American Banking Association (ABA) standards or encoded on a 3604 or 4704 using the IBM Specifications code. Data that is entered from either unit is stored in a terminal buffer until it is transmitted to the controller.

The Keyboard

The 16-key keyboard consists of 10 numeric keys, 4 user-definable function keys, a terminal function key (Clear), and a system function key (Send). A special feature provides a keyboard arrangement that is compatible with calculators, adding machines, and cash registers.

The Clear and Send Keys

The Clear key erases keyboard or magnetic stripe reader data from the terminal buffer and resets the terminal, thus permitting a user to cancel a transaction and begin data entry again, or to recover from a retry condition (buffer overflow or an internal error in the terminal). In the retry state, the magnetic stripe reader and the keyboard (except the Clear key) are electronically locked. No data can be entered until the buffer is cleared and the terminal logic is reset. No scan code is associated with the Clear key.

The Send key initiates data transmission to the controller and acts as the end-of-message (EOM) key, also. Pressing the Send key signals the controller that data is ready for transmission. The Send key is the last key pressed when a query or message is entered. Pressing the Send key locks the magnetic stripe reader and keyboard (including the Clear key) until the application program in the controller has issued an LREAD for the data and has unlocked the keyboard with an LWRITE.

Pressing the Clear and Send keys erases the display.

Numeric and Function Keys

The numeric keys are tracked on the display screen as they are entered, and placed in the terminal buffer. (Only the numeric keys are tracked; pressing any of the user function keys, or Clear, or Send clears the display.) Characters in excess of 8 are shifted off the 8-position display, but remain in the terminal buffer. The terminal buffer capacity is 56 bytes (including the Send key scan code).

The four user-definable function keys may be defined, as required, by the application program.

The Keyboard Translation Table

The INTRTBL and TRTBHDR configuration macros are used to specify the translation table that is to be used with the 3606/3608 keyboard. Because the terminals do not support end-of-message key designations or "cancel" key designations (these functions are fixed, supplied by the Send and Clear keys), these designations do not have to be made in the configuration macros. Neither do the terminals have shift-case capability; therefore, only one translation table is associated with the keyboard for any given configuration.

The Magnetic Stripe Reader

The magnetic stripe reader on the 3606 and 3608 terminals reads up to 40 data characters (including delimiters and check characters), encoded to American Banking Association (ABA) standards, on the ABA track 2 of the American National Standard magnetic stripe (which has 3 tracks).

Data that is read from the stripe is not displayed and is placed directly into the terminal buffer. The data is not read by the controller and controller application program until the user presses the Send key. The user can enter additional data (up to the terminal buffer capacity of 55 bytes) from the keyboard before pressing the Send key. Magnetic stripe reader data in the buffer is always an integral number of full bytes and always precedes keyboard data. An attempt to read a magnetic stripe after keyboard data has been placed in the terminal buffer results in a retry condition; the operator must use the Clear key to continue.

The data on the stripe must be encoded in the 4-bit, 75 bpi, ABA magnetic stripe code that represents the values X'0' to X'F'. Delimiters, X'B' for "start-of-stripe" and X'C' or X'F' for "end-of-stripe," mark the beginning and end of the encoded data. The data passed to the application program depends on the translation table used.

Magnetic Stripe Reader Translation Table

The MSTRTBL configuration macro is used to build an input translation table for the magnetic stripe reader. Because terminal logic ties the X'C', X'F', and X'B' values to the end-of-stripe and start-of-stripe functions, these particular values cannot be assigned conflicting meanings, but may be assigned translation values or functions. The standard translation table that is used with the magnetic stripe reader is shown later in this chapter. The label of the MSTRTBL macro is used in the DEV3606 and DEV3608 configuration macros to associate the translation table with the magnetic stripe reader.

Keyboard and Magnetic Stripe Reader Read Operations

A logical work station reads data from the keyboard and magnetic stripe reader by issuing LREAD instructions. The data is translated by the controller (as designated in the keyboard translation table) and is placed in the specified segment of work-station storage, starting at the position indicated by the primary field pointer (PFP). Until the read operation is completed and status stored, the work station is in a wait state. The read operation is completed when end-of-data is recognized or the user segment is filled.

Data from the magnetic stripe reader is always transmitted before data from the keyboard.

The segment 1 fields, SMSIML, SMSMSL, SMSTGU, SMSCCD, and SMSDST, are set at the completion of a read operation.

The SMSTGU Field

The SMSTGU field contains the address of the terminal component (X'00' through X'0F') that is currently in use within the terminal group. A read operation can be performed with SMSTGU containing zero, signifying that the terminal to be read is the first that caused an attention. (See "Polling Terminals," next.)

The controller places the address in SMSTGU when handling an asynchronous interrupt and read operation from a 3606/3608 keyboard and/or magnetic stripe reader. All subsequent LREADs (and LWRITEs/WRTIs) are to this address until the SMSTGU field is reset as follows:

- The SMSTGU field is always reset to zero by the execution of an LEXIT instruction.
- The work station can access SMSTGU and set the field to any desired address or to zero.

In an inquiry and reply situation, the work station does not have to reset SMSTGU in order to write to the display, indicators, or 3608 printer associated with the keyboard/magnetic stripe reader making the inquiry. To initiate a read to a specific terminal component, the work station must place the component address in SMSTGU and then issue the read. A status code of X'4001' is returned if no data is ready to be transmitted.

Note: When the application program is doing overlapped I/O operations, SMSTGU should be saved immediately after it is returned by LREAD because it may be altered by subsequent LREAD or LWRITE operations addressed to other LDAs.

Polling Terminals

The work station can poll the individual terminals of a terminal group by addressing each in turn, as noted above, or it can issue LREADs against a zeroed SMSTGU field. When the SMSTGU field is zero and an LREAD is issued to an LDA associated with a terminal group, the first terminal in that group that presented attention to the controller is selected for the read. After processing the message, the work station zeros SMSTGU and issues another read; the next terminal in the attention stack is selected. A status code of X'4001' will be returned when the stack is empty. The work station can then issue an LEXIT instruction or poll another LDA (terminal group).

Keyboard/Magnetic Stripe Reader Lock

The keyboard and magnetic stripe readers of the 3606 and 3608 terminals are electronically locked when the Send key is pressed. No more data entry is allowed until the lock has been reset by the work station (application program). The lock cannot be reset until the work station has issued an LREAD for the data. Reset is accomplished by issuing an LWRITE to the terminal and transmitting the indicator control function code, X'17', along with data bytes indicating the operation that is to be performed. Setting indicator number 7 (Kybd Lock) off resets the lock.

The Display

The display screen displays eight characters on one line. The character generator for the display can generate 15 characters: the 10 numeric symbols, 0 through 9, and five special characters. The special characters can be displayed only as output from the application program. The OUTRTBL configuration macro is used to build an output translation table for the display.

The initial character to be displayed is presented in the right-hand position of the screen, and, as subsequent characters are entered, the preceding characters shift to the left.

If more than eight characters are written to the display by the application program, or tracked from the numeric keyboard, only the last eight characters written or tracked remain on the display.

Any write operation to the display first clears the display. Pressing any key, except a numeric key, clears the display.

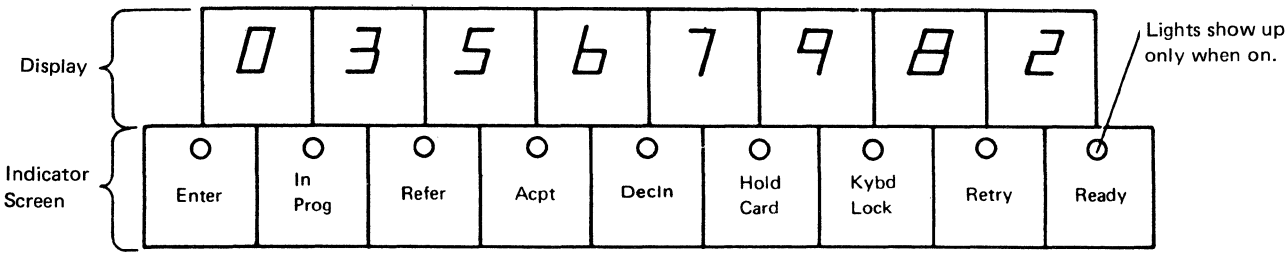


Figure 11-3. Numeric Display and Indicator Lights

The Indicator Lights

The nine indicator lights (Figure 11-3) consist of six indicators that can be set off and on by the application program, two indicators that are internally controlled and have meanings related to the controller subsystem and terminal, and one that is controlled by terminal logic and the work station. The internally controlled indicators are Retry and Ready; the Kybd Lock indicator is controlled by the terminal and the user (work station).

Kybd Lock is set on when the user presses the Send key. It indicates that the keyboard and the magnetic stripe reader are electronically locked and no more data entry is possible. The lock and indicator can be reset only by the work station and only after an LREAD instruction has been issued again to read the data in the terminal buffer.

Retry indicates that terminal buffer capacity has been exceeded (55 bytes), that a magnetic stripe was read after a keystroke, or that an internal keyboard or magnetic stripe reader error has been detected. The magnetic stripe reader and all keys (except Clear) are locked. Pressing the Clear key erases the buffer and resets the terminal so that the data can be reentered.

Ready indicates the status of communications between the terminal and the controller. Normally, the Ready light is on; when off or blinking, the Ready light indicates a problem on the loop.

Display and Indicator Write Operations

The LWRITE or WRTI instructions are used to write to either the display or the indicators. The function code, X'17', is used to control the indicators and it, along with two bytes of data used to describe the operation, can be embedded in a display write or can be written separately. The data is written from a segment of private storage and is pointed to by the secondary field pointer of the specified segment. The primary field pointer of the segment is positioned one byte past the end of the data that is to be written.

The controller translates data using the OUTRTBL configuration macro associated with the display. When the indicator control code (X'17') is embedded in a data string that is being translated, the translation process stops when the code is encountered. The contents of the controller output buffer are written to the display, and the indicator function is performed. Translation and display are resumed as another write operation. Because any write operation to the display causes the previously-written data to be cleared, any data written before encountering the control character is lost. *Any write to the display causes the previous contents of the display to be cleared before the new data is written.*

The work station can write more than eight characters to the display; only the last eight characters written remain displayed at the completion of the write.

The 3608 Printing Component

The 3608 printing component prints a maximum of 3 physical lines on 1-to-3-part cut forms. Printing is done by typewheels that are positioned horizontally in a vertical stack. The character set is engraved on the outer circumference of the typewheel.

The spacing between print lines is determined by the positioning of the typewheels in the stack. Of the 10 possible positions, 3 are selected when the terminal is ordered.

The average printing speed is 15 characters per second.

The standard character set consists of 45 10-pitch (10 characters per inch) alphameric characters (alphabet, numerics, some special characters). As an option, the first (top) typewheel can provide a 7-pitch Optical Character Recognition (OCR) 7B font, consisting of 10 numeric characters.

The 10-pitch alphameric typewheels are inked by operator-replaceable ink rolls. The 7-pitch OCR typewheel is not inked; *printing* is accomplished using impact or double-faced carbon paper.

Minimum and maximum form sizes are:

- Width -- 69.9 mm (2.75 in.) to 82.6 mm (3.25 in.)
- Length -- 122 mm (4.8 in.) to 216 mm (8.5 in.)
- Thickness -- 0.10 mm (0.004 in.) to 0.43 mm (0.017 in.)

Forms move through the printer from right to left, and the print lines run parallel with the long edge of the forms.

Operating Characteristics

The printer motor is normally stopped and started by a combination of inserting a form and pressing the Send key. For a terminal-initiated transaction, this readies the printer to receive any transmission required by the transaction. For work-station-initiated transmission, issuance of write instructions activates the printer. If the printer is not ready, an "intervention required" status is returned, and the work station must signal the operator to ready the printer.

Printing occurs three lines at a time, serially by column. This places a restriction on the positioning control order:

- Horizontal positioning to a previous print position (column) is not allowed, and results in an error condition if attempted.

Data to be printed is presented to the printer as a data stream of control characters and information characters, structured into print lines and pages. Because of the ability to create "pages," more than three print lines can be delimited in the data stream.

Print Line and Page Definition

To group print lines into pages and to print more than one page on a form, the following definitions must be adhered to in structuring the data stream:

- A *print line* is a portion of the data stream consisting of one or more bytes that is delimited by a control character that causes a positioning change from the current line. End of data also ends a print line.

- A *page* is a group of from 1 to 3 print lines delimited by:

  - End of data

  - A new line control character ending the third print line

  - A vertical position change to a line past the third print line

  - A vertical position change to a previous print line

  - A form feed control character

Pages are printed horizontally adjacent to each other across the form, and the width of a page is determined by its longest print line. Spacing between pages in the terminal configuration macro (DEV3608), but can be changed if your program issues the DEVPARM instruction with the desired spacing parameters.

A print line comprises all of the information characters and blank positions between two consecutive form movement commands. You create blank positions in the print line by specifying blank characters, or by issuing control characters that cause spacing in the line's data stream.

Variable Parameters

A set of variable parameters -- initial offset (the number of print positions (columns) to be skipped before printing is begun), maximum print line width, 7-pitch OCR or 10-pitch characters for first (top) line, and spacing between pages -- is used to describe the characteristics of the printing that is to be done. These parameters are specified initially in the DEV3608 configuration macro describing the terminal, and can be altered by the application program when it uses the DEVPARM instruction.

Data and Control Character Translation

The application program (work station) sends information and control requests to the printer by means of write instructions. The data stream specified in the write instruction can consist of information characters only, control characters only, or information and control characters in any combination. The controller inspects each byte of the data stream and considers each byte having a value of X'40' or greater to be an information byte and a byte of value X'3F' or less a control byte. Within these two categories, not all hexadecimal values are valid information or control characters.

The OCR font characters are translated as a subset of the standard 45-character, 10-pitch character set. The 3608 uses a single translation table with the valid OCR characters flagged in the table. When the transparent write control function (X'35') is being used, values X'00' to X'09' are valid for the OCR character set, and values X'00' to X'2C' are valid for the 45-character, 10-pitch character set.

Printer Write Operations

The work station (application program) writes to the printer by issuing LWRITE or WRTI instructions. The data to be written is in a segment of storage, with the start of data pointed to by the secondary field pointer and the primary field pointer positioned one byte past the end of data. Condition code and status for the operation are posted in the SMSCCD and SMSDST fields, respectively, in segment 1 at the completion of the operation.

If an exceptional condition occurs during the write operation, a *residual count* -- the number of data bytes in the data stream that were *not* processed -- are returned in the SMSIML field in segment 1. If no error occurred during the write operation, the residual count is zero. An error ends processing of the data stream, and the residual count can be used to determine where, in the data stream, the error occurred. Data up to the point of error can have been sent to the printer or can remain in the controller's output buffer.

Formatting the Data Stream

Figure 11-4 shows a sample document with print lines and pages identified. The discussion that follows uses the illustration as an example.

The valid control characters used in formatting the data stream for output to the printer are Line Feed (X'25'), New Line (X'15'), Form Feed (X'0C'), and Position (X'34'). An explanation of these control characters appears at the end of this chapter.

	Page 1		Page 2
Page 1	8217 3394 617 576	– **Line 1** –	88016 51175	**Page 2**
	YOUR STORE	– **Line 2** –	JOE SMITH	
	ANYWHERE, USA	– **Line 3**		

Figure 11-4. Printer Output

In structuring the data stream to produce the printed output shown in Figure 11-4, the user placed a new line control character in the data stream after the 6, the E, and the A of lines 1, 2, and 3, respectively. In line 3, besides ending the print line, the control character also acted to define the page and caused the next print line (880...) to be printed as line 1 of the next page after an appropriate page spacing (as specified) had been performed.

A form feed control character following the H in line 2 of the second page would indicate completion of the print operation and cause the form to be ejected. If the data stream did not end with the form feed control character, the form would remain in the printer and the next write operation would resume printing on this form after an appropriate page spacing (as specified in the DEV3608 configuration macro or in the DEVPARM instruction) has been performed.

Position Control

The positioning control character (X'34') precedes a 2-byte sequence. The first of these bytes, a code specifying whether horizontal or vertical positioning is required, also indicates whether the second byte is an absolute or relative signed value. If the second byte is declared to be absolute, it is considered to be the target column or line number. If relative, the target column or line number is obtained by adding the second byte to the current column or line number. Once the target column or line number is determined, positioning proceeds according to the following rules:

- Horizontal Positioning:
  - If the target column equals the current column, no operation is performed.
  - If the target column is less than the current column, an error results.
  - If the target column is greater than the maximum line width value, an error results.
  - If the target column is greater than the current column, and within the maximum line width value, the print line is padded with blanks from the current column to the target column, where printing is resumed.
- Vertical Positioning:
  - If the target line is equal to the current line, no operation is performed. If the target line is less than the current line, printing will resume on line 1 of the next page. All print lines from the current line to, and including, the last print line will be blanked.
  - If the target line is greater than the current line, but not greater than the number of print lines, printing will resume on the target line in the current print position. Intervening lines will be blanked.
  - If the target line is greater than the current line and greater than the number of print lines, printing will resume on line 1 of the next page.

Controller Configuration Macro Instructions

Controller configuration macro instructions are used to specify the physical and logical configuration of each 4701 and its associated terminals. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. Refer to Volume 6 for descriptions of the DEV3606 and DEV3608 configuration macros, and how they are coded.

Terminal Control Characters

Control characters are embedded in the data stream and indicate operations to be performed at a terminal.

The following are the control characters for the 3606 and 3608.

Function	Control Character	Explanation
Form Feed	X'0C'	The display is completely erased and the cursor moves to position 1.
Return	X'0D'	The cursor moves to position 1 on the display.
New Line	X'15'	The cursor moves to position 1 on the display.
Program Operator Communication	X'17'	This 3-byte control string specifies an indicator update operation, and can be used to set or reset any single indicator, or to reset all indicators with a combination on/off mask. The second byte (operation) X'C0' -- no operation X'C1' -- set on individual indicator X'C2' -- set off individual indicator X'C3' -- reset indicators by mask The third byte (mask values) for X'C0' -- any value for X'C1' or X'C2' -- the indicator number (0-7) for X'C3' -- a hexadecimal mask. Each bit represents an indicator, and is 0 if the indicator is to be set off and 1 if the indicator is to be on. The bit positions correspond directly with the positions of the indicator lights.
Line Feed	X'25'	The cursor moves to position 1 on the display.

Function	Control Character	Explanation
Position	X'34'	This 3-byte string controls a column or line position. The second byte is the flag byte: Bits 0-4 are ignored. Bit 5: 0 = Position change is horizontal. 1 = Position change is vertical (no operation takes place). Bits 6, 7 are ignored. The third byte controls position. The positioning byte indicates the number of blanks that will be inserted in the output to the display. If it requires more blanks than the output buffer can contain, the buffer is written and translation continues at the next byte.
Transparent Write	X'35'	A transparent write (a write without translation) begins. The format of this control sequence is: Byte 0: The control character, X'35'. Byte 1: The number of positions in a data string to be displayed. Byte 2 to Byte n: The positions in the data string to be displayed. The valid positions are X'00' to X'0F'.

3608 Printer Control Characters

The following are the control characters for the 3608 printer:

Function	Control Character	Explanation
Select	X'04'	The byte following this control character is ignored; automatic new line is suppressed.
Form Feed	X'0C'	This signifies the completion of printing and the document is ejected.
Return	X'0D'	This control character is ignored (no operation takes place).
New Line	X'15'	A relative vertical skip of one line is performed and printing resumes at the first print position after offset except when the control character occurs on the last print line. For this condition, printing resumes on the first line in the first print position of the next page. **Note:** A page is a grouping of one or more print lines, up to three, and is delimited by one of the following conditions: 1. End of data 2. A New Line control character (X'15') following print line for the third line 3. A vertical tab to a line beyond the third line 4. A vertical tab to a previous print line 5. A Form Feed control character (X'0C')
Program Operator Communication	X'17'	The two bytes following this control character are ignored (no operation takes place).
Line Feed	'25'	A vertical skip of one line is performed with printing resuming on the next line in the current column. All print positions on this line up to the current column will be blank. If the control character is located in the last print line, printing will resume on the first line of the next page.

Function	Control Character	Explanation
Position	X'34'	This 3-byte string controls a line or column position. The second byte is the flag byte: Bits 0-3 are ignored. Bit 4: 0 = Position is an absolute value. 1 = Position is relative to the present position. Bit 5: 0 = Position change is horizontal. 1 = Position change is vertical. Bits 6,7 are ignored The third byte controls position. The positioning byte indicates a line or column number.
Transparent Write	X'35'	A transparent write (a write without translation) begins. The format of this control sequence is: Byte 0: The control character, X'35'. Byte 1: The number of wheel positions to be printed. Byte 2 to Byte n: The wheel positions to be printed. The valid positions are X'00' to X'09' for the OCR print line, and X'00' to X'2C' for the 10-pitch lines.

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or they can be restarted, or wrapped back, to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 3606 and 3608.

3606/3608 Keyboard/Display

Counter	Explanation
1	Loop error checks
2	Terminal address card unit checks
3	Read time-out
4-9	Reserved
10-11*	Number of transactions for the terminal group
12-13*	Cumulative transactions queued (unserviced) when another transaction is received
14-15*	Number of transactions received while the station is not idle
16	Highest number of transactions queued

*These are 2-byte counters.
The count value is the value in the even counter (10, 12 or 14), multiplied by 256, plus the value in the odd counter (11, 13 or 15).

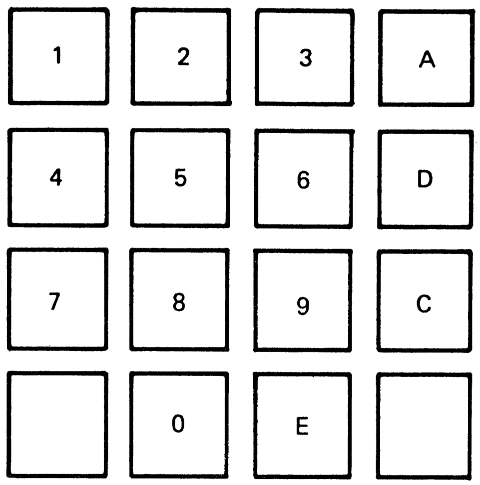
3608 Printer

Counter	Explanation
1	Loop error checks
2	Terminal address card unit checks
3	Printer failures
4	Incorrect message length
5	Intervention required
6	Time-out

Keyboards and Translation Tables

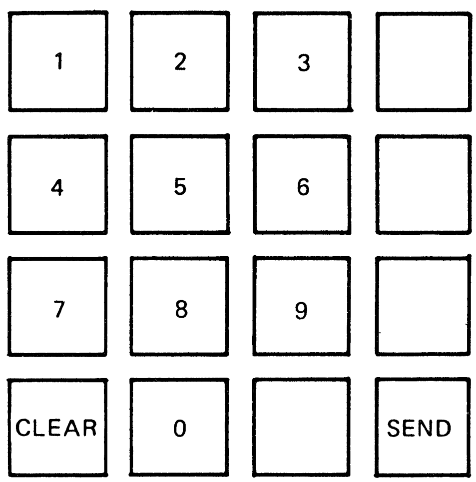
The keyboard and the standard input translation table for the 3606 and 3608 are shown below. The scan-code scheme for the keyboard is also pictured.

3606/3608 16-Key Keyboard



Scan Codes

Scan Code	Case 1
0	X'F0'
1	X'F1'
2	X'F2'
3	X'F3'
4	X'F4'
5	X'F5'
6	X'F6'
7	X'F7'
8	X'F8'
9	X'F9'
A	X'C1'
C	X'C3'
D	X'C4'
E	X'C5'



Engraved Keys

The following table is the standard output translation table for the 3608. Characters that do not have corresponding hexadecimal values in this table are nonstandard. To refer to these characters, you must either give each of them a unique hexadecimal value when it is specified in the OUTRTBL macro, or use a transparent write operation.

For 3608, 45-Character Set

Position	*Hex Value*	*Character*			
0	F0	0 Note 1	23	E3	T
1	F1	1 Note 1	24	E4	U
2	F2	2 Note 1	25	E6	W
3	F3	3 Note 1	26	E9	Z
4	F4	4 Note 1	27	6B	,
5	F5	5 Note 1	28	D1	J
6	F6	6 Note 1	29	D3	L
7	F7	7 Note 1	30	D6	O
8	F8	8 Note 1	31	D7	P
9	F9	9 Note 1	32	D8	Q
10	C1	A	33	D9	R
11	C2	B	34	60	-
12	C3	C	35	5B	$
13	C4	D	36	C5	E
14	D2	K	37	C6	F
15	D4	M	38	C7	G
16	D5	N	39	C8	H
17	E5	V	40	C9	I
18	E7	X	41	4B	.
19	E8	Y	42	50	&
20	7B	#	43	6C	%
21	7C	@	44	61	/
22	E2	S			

Notes:

1. These are the only valid OCR print characters. An attempt to print any other character on the OCR print wheel results in an error (status X'2000'). Only these characters, or characters that translate to these characters can be used as OCR print characters.
2. With the standard translation table for the 3610 Model 4, referring to position 128 results in a space.

Chapter 12. IBM 4710 Receipt/Validation Printer

The IBM 4710 Receipt/Validation Printer (Figure 12-1) is a tabletop receipt/validation and journal printer designed to be incorporated into a teller's work station. The 4710 prints on a variety of single, double, and triple part forms bidirectionally at up to 120 characters per second. The 4710 features:

- Printing on cut forms and journals
- Ten or twelve pitch (characters per inch)
- Bold characters (twice normal width)
- Six-lines-per-inch vertical spacing
- Prints from 1 to 4 lines per document on a cut-form 4-inch print line
- A standard 96-character set
- Selectable special characters for national use
- Sixteen customer-defined characters
- Can be shared between two teller work stations
- Ready and Check indicators
- Three customer-programmable indicators
- Integrated self-testing procedure
- On/Off and Stop switches
- Journal Advance Switch
- Single-part and two-part journal

The journal print station is behind the receipt/validation print station so that all data printed on a receipt is also printed on the journal roll when the journal roll contains impact paper. The journal station can be disabled while the receipt/validation print station continues to operate.

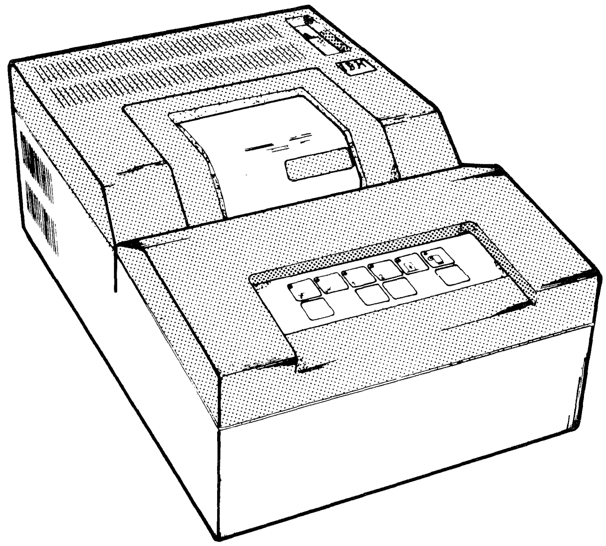


Figure 12-1. IBM 4710 Receipt/Validation Printer

Controls and Indicators

Power On/Off Switch

This two-position switch turns power on and off to the printer. Use this switch with care; turning power on and off at one terminal of a loop may cause errors at other terminals on the same loop.

Start A and Start B Keys

These two keys, marked +P1 and +P2, activate the printer. If the printer is not shared between two work stations, either key activates printing. If the work station is *shared*, the Start A (+P1) key activates printing for the work station identified as A; Start B (+P2) activates printing for the work station identified as B. (Work stations and shared printers are identified during the configuration process.)

Inhibit Print Key

This key stops printing at the end of the current print line and releases any document in the receipt/validation printer.

Journal Advance Key

This key advances the journal paper at approximately two lines per second. When you release this key, the advancing stops.

Test Switches

The 2-position test switches determine which tests are to be run. When the switches are off, no test is run. These switches are at the back of the printer.

Journal Disable/Enable Switch

This 2-position switch, at the back of the printer, enables you to activate and deactivate the journal station without affecting the rest of the printer. If you disable the journal station, the 4710 rejects all attempts to print on the journal.

Ready Light

The Ready light indicates the status of the loop. If the Ready light is on, the printer and loop are operating correctly. If it is off or flashing on and off, the printer or loop is not operating correctly. When it is off, either the power to the printer is off, the indicator has failed, or the printer or loop is not operating correctly.

Check Indicator

When this indicator is on, one of these errors occurred:

- Line Length error
- Print Emitter error
- Translation error
- DEVPARM error
- Command Sequence error
- End-of-forms error
- Self test mode error (indicator blinking)
- Overlength Message error

This indicator is goes off when you correct the error and start a new printing operation.

End-of-Journal-Roll Indicator

This indicator lights when the supply of journal paper is exhausted. This indicator does not light while the journal station is disabled.

Programmable Indicators

Three user-programmable indicators are provided, as shown in the SIGNAL instruction.

Indicator 1 and indicator 2 are programmable indicators and can be lit for any user-defined purpose. Be sure that the operator and the programmer agree on the meanings of these indicators.

The Insert Receipt indicator, although programmable, accompanies the readying of the printer to receive a form for printing. The application program determines when to light this indicator.

Document Printing

The 4710 prints on both a cut form and a journal roll.

Cut-Form Printing

Cut forms are inserted by the operator one at a time. You can use single-part forms, or multipart forms. An example of a cut form is a deposit slip. The 4710 can print on cut forms with these dimensions:

Width	74 to 229 mm (2.91 to 9.0 in)
Length	68 to 152 mm (2.67 to 6.0 in)
Weight	12-to-99 lb stock
Thickness	0.254 mm (0.010 in) maximum single part 0.432 mm (0.017 in) maximum multipart (3 parts plus carbon)

If one part of a multipart form is card stock, it must be the last part. Multipart forms must be fastened on one side *without* metal fasteners. You must not try to print within 6.35 mm (.25 in) of the area used for fastening.

The bottom margin of the cut form must be from 20 to 32.92 mm (from .787 to 1.296 in), depending on the number of lines per page.

At power-on, the 4710 is ready to print cut forms with the parameters provided at configuration.

Journal Printing

You may print on a single-part or two-part journal roll with these dimensions:

Width	114.3 mm (4.5 in.)
Length	30.48 m (100 ft)single part 15.24 m (50 ft) two-part
Thickness	0.0914 mm (0.0036 in.) max single part 0.183 mm (0.0072 in) max two part
Diameter	63.5 mm (2.5 in)
Weight	11 to 15 lb uncoated self-contained carbonless paper

Printing Modes

The printer operates in cut-forms or continuous-forms mode, and may be in a ready or not-ready state. The mode of the printer determines its initial state, and the state of the printer determines how the printer is started. Certain actions cause the state of the printer to change. Figure 12-2 shows the changes that can occur in the state of the printer and the reasons for the changes.

Cut-Forms Mode

In cut-forms mode, the 4710 is initially in a not-ready state. The operator must insert the document to be printed and press a Start Print (+P1 or +P2) key, unless the printer is in cut-forms, automatic-start mode. This action causes the platen mechanism to grip the document and signal the controller that the printer is ready.

The document is released when:

- The application program writes a form-feed control character.
- The application program attempts to index past the defined end of page.
- The application program issues a DEVPARM instruction to redefine the document characteristics.
- The operator presses the Inhibit Print Key.

Any of these actions returns the printer to the not-ready state.

The not-ready state of a printer in cut-forms mode can be further described as either soft-stopped or hard-stopped, depending on the way the printer entered the not-ready state. When the printer is soft-stopped, press a Start Print key before further cut-forms printing can occur. If the mode is changed to continuous forms by a DEVPARM instruction, however, you can start the printer automatically by issuing a write instruction. When the printer is hard-stopped, you must press a Start Print key before any further printing can occur even if the mode is changed by a DEVPARM instruction. The initial not-ready state is soft-stopped.

Continuous-Forms Mode

In continuous-forms mode, a printer is initially in a ready state. In this mode, the printer can operate without an operator (for example, to print teller or administrative reports).

A 4710 in continuous-forms mode is put in a not-ready state if:

- You stop the 4710 by pressing the Inhibit Print key.
- A 4710 printer detects the end of journal forms. The operator can enable successive lines to be printed, one at a time, by repetitively pressing a Start Print key until the end of a page is reached. At that time, the operator can replace the forms and press the Start Print key once more to resume normal printing.
- After a form feed control character has been sent to the printer in cut-forms mode to release the document, the mode is changed to continuous forms by means of a DEVPARM instruction. Pressing the Start Print key is an indication that the cut form was removed from the printer.

Press the Start Print key to ready a printer that is in continuous-forms and not-ready state.

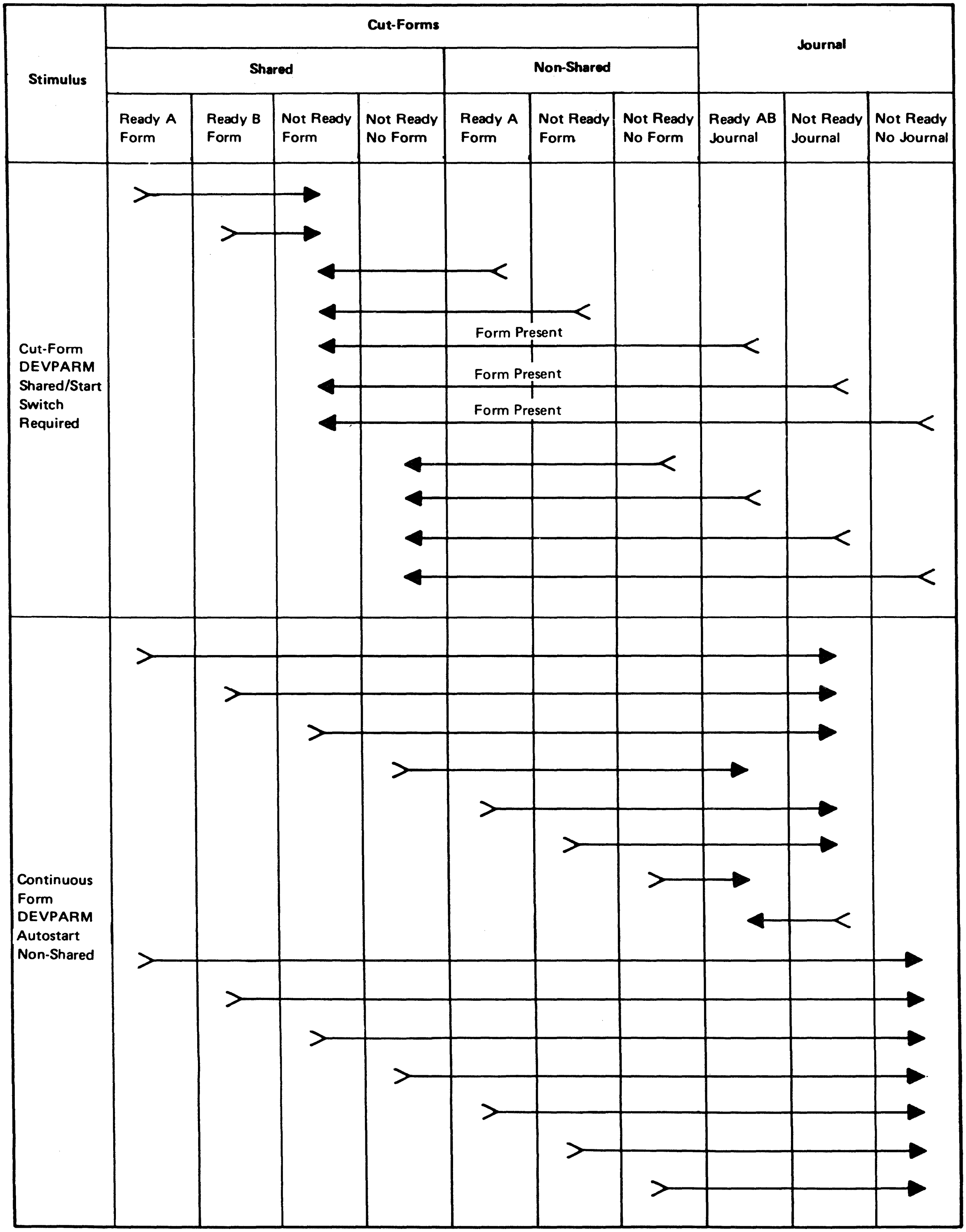


Figure 12-2 (Part 1 of 2). Actions That Change the State of the Document Printer

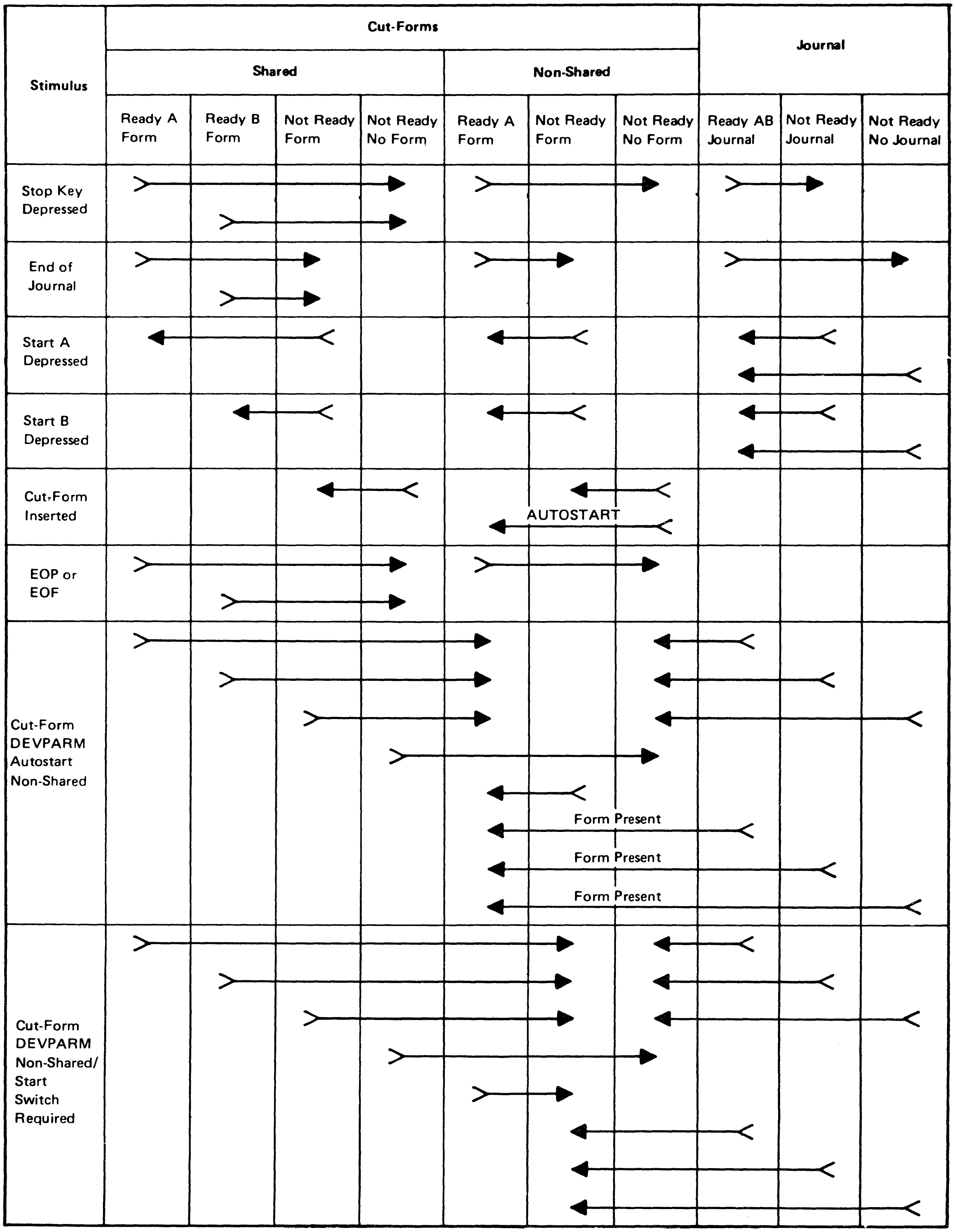


Figure 12-2 (Part 2 of 2). Actions That Change the State of the Document Printer

Translating Data and Handling Control Characters

All data and control characters provided by the application program using LWRITE and WRTI are transmitted directly from the program area to the 4710 without translation.

Variable Parameters

A set of variable parameters describes the physical characteristics of the documents to be printed. The default parameters are specified during configuration, and can be changed when the program issues a DEVPARM instruction. When the 4710 is turned on, the configuration-defined default parameters are in effect.

Validation Printing Parameters

Forms Type: A code indicating the forms type as cut forms (X'01').

Page Size: The number of print lines per page. Indexing past the last line of a page causes unit-exception status to be returned to the program, and the document to be released. The maximum valid page size is 4 lines.

Warning Line: The number of a print line on the page. Indexing past this line causes unit exception to be returned to the application program, and the form to be repositioned one line after the warning line. You can bypass this by defining the warning line as line 0. The warning line number must be less than the page size.

Line Offset: The relative offset to line 1 from the clamping position. The line offset must be less than the page size, and less than the warning line number.

Line Length: This sets the length of a print line. At 10 characters per inch, the maximum length is 40 characters; at 12 characters per inch, the maximum length is 48 characters. If you select 0, the maximum length is used.

Shared/Not Shared: This parameter indicates whether two stations can share this printer. (See Chapter 3, "Programming for Printers" on page 3-1 for more information about sharing printers.) If the printer is not shared, either Start Print key activates printing. If the printer is shared, the appropriate Start Print key must be used: Start A (+P1) for station A and Start B (+P2) for station B.

Start Key Required/Autostart: This parameter indicates whether the operator must press a Start Key to begin printing, or the printer will start automatically.

Automatic Newline: This parameter indicates whether the 4710 moves to a new line when it encounters a data character after a control character at the end of a line. If you do not select Automatic Newline, the application program must request a new line.

10 or 12 Characters Per Inch: This parameter selects the print density at either 10 or 12 characters per inch.

Journal Printing Parameters

Forms Type: A code indicates the type of forms. A X'82' indicates continuous forms with concurrent sharing; a X'02' indicates continuous forms with no concurrent sharing. (See Chapter 3, "Programming for Printers" on page 3-1 for more information on sharing.)

Page Size: This parameter selects the number of print lines per page, to a maximum of 63 lines. If you index past the last line defined for a page, unit exception status is returned to the application program, and the form is positioned to column 1, line 1 of the next page. You can bypass this function by selecting a page size of 0.

Warning Line: This parameter, the number of any print line on the page, selects the warning line. If you index past the warning line, unit exception is returned to the application program, and the form is moved to the line following the warning line. You can bypass this function by selecting 0 as a warning line. The warning line number must be less than the page size, if a page size is specified.

Line Length: This parameter selects the length of a print line. The maximum length is 40 characters at 10 characters per inch, and 48 characters at 12 characters per inch. Length 0 defaults to the maximum for the character pitch.

10 or 12 Characters Per Inch: This parameter indicates whether the print density will be 10 or 12 characters per inch.

Terminal Control Characters

The following terminal control characters, in hexadecimal notation, can be placed in the 4710 data stream to control the printing operation:

Function	Control Character	Explanation
Carriage Return	X'0D'	The logical print element is returned to the extreme left print position; the paper line is unchanged.
Line Feed	X'25'	The paper is spaced up one line; the print position is unchanged.
New Line	X'15'	The paper is spaced up one line and the logical print element returns to the extreme left print position.

Function	Control Character	Explanation
Position	X'34'	This control character is followed by a flag byte and a 1-byte positioning value that control the repositioning of the print element and the paper. The flag byte has the following control: Bits 0-3 are ignored. Bit 4: 0 = Position is an absolute value. 1 = Position is relative to the present position. Bit 5: 0 = Position change is horizontal. 1 = Position change is vertical. Bits 6-7 of the flag byte are ignored. The positioning value indicates a line or column number. It contains an absolute value for use under the direction of the flag byte. A 0 results in no operation.
Form Feed	X'0C'	Release the document for cut-forms printing; skip to a new page for continuous-forms printing.
Select User Defined Characters	X'0E'	Characters in the range X'B0' through X'BF' are found in the user-defined table.
Deselect User Defined Characters	X'0F'	Character selection returns to the base translation table.
Select Bold Print	X'11'	Print the following characters in bold print.
Deselect Bold Print	X'12'	Return print density to normal density.

Programming Considerations

When used with the 4710, the LWRITE, WRTI, SIGNAL, and DEVPARM instructions perform actual output. You can perform all operations *without* the WAIT operand or subsequent LCHECK instruction for faster printing. However, the application program must refrain from using the output data area until the operation requested with the NOWAIT operand is completed. You can achieve maximum throughput by grouping print lines in one LWRITE instruction.

Controller Configuration Macro Instructions

Controller configuration macro instructions are used to specify the physical and logical configuration of each 4701 and its associated terminals. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. Refer to Volume 6 for detailed information on coding the configuration macros.

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or wrap around to begin counting at 128. Use the STATS operand of STARTGEN to indicate your choice. These are the statistical counters for the 4710:

Counter	**Explanation**
1	Loop error
2	Terminal address card unit check
3	Time-out
4	Protocol violation
5-8	Unused
9	Power-on reset
10	Unused
11	A operator active
12	B operator active
13	Controller protocol violation
14	DEVPARM request rejected
15	Incorrect message length
16	Inhibit Print key pressed
17	Line length exceeded
18	Warning line
19	End of form
20	Print emitter check
21	Unused
22	Translate check
23	End of page
24	Unused

Translation Tables

Standard data characters are sent directly from the application program to the 4710 with no translation. Translation occurs in the 4710, according to the tables shown here.

This section describes the translation tables that are used within the 4710. The standard Katakana character set is included in the 4710. Select the national use (Table 5a) special characters with the OUTBHDR and OUTSPEC macros. Select user-defined character sets with the OUTBHDR and CHARDEF macros.

For 4710 Standard Character Set

Hex Value	*Character*	*Hex Value*	*Character*	*Hex Value*	*Character*
F1	1	81	a	56	î
F2	2	82	b	57	ï
F3	3	83	c	5C	*
F4	4	84	d	5D	)
F5	5	85	e	5E	;
F6	6	86	f	60	-
F7	7	87	g	61	/
F8	8	88	h	62	Â
F9	9	89	i	64	À
F0	0	91	j	68	Ç
C1	A	92	k	6B	,
C2	B	93	l	6C	%
C3	C	94	m	6D	_
C4	D	95	n	6E	>
C5	E	96	o	6F	?
C6	F	97	p	71	É
C7	G	98	q	72	Ê
C8	H	99	r	73	Ë
C9	I	A2	s	74	È
D1	J	A3	t	76	Î
D2	K	A4	u	77	Ï
D3	L	A5	v	7A	:
D4	M	A6	w	7D	'
D5	N	A7	x	7E	=
D6	O	A8	y	7F	"
D7	P	A9	z	CB	ô
D8	Q	40		DB	û
D9	R	42	â	DC	ü
E2	S	48	ç	EB	Ô
E3	T	4B	.	FB	Û
E4	U	4C	<	FC	Ü
E5	V	4D	(	FD	Ù
E6	W	4E	+	FF	█
E7	X	50	&		
E8	Y	52	ê		
E9	Z	53	ë		

For 4710 Katakana Character Set (continued)

Hex Value	*Character*
F1	1
F2	2
F3	3
F4	4
F5	5
F6	6
F7	7
F8	8
F9	9
F0	0
C1	A
C2	B
C3	C
C4	D
C5	E
C6	F
C7	G
C8	H
C9	I
D1	J
D2	K
D3	L
D4	M
D5	N
D6	O
D7	P
D8	Q
D9	R
E2	S
E3	T
E4	U
E5	V
E6	W
E7	X
E8	Y
E9	Z
81	ｱ
82	ｲ
83	ｳ
84	ｴ
85	ｵ
86	ｶ
87	ｷ
88	ｸ
89	ｹ
8A	ｺ
8C	ｻ

Hex Value	*Character*
8D	ｼ
8E	ｽ
8F	ﾔ
90	ｿ
91	ﾀ
92	ﾁ
93	ﾂ
94	ﾃ
95	ﾄ
96	ﾅ
97	ﾆ
98	ﾇ
99	ﾈ
9A	ﾉ
9D	ﾊ
9E	ﾋ
9F	ﾌ
A2	ﾍ
A3	ﾎ
A4	ﾏ
A5	ﾐ
A6	ﾑ
A7	ﾒ
A8	ﾓ
A9	ﾔ
AA	ﾕ
AC	ﾖ
AD	ﾗ
AE	ﾘ
AF	ﾙ
BA	ﾚ
BB	ﾛ
BC	ﾜ
BD	ﾝ
BE	ﾞ
BF	•
40	
41	•
42	｢
43	｣
44	,
45	.
46	ｦ
47	ｧ
48	ｨ
49	ｩ
4B	•

For 4710 Katakana Character Set (continued)

Hex Value	*Character*	*Hex Value*	*Character*
4C	<	5E	;
4D	(	60	–
4E	+	61	/
50	&	6B	,
51	ｪ	6C	%
52	ｫ	6D	_
53	ｬ	6E	>
54	ｭ	6F	?
55	ｮ	7A	:
56	ｯ	7D	'
58	ｰ	7E	=
5C	*	7F	"
5D	)	FF	▮

National Use Graphics Table Special Character Sets (5a)

<table>
<tr><th>Hex Value / Country</th><th>4A</th><th>4F</th><th>5A</th><th>5B</th><th>5F</th><th>6A</th><th>79</th><th>7B</th><th>7C</th><th>A1</th><th>C0</th><th>D0</th><th>E0</th></tr>
<tr><td>US</td><td>¢</td><td>|</td><td>!</td><td>$</td><td>¬</td><td>¦</td><td>`</td><td>#</td><td>@</td><td>~</td><td>{</td><td>}</td><td>\</td></tr>
<tr><td>International</td><td>[</td><td>!</td><td>]</td><td>$</td><td>^</td><td>¦</td><td>`</td><td>#</td><td>@</td><td>~</td><td>{</td><td>}</td><td>\</td></tr>
<tr><td>Austria/Germany</td><td>Ä</td><td>!</td><td>Ü</td><td>$</td><td>^</td><td>Ö</td><td>`</td><td>#</td><td>§</td><td>ß</td><td>ä</td><td>ü</td><td>ö</td></tr>
<tr><td>Belgium</td><td>[</td><td>!</td><td>]</td><td>$</td><td>^</td><td>ù</td><td>`</td><td>#</td><td>à</td><td>¨</td><td>é</td><td>è</td><td>ç</td></tr>
<tr><td>Brazil</td><td>É</td><td>!</td><td>$</td><td>Ç</td><td>^</td><td>ç</td><td>ã</td><td>Õ</td><td>Ã</td><td>~</td><td>õ</td><td>é</td><td>\</td></tr>
<tr><td>Denmark/Norway</td><td>#</td><td>!</td><td>¤</td><td>Å</td><td>^</td><td>ø</td><td>`</td><td>Æ</td><td>Ø</td><td>ü</td><td>æ</td><td>å</td><td>\</td></tr>
<tr><td>Finland/Sweden</td><td>§</td><td>!</td><td>¤</td><td>Å</td><td>^</td><td>ö</td><td>é</td><td>Ä</td><td>Ö</td><td>ü</td><td>ä</td><td>å</td><td>É</td></tr>
<tr><td>France</td><td>°</td><td>!</td><td>§</td><td>$</td><td>^</td><td>ù</td><td>`</td><td>£</td><td>à</td><td>¨</td><td>é</td><td>è</td><td>ç</td></tr>
<tr><td>Italy</td><td>°</td><td>!</td><td>é</td><td>$</td><td>^</td><td>ò</td><td>ù</td><td>£</td><td>§</td><td>ì</td><td>à</td><td>è</td><td>ç</td></tr>
<tr><td>Japan</td><td>£</td><td>|</td><td>!</td><td>¥</td><td>¬</td><td>¦</td><td>`</td><td>#</td><td>@</td><td>‾</td><td>{</td><td>}</td><td>$</td></tr>
<tr><td>Portugal</td><td>[</td><td>!</td><td>]</td><td>$</td><td>^</td><td>õ</td><td>`</td><td>Ã</td><td>Õ</td><td>ç</td><td>ã</td><td>'</td><td>Ç</td></tr>
<tr><td>Spain</td><td>[</td><td>|</td><td>]</td><td>Pt</td><td>¬</td><td>ñ</td><td>`</td><td>Ñ</td><td>@</td><td>¨</td><td>{</td><td>}</td><td>\</td></tr>
<tr><td>Spanish Speaking</td><td>[</td><td>|</td><td>]</td><td>$</td><td>¬</td><td>ñ</td><td>`</td><td>Ñ</td><td>@</td><td>¨</td><td>{</td><td>}</td><td>\</td></tr>
<tr><td>United Kingdom</td><td>$</td><td>|</td><td>!</td><td>£</td><td>¬</td><td>¦</td><td>`</td><td>#</td><td>@</td><td>‾</td><td>{</td><td>}</td><td>\</td></tr>
<tr><td>Canadian French</td><td>à</td><td>!</td><td>'</td><td>$</td><td>^</td><td>ù</td><td>`</td><td>#</td><td>@</td><td>¨</td><td>é</td><td>è</td><td>¸</td></tr>
<tr><td>Katakana</td><td>£</td><td>|</td><td>!</td><td>¥</td><td>¬</td><td>¦</td><td>`</td><td>#</td><td>@</td><td>‾</td><td>{</td><td>}</td><td>$</td></tr>
<tr><td>Yugoslavia</td><td>$</td><td></td><td>ć</td><td>Č</td><td></td><td></td><td></td><td>Š</td><td>Ž</td><td></td><td></td><td></td><td>Đ</td></tr>
</table>

Chapter 13. IBM 4720 Forms and Passbook Printers, Models 1-4

The 4720 printers are desk-top cut form/envelope, journal, and passbook printers that perform either transaction or administrative printing. All four printers attach to the banking loop; the following table summarizes their printing capabilities:

Printing Type	Model 1	Model 2	Model 3	Model 4
Cut form	X	X	X	X
Journal		X		X
Passbook			X	X

The 4720 printers all have the following printing characteristics:

- Printing at 10 or 12 cpi, or 16-2/3 cpi for limited report writing.
- Line lengths of up to 82 characters at 10 cpi, 99 characters at 12 cpi, or 132 characters at 16-2/3 cpi.
- Line spacing of 5 or 6 lpi.
- Programmed line spacing in 0.28 mm (0.011 in) steps, using the presentation position command.
- Dot matrix regular with or without bold, or quality printing:
  - 8-dot by 10-dot regular non-bold printing.
  - 8-dot by 20-dot double-width regular bold printing.
  - 16-dot by 20-dot quality non-bold printing (except 16-2/3 cpi or Katakana printing).
  - 16-dot by 40-dot double-width quality bold printing (except 16-2/3 cpi or Katakana printing).
- Fixed standard (latin), non-latin, or Katakana character sets.

CAUTION
Continuous printing for even a short time heats the printing mechanism. Before touching any area of the printing mechanism after performing continuous printing, allow the printer to cool without operating for at least 10 minutes.

Warning: To avoid damaging the printing mechanism, do not print off the edge of the passbook or document, and do not print over holes, folded edges, folds (including the normal fold), stamps, staples, paper clips, or severely warped pages.

Warning: Performing continuous printing operations for more than 10 minutes could damage the printing mechanism.

Controls and Lights

All 4720 printers have an operator console comprising control keys and light-emitting diode (LED) lights; a set of miniature keys to define the printer's address, subaddress, and loop speed; a fourth group of miniature keys are for service use only, and should always be switched off.

Console Switches

The following sections describe the console keys and how they operate.

Power On/Off Switch

This two-position key switches printer power on and off.

Warning: Use this key with care; switching power on and off at one terminal of a loop may cause errors at other terminals on the same loop.

Print A and Print B (+P1/+P2) Keys

These two keys activate the printer. The function of these keys depends on the printer operating mode set by either the DEV4720 configuration macro or a DEVPARM instruction, and whether or not the printer is being shared between operator stations.

When you specify "print key required" in the DEVPARM SETRET parameter list, the operator must press one of these keys to start printing. When DEVPARM specifies "automatic" mode, inserting the cut form or passbook signals the controller to start printing; the Print keys have no effect. On 4720 models with journal printers, pressing the Start keys allows line-by-line printing on the journal when the End of Journal light signals low journal paper.

For an unshared printer, the operator can press either key. When the 4720 operates in manual mode, pressing one of these keys signals the program that printing can begin. On a shared printer, the pressed key indicates which station is requesting cut form or passbook printing; either key can start the journal printing on a shared printer.

Forms Eject

Pressing this key releases a cut form or passbook, and moves the journal tape ahead one 40 mm (6 lpi) line. Any printing in progress completes to the end of the line. The print mechanism moves to the extreme right.

Advance Journal (Models 2 and 4)

When pressed once briefly, the journal tape moves up one 40 mm (6-lpi) line. When you press and hold the key for a second, the printer moves the last line printed up to the tear bar for removal. This key has no effect when either printer interface is active, or when a document is in the printer.

Console Lights

All models of the 4720 have two printer status lights and three programmable lights. Models 2 and 4 have an additional journal light. These lights and their meanings are:

Ready

Indicates power is on and if the printer is ready for loop operation:

- Off: Printer power is off, or printer is not attached to the B-loop.
- Blinking: Printer is on, but loop is not operable.
- On: Printer is active and connected to a working loop.

Check

Indicates one of the following mechanical or programming errors:

- Hardware (mechanical) problem.
- Print line too long.
- Print message too long.
- DEVPARM format error.
- Translation error.

End of Journal Form (Models 2 and 4)

Approximately 15 cm (6 in) of journal paper remains. To continue printing on the remaining paper, the operator can press either Print key to print each line remaining in the output message.

Programmed Lights

The operation of these three lights depends on your application program, which keys them on and off with the SIGNAL instruction. See Chapter 5, "4700 Loop and DCA Assembler Instructions" for a description of controlling lights with the SIGNAL instruction.

The first light is the "Insert Forms" light, which your program keys on and off at the appropriate time. The meanings of the other two program-controlled lights depend on your institution's particular application.

When power is first switched on, all lights except Ready have meanings different from their normal ones. The automatic power-on reset/test sequence uses these lights to show test progress, and the Check light flashes to indicate that a print example can be taken. A 4720 printer attached to the loop switches on the Ready light and sends the final reset/test sequence status to the controller.

If the power-on reset/test detects device errors, your program cannot change or reset these lights. Refer to the device operating procedures for a description of the light meanings and sequence during the power-on reset/test.

Address, Subaddress, and Loop Speed Switches

These pencil-actuated miniature keys select the basic operating characteristics of the 4720 printer. These keys are set when the printer is attached to the loop, and are rarely changed unless the printer is reinstalled in another station.

The address keys select the loop slots that carry the printer's print and control data. The subaddress keys expand the printer addressing capability to allow up to 15 components to use the same address-selected slot. The subaddress keys select the same even subaddress for the cut forms, passbook, and journal printer. The loop speed keys are set to the appropriate operating speed (600, 1200, 2400, or 4800 baud) for that loop.

Note: The 4700 controller does not support a speed of 600 bps.

4720 Forms

The 4720 prints on passbooks, cut forms, and journal-roll forms with the following dimensions:

Passbook Dimensions

Width: 101 to 218 mm (4.00 to 8.60 in)

Height: 10 to 210 mm (0.39 to 8.27 in)

Opened Thickness (one cover plus pages, one side of fold):

Horizontal fold: 0.18 to 1.12 mm (0.007 to 0.044 in)

Vertical fold: 0.18 to 1.27 mm (0.007 to 0.050 in)

Cut Form/Envelope Dimensions

Cut forms can have the following dimensions:

Width: 99 to 228 mm (3.9 to 9.00 in)

Height: 69 to 368 mm (2.7 to 14.5 in)

Thickness (including carbons):

Single-part: 0.10 to 0.28 mm (0.004 to 0.011 in)

Multi-part: 0.15 to 0.43 mm (0.006 to 0.017 in)

The 4720 can print envelopes ranging up to 240 mm (9.5 in.) long and up to 114 mm (4.5 in) wide.

Journal Form Dimensions

Width: 21.60 cm ± 0.38 mm (8.50 ± 0.015 in)

Maximum Journal Roll Diameter: 6.0 cm (single- or multi-part forms)

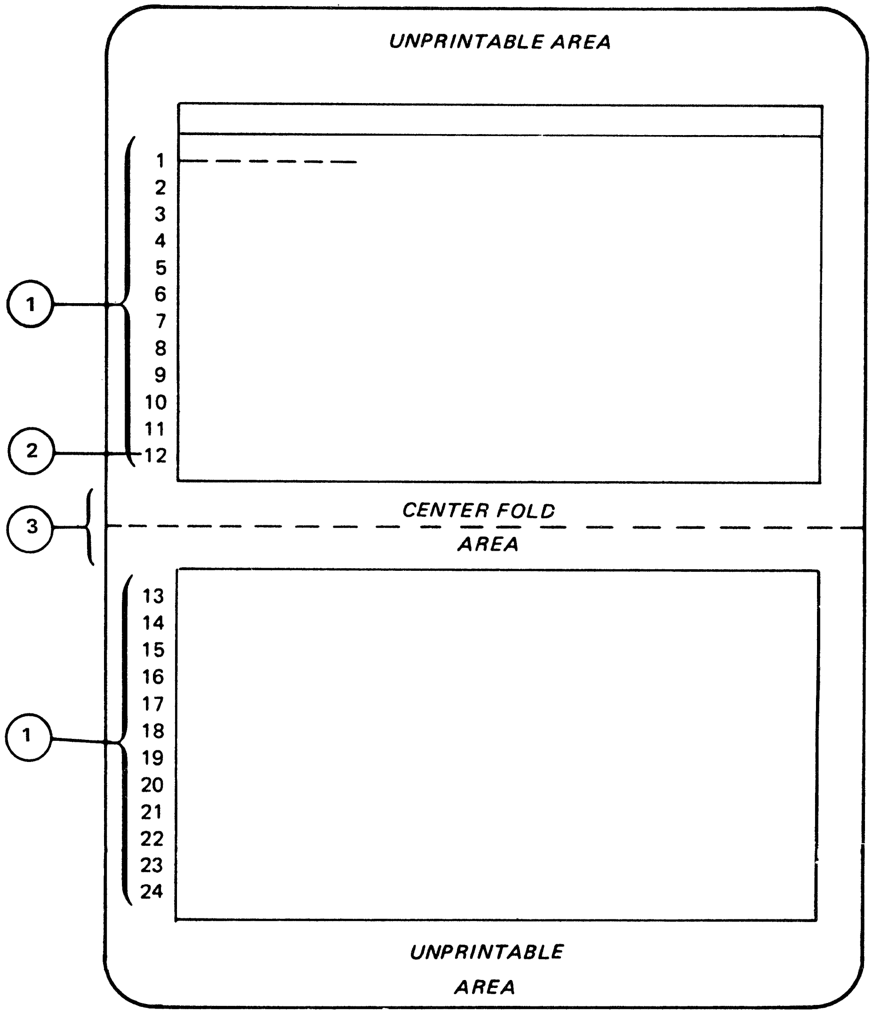
Detailed information concerning forms and paper stock requirements for the 4720 printers is in the *4700 Installation Planning Guide*, GC31-2018.

Passbook Printing

The 4720 can print on passbooks with the following dimensions, measuring to either the horizontal center or to the top or bottom of the character: Margins must be at least 4.0 mm (0.156 in) from either edge of the passbook, at least 6.4 mm (0.25 in) from the top or bottom of the cover or page, and 6.4 mm (0.25 in) above the bottom of a short page or the top of a cutout, notch, window, or other edge. A vertical fold requires at least 7.87 mm (0.31 in) to the left and right of the center fold. A horizontal fold requires at least 6.35 mm (0.25 in) above and below the center fold.

Note: Page size, as defined in the DEVPARM instruction, is the limit to which the printer allows a passbook to be indexed. However, if a passbook of smaller physical size than the passbook defined to the printer is inserted into the 4720, it becomes possible to index off of the page, causing forms movement problems and end-of-forms status to occur.

When passbooks are being printed, four factors must be considered: (1) line finding, (2) page definition, (3) end of page, and (4) center-fold detection. A typical passbook is shown in Figure 13-1.



Legend:
1. Logical print lines that make up the page size
2. Center-fold begin
3. Center-fold skip (at least 6.35 mm [0.25 in.] from the center fold to the center of any character)

Figure 13-1. An Example of a Horizontal-Fold Passbook

There are two common ways to find the next available line for printing in a customer's passbook. The application program can store the next available line number as part of a customer's account record in the host's data base, or the operator can enter the line number at the keyboard during the transaction.

Page Definition

A page is defined as the total available printing area on an open passbook. A horizontal center-fold passbook has unprintable areas above and below the center fold. A vertical-fold passbook has unprintable areas to the left and right of the center fold.

End of Page

An end-of-page condition is reported to the application program by means of the condition code and status bits.

Center-Fold Detection

The last print line or character column before a center fold, and the number of lines or columns to skip must be specified during program execution by the DEVPARM instruction. These values must be specified correctly to avoid printing on the center fold. The application program should use the center-fold device characteristics in the DEVPARM parameter (center-fold begin and center-fold skip) to avoid printing across a center fold (allow at least three character positions for a vertical center fold).

Document Printing

The 4720 can print on cut forms, passbooks, or journal forms. You can also print both cut forms and journal forms simultaneously by specifying journal advance parameters in the DEVPARM cut forms extended parameter byte.

Note: The 132-character line/16-2/3 cpi printing allows limited report printing. *Do not* use any 4720 for extended report printing operations longer than 10 minutes.

Cut Forms and Passbooks

Cut forms and passbooks are inserted one at a time by the operator. An example of a cut form is a deposit slip. Your program controls cut form and passbook printing with the appropriate parameters. In cut forms mode, you can also print on and advance the journal form on Models 2 and 4.

Journals

The journal printer (Models 2 and 4 only) permits printing on rolled paper stock. You control the journal printer with the DEVPARM continuous forms parameters. The journal uses one- or two-part paper. The first part or one-part paper exits past a tear bar after being printed. The journal take-up area stores the second part of the two-part paper on a reel under the locked cover.

The journal printer detects an end-of-forms condition when the end of the continuous forms passes a sensing key, indicating at least 15 cm (6 in) of journal paper remain. Until the operator loads more paper (resetting the key), intervention-required (out-of-forms) status is returned in SMSDST whenever the application program attempts a print operation.

Printing Modes

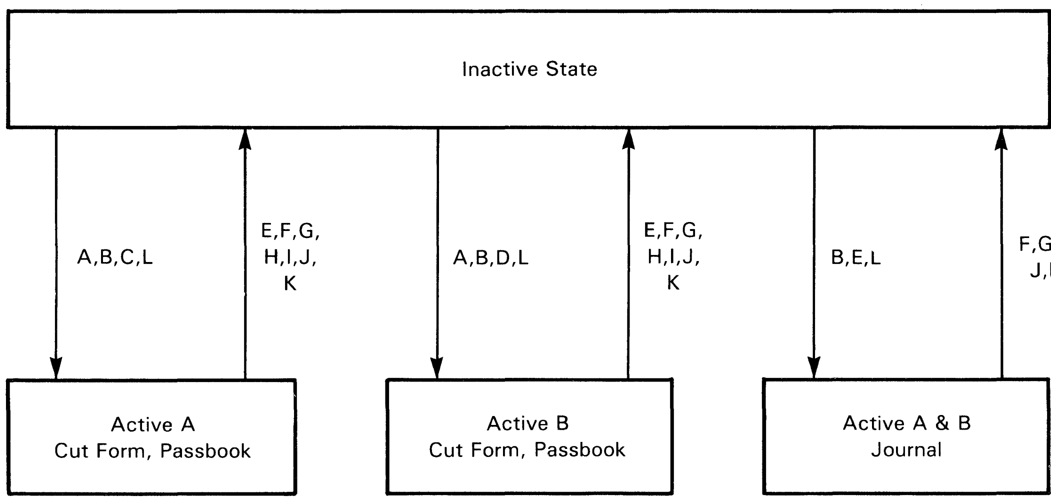
The printer operates in cut forms, continuous forms, or passbook mode, and may be in a ready or not-ready state. All printer models enter cut forms mode when they are switched on. The mode of the printer determines its initial state, and the state of the printer determines how the printer is started. Certain actions cause the state of the printer to change. Figure 13-2 shows the changes that can occur in the state of a printer and the reasons for the changes. The application program can set the mode using the DEVPARM macro.

Cut-Forms Mode

In cut forms mode, the printer is initially in a not-ready state. The operator must make it ready by inserting the document to be printed and pressing a Print key, unless the printer is in cut-forms automatic-start mode. Then the printer signals the application program that the form is properly inserted and that printing can begin.

The 4720 releases the document when:

- The application program changes print characteristics with DEVPARM, or issues a form feed control character.
- The operator presses the Forms Eject key.



ACTION:	NOTES:
A. Document registered	Auto Start, non-shared operation
B. Print A (+P1) or B (+P2) pressed	Non-Shared operation, document already present
C. Print A (+P1 pressed)	Shared operation, document already present
D. Print B (+P2 pressed)	Shared operation, document already present
E. Journal DEVPARM Accepted	If cover not opened and hardware error exists
F. Power on self-test, loop adapter reset, write header reset	
G. Form Eject pressed	If interface already active
H. Form Feed command processed	
I. End of Forms reached	Journal Form will not be ejected
J. Cover opened	Form will not be ejected
K. Cut form or passbook DEVPARM accepted	Current DEVPARM not compatible with new DEVPARM
L. Either +P1 or +P2 will reactivate interface that was deactivated by end of journal form.	

Figure 13-2. Actions That Change Printing States

Any of these actions returns the printer to the not-ready state, and sends status to the program.

Passbook Mode

With respect to its ready and not-ready states, a printer in passbook mode operates in a manner similar to one in cut-forms mode. Figure 13-2 shows the changes that can occur in the state of a passbook printer and the reasons for the changes.

Continuous-Forms Mode

In continuous-forms mode, a printer is initially in a ready state. In this mode, the printer can operate without an operator (for example, to print teller or administrative reports). Figure 13-2 shows changes that can occur in the state of the 4720 printer when printing journal forms, and the reasons for the changes.

A printer in continuous-forms mode is put in a not-ready state if:

- The operator stops the printer by pressing the Forms Eject key or opening the front cover.
- The program issues DEVPARM, requesting document processing.
- A form-feed control character is detected and the current device parameters indicate that a Print key is required.

A printer in continuous-forms mode in a not-ready state can be made ready by pressing a Print key (after closing the cover, if it was opened).

Programming Considerations

The following topics are points to consider when programming the 4720:

Default Printer Status

The following are the default printer mode, page size, and other DEVPARM-controlled parameter settings that are set when a 4720 is switched on:

Setting: *Default:*

Flag/Mode
X'01' (cut forms mode)

Page Size
X'00'; 67 lines at 5 lpi.

Warning Line
X'00'; no warning line

Step Offset
X'00'; no step offset

Line Offset
X'00'; no line offset

Line Length
X'00'; 82 regular non-bold characters at 10 cpi.

Device Characteristics (cut forms):
Unshared, print key required, no device characteristics extension (regular font, no journal advance with cut forms, and skew of 1.37mm/100mm or 0.05 in/4 in), automatic new line for oversized lines, 5 lines and 10 characters per inch.

Refer to "Control and Status" on page 13-11 and to Figure 13-2 for information on setting and changing the printer operating modes and parameters.

Selecting The Types of Printing

Before you begin printing, you must define the print font (normal or quality) either during configuration on the DEV4720 configuration macro, or in your program with the DEVPARM instruction. The configuration default is normal (FONT=N) font. Either choice is valid for all languages except quality printing of Katakana. If you specify this option, the 4720 prints blanks (X'40').

You can choose either regular or bold printing of either normal or quality print fonts by specifying the bold printing terminal control character in the print data. Bold printing remains active until you specify a regular printing terminal control character, or until the operation ends.

If printing will be on dark or shaded forms, you should either select quality printing or overprint the information for the most legible printing.

Translating Data and Processing Control Characters

All data and control characters issued by the program using LWRITE and WRTI go directly to the 4720 without translation.

Suggestions for Faster Printing

The LWRITE, WRTI, SIGNAL, and DEVPARM instructions all perform actual output. You can code all these instructions *without* the WAIT operand or a subsequent LCHECK instruction to permit faster printing. However, the program should not use the output area until the I/O operation specifying NOWAIT completes. You should also group as many print lines as possible into each LWRITE operation.

When you space more than one line space or print multiple character spaces, use the presentation position control character instead of actual line or character spaces to define the new print location.

Addressing

The 4720 printers perform passbook, journal, and cut forms printing in the same physical print area. The subaddress for all three printing functions is the same, and is unlike other 4700-attached printers with one subaddress for the journal printer and another for the cut forms/passbook printer. The subaddress, however, can be any even value ranging 2 through 14.

Defining Printed Characters

You can define special characters for either normal or quality printing, using the CHARDEF configuration macro. Refer to the CHARDEF macro description in the *4700 Programming Library, Volume 6* for detailed information on defining characters.

Control and Status

Control for the three 4720 operating modes comes from the application program. You can change from one mode to another by issuing the DEVPARM instruction with a parameter list containing the mode change in the first (flag) byte and the revised operating characteristics in the remaining bytes.

Using DEVPARM, your program changes the following operating characteristics within each mode. For more information on specifying DEVPARM, refer to Chapter 5, "4700 Loop and DCA Assembler Instructions."

Setting the Skew Threshold for Passbooks and Cut Forms

Documents inserted in the 4720 that are skewed above a certain threshold are returned to the operator for retry. The system-supplied default thresholds provide good registration and ease of operation for most applications. If you want to customize these values, use the DEVPARM macro.

Specify a larger threshold value to cause the 4720 to accept a document entered with more skew. You can use this in an application where ease of insertion is important, and where the additional skew will not affect the output. For example, you might use this on a cut form that does not have specific, preprinted entry areas. Do not allow too much skew for passbook printing. Passbooks are inserted many times to print on the same page, and some lines can be overprinted if the book is inserted with some skew on one insertion and the opposite skew on the next.

You can specify a smaller skew threshold for passbooks. The operator must be more careful when inserting the document, but there will be less chance of overprinting lines.

Set the skew threshold with bits 6 and 7 of the Extended Device Parameter byte in the cut form and passbook versions of the DEVPARM macro (as shown in Chapter 5):

Cut-Form DEVPARM

Bit Configuration	**Allowable Skew**
x'00' (default)	1.37 mm (0.05 in) per 100 mm (3.94 in) of line length
x'01'	2.05 mm (0.08 in) per 100 mm (3.94 in) of line length
x'10'	2.74 mm (0.11 in) per 100 mm (3.94 in) of line length
x'11'	3.42 mm (0.13 in) per 100 mm (3.94 in) of line length

Passbook DEVPARM

Bit Configuration	**Allowable Skew**
x'00' (default)	1.37 mm (0.05 in) per 100 mm (3.94 in) of line length
x'01'	0.68 mm (0.03 in) per 100 mm (3.94 in) of line length
x'10'	2.05 mm (0.08 in) per 100 mm (3.94 in) of line length
x'11'	2.05 mm (0.08 in) per 100 mm (3.94 in) of line length

Setting Passbook Parameters

You can set and change the following operating characteristics by issuing DEVPARM with a flag byte of X'00' to set passbook mode. Refer to the DEVPARM instruction description for the specific bit settings.

Page size: Specify this logical page size as a value ranging 1 to the most lines allowed for the passbook being used. Specifying zero (0) forces the maximum allowable size, either 38 lines (5 lines per in.) or 46 lines (6 lines per in.); refer to "Device Characteristics", bit 06.

Begin Centerfold: Specify the last line (for horizontal-fold passbooks) or character (for vertical-fold passbooks) to be printed before skipping over the passbook fold. For vertical folds, this may be a value ranging 1 to the maximum allowable form width (refer to "Line Length"); for horizontal folds, the value can range 1 to the maximum allowed by the page size. In either case, specifying zero (0) centers the beginning of the centerfold to the nearest integral character or line, as appropriate.

Centerfold Skip: Specify the total number of characters (vertical centerfold) or lines (horizontal centerfold) to be skipped for the fold. This value can range from 0 to 255, as long as the total of this value and the appropriate line or character count is not greater than the total allowed for the form.

Step Offset: Specify a hexadecimal count of steps that, when added to the line offset, starts the first print line at the desired point on the passbook. A step is 0.28mm (0.011 in.). There are 18 steps per line if you specify 5 lines per inch, and 15 steps per line if you specify 6 lines per inch. You can specify any value ranging 0 to 255 steps as long as the total step/line offset plus centerfold skip and page size is not greater than the total lines allowed for a page. The printer adds the total offset to the minimum 6.3mm (0.25 in.) distance allowed from the passbook's edge.

Vertical Line Offset: Specify this hexadecimal count of the number of line spaces to print the first line beyond the minimum distance from the top edge of the passbook. You can specify any value ranging 0 to 255 lines as long as the total step/line offset plus centerfold skip and page size is not greater than the total lines allowed for a page. The printer adds the total offset to the minimum 6.3mm (0.25 in.) distance allowed from the passbook's edge.

Horizontal Character Offset: Specify this hexadecimal count of the character spaces that the first printed character is to be offset beyond the minimum 4.0mm (0.16 in.) distance from the left edge of the passbook. This value can range 0 to 255, as long as the vertical offset plus line length does not exceed the total allowable line length in characters.

Line length: Specify a hexadecimal character count that defines the passbook line length in printable characters per line. This value, which must be based on the characters per inch specified under "Device Characteristics", can be any value ranging 1 to the maximum allowable character count permitted by the passbook: 82 regular characters at 10 regular characters per inch (41 for bold printing), 99 at 12 regular characters per inch (49 for bold printing), or 132 regular characters per inch (66 for bold printing). If you specify zero (0), the maximum length appropriate for the specified printing density is assumed.

Device Characteristics: This last passbook parameter list byte specifies the type of passbook fold, sharing or nonsharing printer, automatic or manual start of printing, whether or not a parameter list extension byte exists, the action to take if a line is too long (force a new line or issue "line length exceeded" check), and the count of lines and characters per inch. If you specify the extension byte, you can also specify print skew characteristics and 16-2/3 cpi printing for lines up to 132 characters long. Refer to the DEVPARM instruction description in Chapter 5, "4700 Loop and DCA Assembler Instructions" for details on coding these parameters.

Setting Cut Form Parameters

Set the following operating characteristics with a DEVPARM instruction having a X'01' flag byte in the parameter list for cut forms:

Page size: Specify this logical page size as a hexadecimal value ranging 1 to the most lines allowed per page for the lines per inch that you choose. Specifying zero (0) forces the maximum allowable size, either 67 lines (5 lines per in.) or 81 lines (6 lines per in.); refer to "Device Characteristics", bit 06.

Warning line: When the printer steps past this line, which can be any line on the form, a unit-exception status is sent to the application program and the form is positioned at the line following the warning line. The application program can use the warning line to indicate when to print footings or when to skip to a new page. Specify a line number less than the total page size; however a warning line value or page size of 0 causes no warning line to be set.

Step Offset, Line Offset, and Line Length.: Specify these values the same as you would for passbook mode.

Device Characteristics: This last DEVPARM parameter list byte specifies sharing or non-sharing printer, automatic or manual start of printing, whether or not an added extension byte is in the parameter list, the action to take if a line is too long (force new line or issue a device check), and lines and characters per inch.

Additional device characteristics (4720 print font, journal control, 16-2/3 cpi printing, and print skew parameters) can be specified in an extension device characteristics byte. Refer to the DEVPARM instruction description in Chapter 5, "4700 Loop and DCA Assembler Instructions" for details on setting the cut form parameters.

Setting Journal Parameters

You can set the following operating characteristics with a DEVPARM instruction having a X'02' flag byte at the beginning of the parameter list for journal printing:

Page Size: Specify this as a hexadecimal value ranging 1 up to the limit of 67 at 5 lines per inch or 81 at 6 lines per inch. Specifying zero (0) causes an undefined maximum size, and no page size checking occurs.

Warning line: When the printer steps past this line, which can be any line on the form, a unit-exception status is sent to the application program and the form is positioned at the line following the warning line. The application program can use the warning line to indicate when to print footings or when to skip to a new page. Specify a line number less than the total page size; however a warning line value or page size of 0 causes no warning line to be set.

Line Length: Specify this value as you would for passbook or cut form printing.

Device Characteristics: This final journal printer parameter byte defines the line count per centimeter or inch, the characters per centimeter or inch if the extended device characteristics byte 5 is present. Byte 5 allows you to select any of the three possible print pitches (10, 12, or 16-2/3 cpi).

Reading 4720 Parameters and Status

By issuing a DEVPARM SETRET=RETURN, you can read the current 4720 operating parameters into the ten bytes beginning at the location defined by DEVPARM. DEVPARM reads the parameters in the same order as they are issued and therefore can be reissued to the 4720 from the same parameter list location.

Terminal Control Characters

To control the 4720 while it is printing, you must insert the following control characters in the output data:

Form Feed (X'0C')
Releases the cut form or passbook, or skips to the next page, as defined for journal printing.

Carriage Return (X'0D')
Sets the current print position at the extreme left side of the current line.

Enter User-Defined Character Set (X'0E')
Causes all subsequent output data to be translated according to the user-defined character set.

Enter Standard Character Set (X'0F')
Returns the printer to the standard translation table. All subsequent output data translates according to that table.

Bold Printing (X'11')
Enables bold (double-width) printing of all subsequent output characters until a "regular print" (X'12') is detected. Your program must allow for the doubling of character width (halving the line length) when printing in bold.

Regular Printing (X'12')
Returns from the bold print to regular (single-width) character printing mode. Your program must readjust the line length, if necessary.

New Line (X'15')
Advances the cut form, journal, or passbook to a new print line.

Line Feed (X'25')
Advances the cut form, journal, or passbook to a new print line without moving the print mechanism.

New Print Position (X'34ccaa')

Moves to a new print position defined by the following control (*cc*) and amount (*aa*) bytes:

Byte 1 (Control)

Enter this byte with the following control bits:

Bits 00-02

unused

Bit 03-Line/Step Positioning

Set to '0' for line (vertical) or character (horizontal) positioning, or to '1' for vertical step positioning. Each vertical step is 0.28 mm (0.011 in.).

Bit 04-Relative/Absolute Positioning

Absolute positioning (bit 04 set to '0') moves the printing to a point measured from either the first line or step or the leftmost character position on the page. Relative positioning (bit 04 set to '1') moves the printing to a new position measured from the current position.

Bit 05-Horizontal/Vertical Positioning

Set bit 05 to '0' for horizontal (up or down) positioning, or to '1' for vertical (left or right) positioning.

Bits 06-07

unused

Byte 2 (Positioning Amount)

Specify a hexadecimal value ranging 1 to 255 that is the count of lines, steps, or characters for repositioning the printing. Specifying zero ('0') causes a "no operation".

Note: You cannot reposition the 4720 vertically to a line or step above the current position. If you specify an absolute line value less than the current absolute line, the passbook/cut form printer advances to the line following the last line on the current page. The journal printer advances to the selected position on the next page.

You can specify an absolute character position that is less than (to the left of) the current absolute position; the current print position changes to that position on the current line.

Controller Configuration for the 4720

The 4720 printers require at least one DEV4720 device configuration macro for each model installed on your system, at least one OUTBHDR table pointer configuration macro to select the macros that define the translation tables and user-defined special characters, and the translation table (OUTSPEC) and character definition (CHARDEF) macros selected as default values. Refer to Volume 6 of this programming library for a detailed description of how to code the configuration macros.

Statistical Counters

The statistical counters record various types of events that occur when printing on the 4720. The counters reach a maximum of 255 before either stopping or beginning the count again ("wrapping") from 128, depending on what is specified on the STATS parameter of the STARTGEN configuration macro.

Counter:	**Definition:**
1	Loop error check
2	Terminal address card check
3	Time-out
4	Terminal protocol violation
5-8	(reserved)
9	Power-on test completed
10	(reserved)
11	Operator A interface active
12	Operator B interface active
13	(reserved)
14	DEVPARM or table format error
15	Incorrect message length
16	Operator made interface inactive
17	Line length exceeded
18	Warning line reached
19	End of journal form
20	Printer error (with counters 25-30)
21	(reserved)
22	Translate check
23	End of page
24	End of cut form
25	Left margin sensor/head calibration error
26	Print wire error
27	Power supply error
28	Timer or stop latch error
29	Forms motor error
30	Sensor error

Refer to the *4700 Subsystem Operating Procedures* for detailed descriptions of the statistical counters and their meanings.

Translation Tables

The 4720 translates data sent by your program for printing according to the tables shown here. You must select the table on the OUTBHDR and OUTSPEC configuration macros. You can also specify user-defined character sets on the OUTBHDR and CHARDEF configuration macros.

There are 22 feature character sets available for the 4720 printer:

Arabic
Belgian
Brazilian
Canadian
Danish/Norwegian
Finnish/Swedish
French
Greek
German/Austrian
Hebrew
International
Italian
Japanese
Katakana
Latin American
Portuguese
South African
Spanish
Turkish
UK English
United States
Yugoslavian

The following three tables define the characters for all of the character sets available. Figure 13-3 defines all characters available in all character sets except Arabic, Greek, Hebrew, and Katakana. Figure 13-22 on page 13-39 defines the characters and their locations for Arabic, Greek, and Hebrew. Figure 13-26 on page 13-43 defines the Katakana characters and their locations in the 4720.

You can use these figures to identify the characters for all available character sets and—for Arabic, Greek, Hebrew, International, and Katakana—you can also use these figures to change the translation tables for the 4720. If your system defines some other character set, use Figure 13-3 *only* to identify characters (using columns 2 and 3). You can find the character position in your character set by referring to the character set tables that follow Figure 13-3.

In Figure 13-3, the first column is the location of each character in the International character set. If this character set is defined for your 4720, use this column to locate characters or redefine positions in your translation table.

The second column is a graphic identifier (ID) of each character. This four-character ID is an IBM identifier you can use to find the character's translation table position in the appropriate character set table following Figure 13-3 .

The third column is the character name, in English.

Position	ID	Description
00	sp01	Space
01	nf17	One Eighth
02	la15	a Circumflex
03	la17	a Diaeresis
04	la13	a Grave
05	la11	a Acute
06	la19	a Tilde
07	la27	a Overcircle
08	lc41	c Cedilla
09	ln19	n Tilde
0A	sm06	Left Bracket
0B	sp11	Period
0C	sa03	Less Than
0D	sp06	Left Parenthesis
0E	sa01	Plus
0F	sp02	Exclamation Point
10	sm03	Ampersand
11	le11	e Acute
12	le15	e Circumflex
13	le17	e Diaeresis
14	le13	e Grave
15	li11	i Acute
16	li15	i Circumflex
17	li17	i Diaeresis
18	li13	i Grave
19	ls61	Sharp s
1A	sm08	Right Bracket
1B	sc03	Dollar Sign
1C	sm04	Asterisk
1D	sp07	Right Parenthesis
1E	sp14	Semicolon
1F	sd15	Circumflex Accent
20	sp10	Hyphen
21	sp12	Slash
22	la16	A Circumflex
23	la18	A Diaeresis
24	la14	A Grave
25	la12	A Acute
26	la20	A Tilde
27	la28	A Overcircle
28	lc42	C Cedilla Capital
29	ln20	N Tilde Capital
2A	sm65	Vertical Line Brok
2B	sp08	Comma
2C	sm02	Percent Sign
2D	sp09	Underline
2E	sa05	Greater Than
2F	sp15	Question Mark
30	lo61	o Slash Small
31	le12	E Acite Capital
32	le16	E Circumflex
33	le18	E Diaeresis Capital
34	le14	E Grave
35	li12	I Acute
36	li16	I Circumflex
37	li18	I Diaeresis
38	li14	I Grave
39	sd13	Grave Accent
3A	sp13	Colon
3B	sm01	Number Sign
3C	sm05	At Sign
3D	sp05	Apostrophe
3E	sa04	Equal Sign
3F	sp04	Quotation Marks
40	lo62	O Slash
41	la01	a
42	lb01	b
43	lc01	c
44	ld01	d
45	le01	e
46	lf01	f
47	lg01	g
48	lh01	h
49	li01	i
4A	sp17	left angle quotation marks
4B	sp18	right angle quotation marks
4C	ld63	eth Icelandic
4D	ly11	y acute
4E	lt63	Thorn Icelandic
4F	sa02	Plus or minus
50	sm19	Degree/Overcircle
51	lj01	j
52	lk01	k
53	ll01	l
54	lm01	m
55	ln01	n
56	lo01	o
57	lp01	p
58	lq01	q
59	lr01	r
5A	sm21	Ordinal, feminine
5B	sm20	Ordinal, masculine
5C	la51	ae Diphthong
5D	sd41	Cedilla accent
5E	la52	AE Diphthong
5F	sc01	International currency symbol
60	sm17	Micro Symbol
61	sd19	Tilde Accent
62	ls01	s
63	lt01	t
64	lu01	u
65	lv01	v
66	lw01	w
67	lx01	x
68	ly01	y
69	lz01	z

Figure 13-3 (Part 1 of 2). International Character ID Table

Position	ID	Description
6A	sp03	Exclamation, inverted
6B	sp16	Question, inverted
6C	ld62	Eth Icelandic
6D	ly12	Y Acute Capital
6E	lt64	Thorn Iceland
6F	sm53	trademark
70	sc04	Cent sign
71	sc02	Pound sign
72	sc05	Yen sign
73	sc06	Peseta sign
74	sc07	Florin, Guilder
75	sm24	Section symbol
76	sm25	Paragraph symbol
77	nf04	One-quarter
78	nf01	One-half
79	nf05	Three quarters
7A	sm66	Logical NOT
7B	sm13	Vertical Bar
7C	sm15	Overline
7D	sd17	Diaeresis
7E	sd11	Acute
7F	sm10	Double Underscore
80	sm11	Left Brace
81	la02	A
82	lb02	B
83	lc02	C
84	ld02	D
85	le02	E
86	lf02	F
87	lg02	G
88	lh02	H
89	li02	I
8A	nf19	Three-eighths
8B	lo15	o Circumflex
8C	lo17	o Diaeresis
8D	lo13	o Grave
8E	lo11	o Acute
8F	lo19	o Tilde
90	sm14	Right brace
91	lj02	J
92	lk02	K
93	ll02	L
94	lm02	M
95	ln02	N
96	lo02	O
97	lp02	P
98	lq02	Q
99	lr02	R
9A	li61	i dotless
9B	lu15	u Circumflex
9C	lu17	u Diaeresis
9D	lu13	u Grave
9E	lu11	u Acute
9F	ly17	y Diaeresis
A0	sm07	Backslash
A1	nf20	Five-eighths
A2	ls02	S
A3	lt02	T
A4	lu02	U
A5	lv02	V
A6	lw02	W
A7	lx02	X
A8	ly02	Y
A9	lz02	Z
AA	ns02	Two, superscript
AB	lo16	O Circumflex
AC	lo18	O Diaeresis
AD	lo14	O Grave
AE	lo12	O Acute
AF	lo20	O Tilde
B0	nd10	Zero
B1	nd01	One
B2	nd02	Two
B3	nd03	Three
B4	nd04	Four
B5	nd05	Five
B6	nd06	Six
B7	nd07	Seven
B8	nd08	Eight
B9	nd09	Nine
BA	ns03	Three superscript
BB	lu16	U Circumflex
BC	lu18	U Diaeresis
BD	lu14	U Grave
BE	lu12	U Acute
BF	ss99	Eight ones
C0	nf21	Seven-eighths
C1	lz22	Z Caron
C2	ls22	S Caron
C3	lc12	C Acute
C4	lc22	C Caron
C5	lz21	z Caron
C6	ls21	s Caron
C7	ld61	d Stroke
C8	lc11	c Acute
C9	lc21	c Caron
CA	li30	I Overdot
CB	ls42	S Cedilla
CC	lg24	G Breve
CD	ls41	s Cedilla
CE	lg23	g breve
CF	sa06	Division symbol
D0	sm16	Liter
D1	ln63	High Comma n
D2	ll63	l Middle Dot
D3	ll64	L Middle Dot

Figure 13-3 (Part 2 of 2). International Character ID Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	80 sm11	90 sm14	A0 sm07	B0 nd10
1	00 sp01	11 le11	21 sp12	31 le12	41 la01	51 lj01	61 sd19	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 lc41	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	0A sm06	1A sm08	2A sm65	3A sp13	4A sp17	5A sm21	6A sp03	7A sm66	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	1B sc03	2B sp08	3B sm01	4B sp18	5B sm20	6B sp16	7B sm13	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	3C sm05	4C ld63	5C la51	6C ld64	7C sm15	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	0F sp02	1F sd15	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-4. International Translation Table

LEFTMOST HEX CHARACTER (columns) / RIGHTMOST HEX CHARACTER (rows)

	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	11 le11	14 le13	08 lc41	B0 nd10
1	00 sp01	80 sm11	21 sp12	31 le12	41 la01	51 lj01	7D sd17	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	3C sm05	90 sm14	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	A0 sm07	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	0A sm06	1A sm08	9D lu13	3A sp13	4A sp17	5A sm21	6A sp03	7A sm66	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	1B sc03	2B sp08	3B sm01	4B sp18	5B sm20	6B sp16	7B sm13	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	04 la13	4C ld63	5C la51	6C ld64	7C sm15	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	8D lo13	2A sm65	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	0F sp02	1F sd15	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-5. Belgian Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER

	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	8F lo19	11 le11	A0 sm07	B0 nd10
1	00 sp01	90 sm14	21 sp12	0A sm06	41 la01	51 lj01	61 sd19	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	39 sd13	16 li15	3C sm05	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	2A sm65	18 li13	1A sm08	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	06 la19	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	31 le12	1B sc03	08 lc41	3A sp13	4A sp17	5A sm21	6A sp03	7A sm66	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	28 lc42	2B sp08	AF lo20	4B sp18	5B sm20	6B sp16	7B sm13	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	26 la20	4C ld63	5C la51	6C ld64	7C sm15	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	0F sp02	1F sd15	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	80 sm11	9F ly17	3B sm01	BF sv17

Figure 13-6. Brazilian Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	11 le11	14 le13	5D sd41	B0 nd10
1	00 sp01	80 sm11	21 sp12	31 le12	41 la01	51 lj01	60 sd17	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	0A sm06	90 sm14	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 lc41	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	04 la13	1B sc03	9D lu13	3A sp13	4A sp17	5A sm21	6A sp03	7A sm66	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	1B sc03	2B sp08	3B sm01	4B sp18	5B sm20	6B sp16	7B sm13	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	3C sm05	4C ld63	5C la51	6C ld64	7C sm15	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	A0 sm07	6D ly12	61 sd19	8D lo13	2A sm65	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	1A sm08	8E lo11	9E lu11	AE lo12	BE lu12
F	0F sp02	1F sd15	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-7. Canadian Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	2A sm65	3C sm05	50 sm19	60 sm17	70 sc04	5C la51	07 la27	A0 sm07	B0 nd10
1	00 sp01	11 le11	21 sp12	31 le12	41 la01	51 lj01	9C lu17	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	90 sm14	17 li17	1B sc03	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 lc41	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	3B sm01	5F sc01	30 lo61	3A sp13	4A sp17	5A sm21	6A sp03	7A sm66	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	27 la28	2B sp08	5E la52	4B sp18	5B sm20	6B sp16	7B sm13	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	40 lo62	4C ld63	7E sm11	6C ld64	7C sm15	8C lo17	61 sd19	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	0A sm06	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	0F sp02	1F sd15	2F sp15	3F sp04	4F sa02	1A sm08	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-8. Danish/Norwegian Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	03 la17	07 la27	31 le12	B0 nd10
1	00 sp01	39 sd13	21 sp12	A0 sm07	41 la01	51 lj01	9C lu17	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	80 sm11	13 le17	3B sm01	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	0A le02	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	90 sm14	17 li17	1B sc03	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 lc41	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	11 le11	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	75 sm24	5F sc01	8C lo17	3A sp13	4A sp17	5A sm21	6A sp03	7A sm66	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	27 la28	2B sp08	23 la18	4B sp18	5B sm20	6B sp16	7B sm13	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	AC lo18	4C ld63	5C la51	6C ld64	7C sm15	2A sm65	61 sd19	3C sm05	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	0F sp02	1F sd15	2F sp15	3F sp04	4F sa02	1A sm08	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-9. Finnish/Swedish Translation Table

LEFTMOST HEX CHARACTER (columns) / RIGHTMOST HEX CHARACTER (rows)

	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	0A sm06	39 sd13	70 sc04	11 le11	14 le13	08 lc41	B0 nd10
1	00 sp01	80 sm11	21 sp12	31 le12	41 la01	51 lj01	60 sd17	3B sm01	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	3C sm05	90 sm14	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	1A sm08	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	A0 sm07	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	60 sm17	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	50 sm19	75 sm24	9D lu13	3A sp13	4A sp17	5A sm21	6A sp03	7A sm66	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	1B sc03	2B sp08	71 sc02	4B sp18	5B sm20	6B sp16	7B sm13	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	04 la13	4C ld63	5C la51	6C ld64	7C sm15	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	61 sd19	8D lo13	2A sm65	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	0F sp02	1F sd15	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-10. French Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	03 la17	9C lu17	AC lo18	B0 nd10
1	00 sp01	11 le11	21 sp12	31 le12	41 la01	51 lj01	19 ls61	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	80 sm11	13 le17	0A sm06	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	3C sm05	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 lc41	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	61 sd19	29 ln20	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	23 la18	BC lu18	8C lo17	3A sp13	4A sp17	5A sm21	6A sp03	7A sm66	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	1B sc03	2B sp08	3B sm01	4B sp18	5B sm20	6B sp16	7B sm13	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	75 sm24	4C ld63	5C la51	6C ld64	7C sm15	2A sm65	90 sm14	A0 sm07	1A sm08
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	0F sp02	1F sd15	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-11. German/Austrian Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	0A sm06	60 sm17	70 sc04	04 la13	14 le13	08 lc41	B0 nd10
1	00 sp01	1A sm08	21 sp12	31 le12	41 la01	51 lj01	18 li13	3B sm01	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	80 sm11	90 sm14	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	3C sm05	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	A0 sm07	61 sd19	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	9D lu13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	50 sm19	11 le11	8D lo13	3A sp13	4A sp17	5A sm21	6A sp03	7A sm66	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	1B sc03	2B sp08	71 sc02	4B sp18	5B sm20	6B sp16	7B sm13	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	75 sm24	4C ld63	5C la51	6C ld64	7C sm15	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	2A sm65	39 sd13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	0F sp02	1F sd15	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-12. Italian Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	80 sm11	90 sm14	1B sc03	B0 nd10
1	00 sp01	11 le11	21 sp12	31 le12	41 la01	51 lj01	7C sm15	0A sm06	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	A0 sm07	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 lc41	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	71 sc02	0F sp02	2A sm65	3A sp13	4A sp17	5A sm21	6A sp03	1F sd15	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	72 sc05	2B sp08	3B sm01	4B sp18	5B sm20	6B sp16	1A sm08	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	3C sm05	4C ld63	5C la51	6C ld64	61 sd19	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	7B sm13	7A sm66	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-13. Japanese Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	80 sm11	90 sm14	A0 sm07	B0 nd10
1	00 sp01	11 le11	21 sp12	31 le12	41 la01	51 lj01	7D sd17	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 lc41	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	2A sm65	19 ls61	3B sm01	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	0A sm06	1A sm08	09 ln19	3A sp13	4A sp17	5A sm21	6A sp03	1F sd15	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	1B sc03	2B sp08	29 ln20	4B sp18	5B sm20	6B sp16	0F sp02	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	3C sm05	4C ld63	5C la51	6C ld64	7C sm15	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	61 sd19	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	7B sm13	7A sm66	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-14. Latin American Translation Table

LEFTMOST HEX CHARACTER (columns) / RIGHTMOST HEX CHARACTER (rows)

	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	06 la19	7E sd11	28 lc42	B0 nd10
1	00 sp01	11 le11	21 sp12	31 le12	41 la01	51 lj01	08 lc41	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	80 sm11	16 li15	3B sm01	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	61 sd19	18 li13	A0 sm07	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	0A sm06	1A sm08	8F lo19	3A sp13	4A sp17	5A sm21	6A sp03	7A sm66	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	1B sc03	2B sp08	26 la20	4B sp18	5B sm20	6B sp16	7B sm13	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	AF lo20	4C ld63	5C la51	6C ld64	7C sm15	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	90 sm14	8E lo11	9E lu11	AE lo12	BE lu12
F	0F sp02	1F sd15	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	2A sm65	9F ly17	3C sm05	BF sv17

Figure 13-15. Portuguese Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10			50 sm19						B0 nd10
1	00 sp01		21 sp12		41 la01	51 lj01		71 sc02	81 la02	91 lj02		B1 nd01
2					42 lb01	52 lk02	62 ls01		82 lb02	92 lk02	A2 ls02	B2 nd02
3					43 lc01	53 ll01	63 lt01		83 lc02	93 ll02	A3 lt02	B3 nd03
4					44 ld01	54 lm01	64 lu01		84 ld02	94 lm02	A4 lu02	B4 nd04
5					45 le01	55 ln01	65 lv01		85 le02	95 ln02	A5 lv02	B5 nd05
6					46 lf01	56 lo01	66 lw01		86 lf02	96 lo02	A6 lw02	B6 nd06
7					47 lg01	57 lp01	67 lx01		87 lg02	97 lp02	A7 lx02	B7 nd07
8					48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9				39 sd13	49 li01	59 lr01	69 lz01		89 li02	99 lr02	A9 lz02	B9 nd09
A				3A sp13							AA ns02	BA ns03
B	0B sp11	1B sc03	2B sp08									
C		1C sm04	2C sm02	3C sm05								
D	0D sp06	1D sp07	2D sp09	3D sp05				7D sd17				
E	0E sa01	1E sp14		3E sa04	CF sa06	D0 sm16	D1 ln63	7E sd11				
F	0F sp02	1F sd15	2F sp15	3F sp04								BF sv17

Figure 13-16. South African Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER

	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	80 sm11	90 sm14	A0 sm07	B0 nd10
1	00 sp01	11 le11	21 sp12	31 le12	41 la01	51 lj01	7D sd17	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	1B sc03	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 lc41	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	2A sm65	19 ls61	3B sm01	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	0A sm06	1A sm08	09 ln19	3A sp13	4A sp17	5A sm21	6A sp03	1F sd15	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	73 sc06	2B sp08	29 ln20	4B sp18	5B sm20	6B sp16	0F sp02	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	3C sm05	4C ld63	5C la51	6C ld64	7C sm15	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	61 sd19	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	7B sm13	7A sm66	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-17. Spanish Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10						08 lc41	CE lg23	9C lu17	B0 nd10
1			21 sp12		41 la01	51 lj01	61 lo17		81 la02	91 lj02	00 sp01	B1 nd01
2					42 lb01	52 lk02	62 ls01		82 lb02	92 lk02	A2 ls02	B2 nd02
3					43 lc01	53 ll01	63 lt01		83 lc02	93 ll02	A3 lt02	B3 nd03
4					44 ld01	54 lm01	64 lu01		84 ld02	94 lm02	A4 lu02	B4 nd04
5					45 le01	55 ln01	65 lv01		85 le02	95 ln02	A5 lv02	B5 nd05
6				36 li16	46 lf01	56 lo01	66 lw01		86 lf02	96 lo02	A6 lw02	B6 nd06
7				37 li18	47 lg01	57 lp01	67 lx01		87 lg02	97 lp02	A7 lx02	B7 nd07
8				38 li14	48 lh01	58 lq01	68 ly01		88 lh02	98 lq02	A8 ly02	B8 nd08
9				9A li61	49 li01	59 lr01	69 lz01		89 li02	99 lr02	A9 lz02	B9 nd09
A	28 lc42	CC lg24	CD ls41	3A sp13								
B	0B sp11	CA li30	2B sp08	AC lo18								
C	0C sa03	1C sm04	2C sm02	CB ls42								
D	0D sp06	1D sp07	2D sp09	3D sp05								
E	0E sa01	1E sp14	2E sa05	3E sa04								
F	0F sp02	1F sd15	2F sp15	BC lu18								BF sv17

Figure 13-18. Turkish Translation Table

RIGHTMOST HEX CHARACTER \ LEFTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	70 sc04	80 sm11	90 sm14	A0 sm07	B0 nd10
1	00 sp01	11 le11	21 sp12	31 le12	41 la01	51 lj01	7C sm15	0A sm06	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 lc41	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	1B sc03	0F sp02	2A sm65	3A sp13	4A sp17	5A sm21	6A sp03	1F sd15	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	71 sc02	2B sp08	3B sm01	4B sp18	5B sm20	6B sp16	1A sm08	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	3C sm05	4C ld63	5C la51	6C ld64	61 sd19	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	7B sm13	7A sm66	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-19. UK English Translation Table

RIGHTMOST HEX CHARACTER \ LEFTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10	30 lo61	40 lo62	50 sm19	60 sm17	1F sd15	80 sm11	90 sm14	A0 sm07	B0 nd10
1	00 sp01	11 le11	21 sp12	31 le12	41 la01	51 lj01	61 sd19	71 sc02	81 la02	91 lj02	00 sp01	B1 nd01
2	02 la15	12 le15	22 la16	32 le16	42 lb01	52 lk02	62 ls01	72 sc05	82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 la17	13 le17	23 la18	33 le18	43 lc01	53 ll01	63 lt01	73 sc06	83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 la13	14 le13	24 la14	34 le14	44 ld01	54 lm01	64 lu01	74 sc07	84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01	75 sm24	85 le02	95 ln02	A5 lv02	B5 nd05
6	06 la19	16 li15	26 la20	36 li16	46 lf01	56 lo01	66 lw01	76 sm25	86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 la27	17 li17	27 la28	37 li18	47 lg01	57 lp01	67 lx01	77 nf04	87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 lc41	18 li13	28 lc42	38 li14	48 lh01	58 lq01	68 ly01	78 nf01	88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ln19	19 ls61	29 ln20	39 sd13	49 li01	59 lr01	69 lz01	79 nf05	89 li02	99 lr02	A9 lz02	B9 nd09
A	70 sc04	0F sp02	2A sm65	3A sp13	4A sp17	5A sm21	6A sp03	0A sm06	20 sp10	9A li61	AA ns02	BA ns03
B	0B sp11	1B sc03	2B sp08	3B sm01	4B sp18	5B sm20	6B sp16	1A sm08	8B lo15	9B lu15	AB lo16	BB lu16
C	0C sa03	1C sm04	2C sm02	3C sm05	4C ld63	5C la51	6C ld64	7C sm15	8C lo17	9C lu17	AC lo18	BC lu18
D	0D sp06	1D sp07	2D sp09	3D sp05	4D ly11	5D sd41	6D ly12	7D sd17	8D lo13	9D lu13	AD lo14	BD lu14
E	0E sa01	1E sp14	2E sa05	3E sa04	4E lt63	5E la52	6E lt64	7E sd11	8E lo11	9E lu11	AE lo12	BE lu12
F	7B sm13	7A sm66	2F sp15	3F sp04	4F sa02	5F sc01	6F sm53	7F sm10	8F lo19	9F ly17	AF lo20	BF sv17

Figure 13-20. United States Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10						C6 ls21	C8 lc11	6C ld62	B0 nd10
1	00 sp01		21 sp12		41 la01	51 lj01	61 lc21		81 la02	91 lj02	00 sp01	B1 nd01
2					42 lb01	52 lk02	62 ls01		82 lb02	92 lk02	A2 ls02	B2 nd02
3					43 lc01	53 ll01	63 lt01		83 lc02	93 ll02	A3 lt02	B3 nd03
4					44 ld01	54 lm01	64 lu01		84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 la11	15 li11	25 la12	35 li12	45 le01	55 ln01	65 lv01		85 le02	95 ln02	A5 lv02	B5 nd05
6					46 lf01	56 lo01	66 lw01		86 lf02	96 lo02	A6 lw02	B6 nd06
7					47 lg01	57 lp01	67 lx01		87 lg02	97 lp02	A7 lx02	B7 nd07
8					48 lh01	58 lq01	68 ly01		88 lh02	98 lq02	A8 ly02	B8 nd08
9				C5 lz21	49 li01	59 lr01	69 lz01		89 li02	99 lr02	A9 lz02	B9 nd09
A	C2 ls22	C3 lc12	C7 ld61	3A sp13					20 sp10			
B	0B sp11	1B sc03	2B sp08	3B sm01								
C	0C sa03	1C sm04	2C sm02	C1 lz22								
D	0D sp06	1D sp07	2D sp09	3D sp05								
E	0E sa01	1E sp14	2E sa05	3E sa04								
F	0F sp02	C4 lc22	2F sp15	3F sp04								BF sv17

Figure 13-21. Yugoslavian Translation Table

Position	ID	Description	Position	ID	Description
00	sp01	Space	31	ht01	Taw
01	hx33	Alef	32	at03	Teh (small)
02	hb01	Bel	33	at05	Theh (small)
03	hg01	Gimel	34	ag01	Jeem (small)
04	hd01	Dalet	35	ah01	Hah (small)
05	hh01	He	36	ah03	Khah (small)
06	hw01	Waw	37	ag02	Jeem
07	hz01	Zayin	38	ah02	Hah
08	hh45	Het	39	sd13	Grave accent
09	ht45	Tet	3A	sp13	Colon
0A	aa03	Aleph (Con)	3B	sm01	Number sign
0B	sp11	Period	3C	sm05	At sign
0C	sa03	Less than	3D	sp05	Apostrophe
0D	sp06	Left parenthesis	3E	sa04	Equal sign
0E	sa01	Plus	3F	sp04	Quotation marks
0F	sp02	Exclamation point	40	ah04	Khah
10	sm03	Ampersand	41	la01	a
11	hy01	Yod	42	lb01	b
12	hk61	Kaf (final form)	43	lc01	c
13	hk01	Kaf	44	ld01	d
14	hl01	Lamed	45	le01	e
15	hm61	Mem (final form)	46	lf01	f
16	hm01	Mem	47	lg01	g
17	hn61	Nun (final form)	48	lh01	h
18	hn01	Nun	49	li01	i
19	hs01	Samech	4A	ad02	Dal
1A	aa12	Hamza on Aleph (NC)	4B	ad04	Thal
1B	sc03	Dollar Sign	4C	ar02	Reh
1C	sm04	Asterisk	4D	az02	Zain
1D	sp07	Right Parenthesis	4E	as01	Seen (small)
1E	sp14	Semicolon	4F	as03	Sheen (small)
1F	ae03	Ayn	50	as05	Sad (small)
20	sp10	Hyphen	51	lj01	j
21	sp12	Slash	52	lk01	k
22	hx35	Ayin	53	ll01	l
23	hp61	Pe (final form)	54	lm01	m
24	hp01	Pe	55	ln01	n
25	hs61	Sade (final form)	56	lo01	o
26	hs45	Sade	57	lp01	p
27	hq01	Qof	58	lq01	q
28	hr01	Resh	59	lr01	r
29	hs21	Shin	5A	ad05	Dud (small)
2A	sm65	Vertical broken line	5B	at08	Tah
2B	sp08	Comma	5C	az04	Zah
2C	sm02	Percent sign	5D	sd41	Cedilla
2D	sp09	Underline	5E	ae03	Ayn (small)
2E	sa05	Greater than	5F	ag05	Ghayn (small)
2F	sp15	Question mark	60	ae06	Ayn
30	ab01	Beh (small)	61	ag08	Ghayn

Figure 13-22 (Part 1 of 2). Arabic/Greek/Hebrew Character ID Table

Position	ID	Description
62	ls01	s
63	lt01	t
64	lu01	u
65	lv01	v
66	lw01	w
67	lx01	x
68	ly01	y
69	lz01	z
6A	af01	Feh (small)
6B	aq01	Qaf (small)
6C	af02	Feh
6D	ak01	Caf (small)
6E	ak02	Caf
6F	al01	Lam (small)
70	sc04	Cent sign
71	al02	Lam
72	am01	Meem (small)
73	am02	Meem
74	an01	Noon (small)
75	an02	Noon
76	ah07	Heh (small)
77	ah08	Heh
78	aw04	Waw
79	al57	Lamaleph (Con)
7A	sm66	Logical not
7B	sm13	Vertical bar
7C	ay03	Yeh small
7D	aa03	Aleph (Con)
7E	aa04	Aleph (N/Con)
7F	ay02	Yeh Hamza
80	sm11	Left brace
81	la02	A
82	lb02	B
83	lc02	C
84	ld02	D
85	le02	E
86	lf02	F
87	lg02	G
88	lh02	H
89	li02	I
8A	ac03	Kasseh (Tail)
8B	ac01	Tatweel (connector)
8C	sm04a	Asterisk (Arabic)
8D	sp08a	Comma inverted
8E	sp12a	Slash (Arabic)
8F	nd10a	Sefr (zero)
90	sm14	Right Brace
91	lj02	J
92	lk02	K
93	ll02	L
94	lm02	M
95	ln02	N
96	lo02	O
97	lp02	P
98	lq02	Q
99	lr02	R
9A	nd01a	Wahad (One)
9B	nd02a	Ethnan (Two)
9C	nd03a	Thalathah (Three)
9D	nd04a	Arba-Ah (Four)
9E	nd05a	Khamsah (Five)
9F	nd06a	Settah (Six)
A0	sm07	Backslash
A1	nd07a	Sab-ah (Seven)
A2	ls02	S
A3	lt02	T
A4	lu02	U
A5	lv02	V
A6	lw02	W
A7	lx02	X
A8	ly02	Y
A9	lz02	Z
AA	nd08a	Thamaniah (Eight)
AB	nd09a	Tes-Ah (Nine)
AC	sp14a	Semicolon (Arabic)
AD	sp15a	Quest Mark (Arabic)
AE	gg02	Gamma (capital)
AF	gd02	Delta (capital)
B0	nd10	Zero
B1	nd01	One
B2	nd02	Two
B3	nd03	Three
B4	nd04	Four
B5	nd05	Five
B6	nd06	Six
B7	nd07	Seven
B8	nd08	Eight
B9	nd09	Nine
BA	gt62	Theta (capital)
BB	gl02	Lambda (capital)
BC	gx02	Xi (capital)
BD	gp02	Pi (capital)
BE	gs02	Sigma (capital)
BF	sv17	Error, Check Prot
C0	gu02	Upsilon (capital)
C1	gf02	Phi (capital)
C2	gp62	Psi (capital)
C3	go32	Omega (capital)

Figure 13-22 (Part 2 of 2). Arabic/Greek/Hebrew Character ID Table

RIGHTMOST HEX CHARACTER \ LEFTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	8B ac01	8F nd10	8A ac03			4E as01	70 sc04	3F sp04	8D sp08	B0 nd10
1			21 sp12	9A nd01	1F ae03			4F as03	81 la02	91 lj02		B1 nd01
2	0A aa03			9B nd02	1A aa12			50 as05	82 lb02	92 lk02	A2 ls02	B2 nd02
3				9C nd03				5A ad05	82 lc02	93 ll02	A3 lt02	B3 nd03
4				9D nd04		30 ab01	5B at08		84 ld02	94 lm02	A4 lu02	B4 nd04
5				9E nd05			5C az04		85 le02	95 ln02	A5 lv02	B5 nd05
6				9F nd06		32 at03	60 ae06	5E ae03	86 lf02	96 lo02	A6 lw02	B6 nd06
7				A1 nd07		33 at05	61 ag08	5F ag05	87 lg02	97 lp02	A7 lx02	B7 nd07
8				AA nd08		34 ag01		6A af01	88 lh02	98 lq02	A8 ly02	B8 nd08
9				AB nd09					89 li02	99 lr02	A9 lz02	B9 nd09
A	AC sp14	AD sp15	2A sm65	3A sp13	38 ah02	35 ah01	6C af02	6B aq01				
B	0B sp11	1B sc03	2B sp08	3B sm01	40 ah04	36 ah03	6E ak02	6D ak01			77 ah08	76 ah07
C	0C sa03	8C sm04	2C sm02	3C sm05	4A ad02			79 al57			78 aw04	
D	0D sp06	1D sp07	2D sp09	3D sp05	4B ad04		71 al02	6F al01			7F ay02	
E	0E sa01	1E sp14	2E sa05	3E sa04	4C ar02		73 am02	72 am01	0A aa03		7E aa04	7C ay03
F	0F sp02	7A sm66	2F sp15		4D az02		75 an02	74 an01	7B sm13			BF sv17

Figure 13-23. Arabic Translation Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10						80 sm11	90 sm14	A0 sm07	B0 nd10
1	81 la02	92 lk02	21 sp12		41 la01	51 lj01			81 la02	91 lj02		B1 nd01
2	82 lb02	BB gl02	A3 lt02		42 lb01	52 lk02	62 ls01		82 lb02	92 lk02	A2 ls02	B2 nd02
3	AE gg02	94 lm02	A8 ly02		43 lc01	53 ll01	63 lt01		83 lc02	93 ll02	A3 lt02	B3 nd03
4	AF gd02	95 ln02	C1 gf02		44 ld01	54 lm01	64 lu01		84 ld02	94 lm02	A4 lu02	B4 nd04
5	85 le02	BC gx02	A7 lx02		45 le01	55 ln01	65 lv01		85 le02	95 ln02	A5 lv02	B5 nd05
6	A9 lz02	96 lo02	C2 gp62		46 lf01	56 lo01	66 lw01		86 lf02	96 lo02	A6 lw02	B6 nd06
7	88 lh02	BD gp02	C3 go32		47 lg01	57 lp01	67 lx01		87 lg02	97 lp02	A7 lx02	B7 nd07
8	BA gt62	97 lp02			48 lh01	58 lq01	68 ly01		88 lh02	98 lq02	A8 ly02	B8 nd08
9	89 li02	BE gs02			49 li01	59 lr01	69 lz01		89 li02	99 lr02	A9 lz02	B9 nd09
A	70 sc04	0F sp02	2A sm65	3A sp13								
B	0B sp11	1B sc03	2B sp08	3B sm01								
C	0C sa03	1C sm04	2C sm02	3C sm05								
D	0D sp06	1D sp07	2D sp09	3D sp05								
E	0E sa01	1E sp14	2E sa05	3E sa04								
F	7B sm13	7A sm66	2F sp15	3F sp04								BF sv17

Figure 13-24. Greek Translation Table

RIGHTMOST HEX CHARACTER \ LEFTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10						80 sm11	90 sm14	A0 sm07	B0 nd10
1	01 hx33	11 hy01	21 sp12	31 ht01	41 la01	51 lj01	5D sd41		81 la02	91 lj02		B1 nd01
2	02 hb01	12 hk61	22 hx35		42 lb01	52 lk02	62 ls01		82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 hg01	13 hk01	23 hp61		43 lc01	53 ll01	63 lt01		83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 hd01	14 hl01	24 hp01		44 ld01	54 lm01	64 lu01		84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 hh01	15 hm61	25 hs61		45 le01	55 ln01	65 lv01		85 le02	95 ln02	A5 lv02	B5 nd05
6	06 hw01	16 hm01	26 hs45		46 lf01	56 lo01	66 lw01		86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 hz01	17 hn61	27 hq01		47 lg01	57 lp01	67 lx01		87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 hh45	18 hn01	28 hr01		48 lh01	58 lq01	68 ly01		88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ht45	19 hs01	29 hs21	39 sd13	49 li01	59 lr01	69 lz01		89 li02	99 lr02	A9 lz02	B9 nd09
A	0A sc04	0F sp02	2A sm65	3A sp13								
B	0B sp11	1B sc03	2B sp08	3B sm01								
C	0C sa03	1C sm04	2C sm02	3C sm05								
D	0D sp06	1D sp07	2D sp09	3D sp05								
E	0E sa01	1E sp14	2E sa05	3E sa04								
F	7B sm13	7A sm66	2F sp15	3F sp04								BF sv17

Figure 13-25. Hebrew Translation Table

Position	ID	Description
00	sp01	Space
01	jq70	Kana Period
02	jq71	Open Quote
03	jq72	Close Quote
04	jq73	Kana Comma
05	jq74	Center Period
06	jw50	Wo
07	ja01	a
08	ji01	i
09	ju01	u
0A		
0B	sp11	Period
0C	sa03	Less than
0D	sp06	Left Parenthesis
0E	sa01	Plus
0F	sp02	Exclamation Point
10	sm03	Ampersand
11	je01	e
12	jo01	o
13	jy11	ya
14	jy31	yu
15	jy51	yo
16	jt31	tsu
17		
18	jx70	Cho-On
19		
1A		
1B	sc03	Dollar Sign
1C	sm04	Asterisk
1D	sp07	Right Parenthesis
1E	sp14	Semicolon
1F		
20	sp10	Hyphen
21	sp12	Slash
22		
23		
24		
25		
26		
27		
28		
29		
2A		
2B	sp08	Comma
2C	sm02	Percent Sign
2D	sp09	Underline
2E	sa05	Greater Than
2F	sp15	Question Mark

Position	ID	Description
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
3A	sp13	Colon
3B	sm01	Number Sign
3C	sm05	At Sign
3D	sp05	Apostrophe
3E	sa04	Equal Sign
3F	sp04	Quotation Marks
40		
41	ja00	A
42	ji00	I
43	ju00	U
44	je00	E
45	jo00	O
46	jk10	KA
47	jk20	KI
48	jk30	KU
49	jk40	KE
4A	jk50	KO
4B		
4C	js10	SA
4D	js20	SHI
4E	js30	SU
4F	js40	SE
50	js50	SO
51	jt10	TA
52	jt20	CHI
53	jt30	TSU
54	jt40	TE
55	jt50	TO
56	jn10	NA
57	jn20	NI
58	jn30	NU
59	jn40	NE
5A	jn50	NO
5B		
5C		
5D	jh10	HA
5E	jh20	HI
5F	jh30	FU (HU)

Figure 13-26 (Part 1 of 2). Katakana Character ID Table

Position	ID	Description
60		
61		
62	jh40	HE
63	jh50	HO
64	jm10	MA
65	jm20	MI
66	jm30	MU
67	jm40	ME
68	jm50	MO
69	jy10	YA
6A	jy30	YU
6B	jr40	RE
6C	jy50	YO
6D	jr10	RA
6E	jr20	RI
6F	jr30	RU
70	jr50	RO
71	sc02	Pound sign
72	sc05	Yen sign
73		
74		
75		
76		
77		
78		
79		
7A	sm66	Logical NOT
7B	sm13	Vertical bar
7C	jw10	WA
7D	jn00	UN
7E	jx71	Kana sonant
7F	jf72	Semi sonant
80	sm11	Left brace
81	la02	A
82	lb02	B
83	lc02	C
84	ld02	D
85	le02	E
86	lf02	F
87	lg02	G
88	lh02	H
89	li02	I
8A		
8B		
8C		
8D		
8E		
8F		
90	sm14	Right brace
91	lj02	J
92	lk02	K
93	ll02	L
94	lm02	M
95	ln02	N
96	lo02	O
97	klp02	P
98	lq02	Q
99	lr02	R
9A		
9B		
9C		
9D		
9E		
9F		
A0		
A1		
A2	ls02	S
A3	lt02	T
A4	lu02	U
A5	lv02	V
A6	lw02	W
A7	lx02	X
A8	ly02	Y
A9	lz02	Z
AA		
AB		
AC		
AD		
AE		
AF		
B0	nd10	Zero
B1	nd01	One
B2	nd02	Two
B3	nd03	Three
B4	nd04	Four
B5	nd05	Five
B6	nd06	Six
B7	nd07	Seven
B8	nd08	Eight
B9	nd09	Nine
BA		
BB		
BC		
BD		
BE		
BF	sv17	Error, Check Prot

Figure 13-26 (Part 2 of 2). Katakana Character ID Table

LEFTMOST HEX CHARACTER

RIGHTMOST HEX CHARACTER	4	5	6	7	8	9	A	B	C	D	E	F
0	00 sp01	10 sm03	20 sp10			50 js50					1B sc03	B0 nd10
1	01 jq70	11 je01	21 sp12		41 ja00	51 jt10			81 la02	91 lj02	A1 sp01	B1 nd01
2	02 jq71	12 jo01			42 ji00	52 jt20	62 jh40		82 lb02	92 lk02	A2 ls02	B2 nd02
3	03 jq72	13 jy11			43 ju00	53 jt30	63 jh50		83 lc02	93 ll02	A3 lt02	B3 nd03
4	04 jq73	14 jy31			44 je00	54 jt40	64 jm10		84 ld02	94 lm02	A4 lu02	B4 nd04
5	05 jq74	15 jy51			45 jo00	55 jt50	65 jm20		85 le02	95 ln02	A5 lv02	B5 nd05
6	06 jw50	16 jt31			46 jk10	56 jn10	66 jm30		86 lf02	96 lo02	A6 lw02	B6 nd06
7	07 ja01				47 jk20	57 jn20	67 jm40		87 lg02	97 lp02	A7 lx02	B7 nd07
8	08 ji01	18 jx70			48 jk30	58 jn30	68 jm50		88 lh02	98 lq02	A8 ly02	B8 nd08
9	09 ju01				49 jk40	59 jn40	69 jy10		89 li02	99 lr02	A9 lz02	B9 nd09
A	71 sc02	0F sp02		3A sp13	4A jk50	5A jn50	6A jy30	6B jr40				
B	0B sp11	72 sc05	2B sp08	3B sm01				70 jr50				
C	0C sa03	1C sm04	2C sm02	3C sm05	4C js10		6C jy50	7C jw10				
D	0D sp06	1D sp07	2D sp09	3D sp05	4D js20	5D jh10	6D jr10	7D jn00				
E	0E sa01	1E sp14	2E sa05	3E sa04	4E js30	5E jh20	6E jr20	7E jx71				
F	7B sm13	7A sm66	2F sp15	3F sp04	4F js40	5F jh30	6F jr30	7F jx72				BF sv17

Figure 13-27. Katakana Translation Table

Chapter 14. IBM 3610 Document Printer

The IBM 3610 Document Printer is used by tellers and administrators to supplement a keyboard display with printed output. The 3610 printer, shown in Figure 14-1 and Figure 14-2 , is available in five models:

- Model 1 prints cut forms only
- Model 2 prints cut forms and a journal
- Model 3 prints cut forms and administrative reports on continuous forms
- Model 4 prints cut forms and a journal
- Model 5 prints cut forms and a journal

Two additional models that provide the same capability as Models 2 and 3 but have 128-character print sets are available for Japan only. These Japanese models are, respectively, Models 12 and 13.

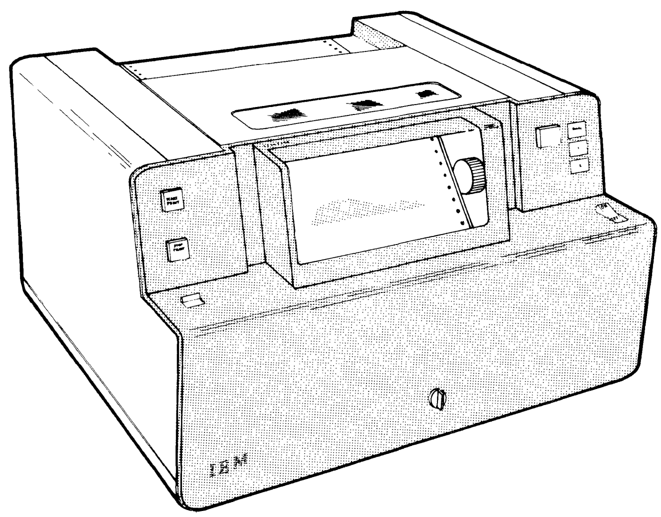


Figure 14-1. IBM 3610 Document Printer Model 1 (Models 2 and 3 are similar)

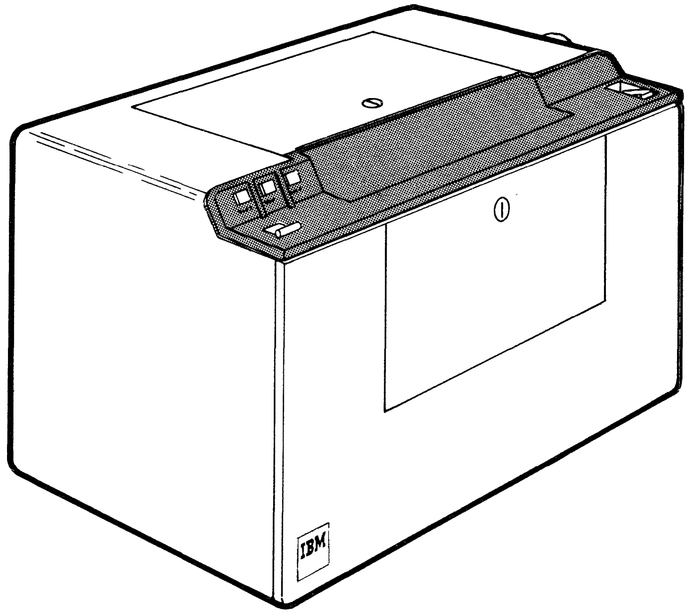


Figure 14-2. IBM 3610 Document Printer Model 4 (Model 5 is similar)

The basic 3610 prints 10 characters per inch, at either 5 or 6 lines per inch. The character set size and print rate depend on the model and feature. All five models can print one or more lines of up to 80 characters each on a cut form that has up to four parts (an original and three copies). The Model 3 can also print on a continuous form that is pin-fed into the printer, and on a cut form at the same time. Models 2, 4, and 5 can also print on a journal, which is a roll of paper (an original with or without one copy) that is friction-fed into the printer. By using impact paper for the journal, you can print on the journal and a cut form at the same time.

Warning: To avoid damaging the print wheel of the document printing mechanism, do not print off the edge of the document and do not print over holes, turned-over edges, folds (including the normal fold), stamps, staples, paper clips, or severely warped pages.

Models 4 and 5 have sets of 48 and 96 characters, respectively, printed at a rate of up to 30 characters per second. They also have a take-up roll, under a locked cover, for the second part of a two-part journal after it is printed. (The original part must leave the printer past the tear bar.)

Three of the special features available are:

- A set of 64 characters printed at 15 characters per second (Models 1, 2, and 3 only)
- A set of 96 characters printed at up to 30 characters per second (Models 1, 2, and 3 only)
- A shared printer option, with operator-identifier keys, that allows two logical work stations to share one 3610 printer.

An additional special feature, available on the Model 2, allows the printer to accumulate one or both parts of a two-part journal on the same take-up roll. As with Models 4 and 5, a locked cover limits access to the journal.

The character set for Models 1, 2, 3, and 5 can be Belgian, Brazilian/Portuguese, Danish/Norwegian, American English, United Kingdom English, Finnish/Swedish, French, German, Italian, Spanish, or Spanish-speaking. Power can be 60 hertz (at 100 or 115 volts) or 50 hertz (at 100, 110, 123.5, 220, or 235 volts).

The Model 4 has an American English character set and requires 60 hertz at 115 volts.

The 3610 printer is specified by the DEV3610 configuration macro instruction with one physical device address (loop number and terminal address). When defined in the STATION configuration macro instruction or a data transmission instruction, the printer is selected by a single logical device address.

Controls and Indicators

The power on/off switch, the READY indicator, and indicators 1 and 2 are the same as for the 3612 printer; see Chapter 10.

Start Print Key

This key activates the printer, and must be pressed before each printing operation on a cut form (unless autostart in cut-form mode has been selected in the DEV3610 macro). Pressing this key closes the platen and turns on the drive motor, which starts the print wheel rotating. If the controller does not send data within 20 seconds, the platen opens and the drive motor stops. A printer with the shared-terminal feature has two START PRINT keys (one for each station) on opposite sides of the printer.

Stop Print Key

This key stops any printing that is in progress: it opens the platen and stops the drive motor, which stops the print wheel. Press a START PRINT key to resume printing.

Document Printing

Document printing and character sets for the 3610 Document Printer are the same as for the document printing mechanism of the 3612 printer. The 3610 printer operates in cut-forms or continuous-forms mode and can be shared between two stations. For discussions of printing operations and character sets, see the 3612 in Chapter 10.

Note: For 3610 Models 4 and 5, the journal width is fixed at 216 mm (8.5 in.).

Translating Data and Handling Control Characters

All data and control characters provided by the application program (using WRTI or LWRITE) are translated by the controller using the translation table shown later in this chapter. The translated data is written to the printer from a buffer in the controller. If a control character is detected during translation, translation is suspended until the requested operation is performed. However, the controller does not suspend translation for control characters that result in no operation or in horizontal movement in the same line. If no status results, translation resumes. The controller reports any status to the application program.

Note that the end of a print line occurs at column 80 regardless of the document being printed. If this value is not appropriate for a given document, the controller application program must include new-line control characters where needed.

Variable Parameters

A set of variable parameters describes the physical characteristics of the documents to be printed. The parameters are specified during controller configuration. They can be changed during execution of the controller application program by the DEVPARM instruction.

The variable parameters for the 3610 printer are the same as those for the document printing mechanism of the 3612 printer; see Chapter 16, "IBM 3612 Passbook and Document Printer" on page 16-1.

Controller Configuration for The 3610

Before the 3610 can be operated by a controller program, controller configuration macro instructions must be written to define the physical and logical configuration of all devices attached to the controller. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. Refer to Volume 6 for macro coding information and descriptions of the DEV3610 and other configuration macros.

Terminal Control Characters

The control characters for the 3610 printer (together with their functions and explanations) are the same as for the 3612; see Chapter 16, "IBM 3612 Passbook and Document Printer" on page 16-1.

Statistical Counters

The statistical counters for the 3610 are the same as for the 3612; see Chapter 16, "IBM 3612 Passbook and Document Printer."

Status Bits

The status bits and their explanations for the 3610 printer are listed in Appendix D, "Terminal and Device Status Codes" on page D-1.

Output Translation Tables

The tables below are the standard output translation tables for the 3610. Characters that do not have corresponding hexadecimal values in these tables are nonstandard. To refer to these characters, you must either give each of them a unique hexadecimal value when it is specified in the OUTRTBL macro, or use a transparent write operation.

For 3610 Model 4, 48-Character Set

Position	*Hex Value*	*Character*	*Position*	*Hex Value*	*Character*
0			25	4B	.
1	4C	⌑	26	D7	P
2	E9	Z	27	F2	2
3	D2	K	28	C2	B
4	E5	V	29	F4	4
5	E8	Y	30	D3	L
6	5B	$	31	F6	6
7	7B	#	32	C6	F
8	C7	G	33	F8	8
9	7C	@	34	D5	N
10	C3	C	35	F9	9
11	C1	A	36	D9	R
12	C4	D	37	C5	E
13	F7	7	38	E3	T
14	E2	S	39	6B	,
15	F5	5	40	C8	H
16	D6	O	41	60	-
17	F3	3	42	E7	X
18	C9	I	43	6C	%
19	F1	1	44	4E	+
20	E4	U	45	D1	J
21	5C	*	46	D8	Q
22	E6	W	47	61	/
23	F0	0			
24	D4	M			

For 3610, 64-Character Set

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

Note: With the standard translation table, referring to position 128 results in a space.

For 3610, 96-Character Set

The first 64 characters of the 96-character set table for the 3610/3611/3612 are identical to the 64-character set. This table shows the final 32 characters beginning with relative position 64.

Position	Hex Value	Character	Position	Hex Value	Character
64	A2	s	80	93	l
65	83	c	81	88	h
66	96	o	82	84	d
67	97	p	83	A6	w
68	89	i	84	A5	v
69	94	m	85	92	k
70	A3	t	86	A7	x
71	A8	y	87	79	`
72	85	e	88	A1	~
73	82	b	89	7F	"
74	81	a	90	98	q
75	86	f	91	91	j
76	95	n	92	A9	ƶ
77	87	g	93	C0	{
78	99	r	94	D0	}
79	A4	u	95	6A	¦

Note: With the standard translation table, referring to position 128 results in a space.

For 3610, 128-Character Katakana Set

Position	*Hex Value*	*Character*	
0		⊗	
1	98	ヌ	Nu
2	A2	ヘ	He
3	A6	ム	Mu
4	99	ネ	Ne
5	AA	ユ	Yu
6	A9	ヤ	Ya
7	A7	メ	Me
8	BB	ロ	Ro
9	A3	ホ	Ho
10	9F	フ	Fu
11	9E	ヒ	Hi
12	A5	ミ	Mi
13	84	エ	E
14	AC	ヨ	Yo
15	8F	セ	Se
16	89	ケ	Ke
17	92	チ	Chi
18	9D	ハ	Ha
19	90	ソ	So
20	95	ト	To
21	93	ツ	Tsu
22	9A	ノ	No
23	AE	リ	Ri
24	94	テ	Te
25	BA	レ	Re
26	91	タ	Ta
27	46	ヲ	Wo
28	8D	シ	Shi
29	5B	¥	Yen
30	86	カ	Ka
31	60	−	minus
32	82	イ	I
33	85	オ	O
34	BE	゛	KanaSonant
35	8C	サ	Sa
36	5C	*	Ast'sk
37	8E	ス	Su
38	4B	•	Period
39	AD	ラ	Ra
40	F0	0	Zero
41	81	ア	A
42	F1	1	One
43	A8	モ	Mo
44	F2	2	
45	56	ッ	Sm. Tsu
46	F3	3	
47	A4	マ	Ma
48	F4	4	
49	87	キ	Ki
50	F5	5	
51	AF	ル	Ru

Position	*Hex Value*	*Character*	
52	F6	6	
53	88	ク	Ku
54	F7	7	
55	96	ナ	Na
56	F8	8	
57	BC	ワ	Wa
58	F9	9	
59	83	ウ	U
60	6B	,	comma
61	BD	ン	N
62	97	ニ	Ni
63	8A	コ	Ko
64	54	ュ	Sm. Yu
65	D3	L	
66	49	ゥ	Sm. u
67	C6	F	
68	55	ョ	Sm. Yo
69	C9	I	
70	53	ャ	Sm. Ya
71	D5	N	
72	BF	゜	Sem.Son
73	E2	S	
74	58	ー	Choon
75	D6	O	
76	42	「	Op.Quot
77	D9	R	
78	43	」	Cl.Quot
79	64	D	
80	C2	B	
81	E3	T	
82	D7	P	
83	C3	C	
84	D4	M	
85	C1	A	
86	E4	U	
87	C5	E	
88	C7	G	
89	C8	H	
90	E8	Y	
91	E7	X	
92	61	/	
93	E5	V	
94	7E	=	
95	4E	+	
96	D2	K	
97	E0	$	
98	5F	¬	
99	41	•	Kana Period
100	44	,	Kana Comma
101	5A	!	
102	7F	"	
103	A1	‾	Overline

For 3610, 128-Character Katakana Set (continued)

Position	Hex Value	Character	
104	7D	'	
105	6D	_	Underscore
106	6E	>	
107	4C	<	
108	5E	;	
109	4F	|	
110	6C	%	
111	E6	W	
112	D1	J	
113	7B	#	
114	50	&	
115	7A	:	
116	45	.	Center Per'd
117	4A	£	
118	7C	@	
119	5D	)	
120	4D	(	
121	D6	Q	
122	E9	Z	
123	6F	?	
124	52	ォ	Small o
125	48	ィ	Small i
126	47	ァ	Small a
127	51	ェ	Small e

Note: With the standard translation table, referring to position 128 results in a space.

For 3610 Model 5, 96-Character Katakana Set

Position	Hex Value	Character	
0			
1	D1	J	
2	E9	Z	
3	E5	V	
4	D2	K	
5	D5	N	
6	C1	A	
7	D3	L	
8	C5	E	
9	C8	H	
10	5C	゛	Kana Sonant
11	E4	U	
12	6B	,	
13	E2	S	
14	60	–	
15	C7	G	
16	F7	7	
17	E8	Y	
18	F6	6	
19	E6	W	
20	F2	2	
21	D4	M	
22	F1	1	
23	6C	%	
24	F0	0	
25	5B	¥	Yen
26	F5	5	
27	D7	P	
28	F3	3	
29	C2	B	
30	F9	9	
31	C4	D	
32	F4	4	
33	61	/	
34	F8	8	
35	D9	R	
36	4B	.	
37	C6	F	
38	E3	T	
39	D6	O	
40	C3	C	
41	D8	Q	
42	E7	X	
43	98	ヌ	Nu
44	C9	I	
45	A7	メ	Me
46	5D	)	
47	4D	(	
48	84	エ	E
49	A2	ヘ	He
50	A6	ム	Mu
51	BB	ロ	Ro
52	99	ネ	Ne
53	AC	ヨ	Yo
54	A3	ホ	Ho
55	AF	ル	Ru
56	87	キ	Ki
57	96	ナ	Na
58	A4	マ	Ma
59	97	ニ	Ni
60	A8	モ	Mo
61	83	ウ	U
62	8C	サ	Sa
63	95	ト	To

For 3610 Model 5, 96-Character Katakana Set (continued)

Position	*Hex Value*	*Character*	
64	85	オ	O
65	94	テ	Te
66	46	ヲ	Wo
67	8D	シ	Shi
68	93	ツ	Tsu
69	86	カ	Ka
70	90	ソ	So
71	BE	*	
72	9D	ハ	Ha
73	82	イ	I
74	92	チ	Chi
75	91	タ	Ta
76	AE	リ	Ri
77	9A	ノ	No
78	BA	レ	Re
79	BD	ン	Un
80	8E	ス	Su
81	BC	ワ	Wa
82	AD	ラ	Ra
83	8A	コ	Ko
84	81	ア	A
85	88	ク	Ku
86	8F	セ	Se
87	89	ケ	Ke
88	9E	ヒ	Hi
89	9F	フ	Fu
90	AA	ユ	Yu
91	A9	ヤ	Ya
92	A5	ミ	Mi
93	42	「	Open Quote
94	43	」	Close Quote
95	BF	°	Semi Sonant

Note: With the standard translation table, referring to position 128 results in a space.

Chapter 15. IBM 3611 Passbook Printer

The IBM 3611 Passbook Printer is used with a keyboard display to form a passbook-handling work station. The 3611 printer (Figure 15-1) can print passbooks or a limited class of cut forms. Model 1 prints passbooks or cut forms that are of a fixed width (specified and set at installation). Model 2 uses a photosensor mounted on the print mechanism to detect the right-hand edge of a form or passbook, and can accept varying widths in passbooks or cut forms without adjustment. Maximum and minimum width accepted by the Model 2 are the same as for the Model 1. Model 2 has a flat top on which a 3604 can be placed. The 3604/3611 combination requires less space than the 3604/3612 combination.

The basic 3611 printer has a 64-character set that is printed at 15 characters per second, at 12 characters per inch, and either 5 or 6 lines per inch. A print line contains up to 100 characters.

Two special features are available: a set of 96 characters that is printed at up to 30 characters per second, and a shared printer with operator identifier keys that allow two logical work stations to share one 3611 printer.

The character set for Model 2 can be American English, Austrian/German, Brazilian, Danish/Norwegian, Finnish/Swedish, French, Italian, Portuguese, Spanish, Spanish-speaking, or United Kingdom English. Power can be 60 hertz (at 100 or 115 volts) or 50 hertz (at 100, 110, 123.5, 220, or 235 volts).

The 3611 printer is specified in the DEV3611 configuration macro instruction with one physical device address (loop number, terminal address). When referred to in the STATION configuration macro instruction or a data transmission instruction, the printer is referred to by one logical device address.

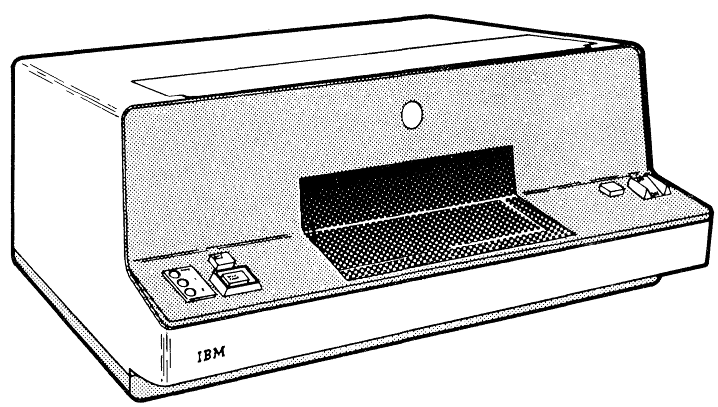


Figure 15-1. IBM 3611 Passbook Printer Model 2

Controls and Indicators

The power on/off switch, the READY indicator, and indicators 1 and 2 are the same as for the 3612 printer; see Chapter 16, "IBM 3612 Passbook and Document Printer" on page 16-1.

Start Print Key

This key activates the passbook mechanism of the printer if a document has been inserted in the printer. It must be pressed before each printing operation on a cut form or a passbook. When this key is pressed, the drive motor is turned on and rotates the print wheel. If the printer is shared, there are two START PRINT keys (one for each station) on opposite sides of the printer.

Stop Print Key

This key stops any printing that is in progress. When this key is pressed, the document is ejected and the drive motor is turned off, stopping the print wheel. The feed rolls are opened so another document can be inserted. A START PRINT key must be pressed before printing resumes.

Passbook and Cut-Forms Printing

Document printing and character sets are the same as for the passbook-printing mechanism of the 3612 printer; see Chapter 16, "IBM 3612 Passbook and Document Printer" on page 16-1.

The 3611 passbook printer can print on passbooks as well as on cut forms such as statements or receipts. The cut forms are in pad form to be torn off for distribution after printing. Capabilities for printing on cut forms are not as varied as these of the 3610 or 3612. Cut-forms specifications are:

- Forms must be in pad form for tearing off after printing.
- Forms width and length restrictions, including the stiffener (backing), are the same as those for the passbook. Minimum length restrictions refer to the torn-off form.
- Minimum thickness is 0.279 mm (0.011 in). This thickness applies to one form plus the backing.
- Maximum thickness is 1.575 mm (0.063 in). This thickness applies to all forms on the pad plus the backing.
- Flexibility and stiffness of the backing must approximate IBM 99-lb card stock.
- Fastening must be on the bottom edge only (unfastened edge to enter the machine first). Staples or any metal or other similarly hard fasteners must not be used.
- Carbon or action paper cannot be used.
- Cut-form pads must not be used if bent, torn, or otherwise mutilated. Mutilated forms can cause a machine malfunction.

Forms that do not meet the specifications outlined above can cause machine damage.

Warning: To avoid damaging the printing mechanism, do not print off the edge of the document and do not print over holes, turned-over edges, folds (including the normal fold), stamps, staples, paper clips, or severely warped pages.

Translating Data and Handling Control Characters

All data and control characters provided by the application program (using WRTI or LWRITE) are translated by the controller using the translation table shown later in this chapter. The translated data is written to the printer from a buffer in the controller. If a control character is detected during translation, translation is suspended until the requested operation is performed. However, the controller does not suspend translation for control characters that result in no operation or in horizontal movement in the same line. If no status results, translation resumes. The controller reports any status to the application program.

Note that the end of a print line occurs at column 100 regardless of the passbook or cut form being printed. If this value is not appropriate for a given passbook or cut form, the controller application program must include new-line control characters where needed.

Variable Parameters

A set of variable parameters describes the physical characteristics of the documents to be printed. The parameters are specified during controller configuration. They can be changed during execution of the controller application program by the DEVPARM instruction. The variable parameters for the 3611 printer are the same as those for the passbook printing mechanism of the 3612 printer.

Controller Configuration for the 3611

Before the 3611 can be operated by a controller program, controller configuration macro instructions must be written to define the physical and logical configuration of all devices attached to the controller. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. Refer to Volume 6 for macro coding information and descriptions of the DEV3611 and other configuration macros.

Terminal Control Characters

The terminal control characters, along with their explanations and functions, are the same as for the 3612; see Chapter 16, “IBM 3612 Passbook and Document Printer” on page 16-1.

Statistical Counters

The statistical counters for the 3611 are the same as for the 3612; see Chapter 16, “IBM 3612 Passbook and Document Printer.”

Status Bits

The status bits and their explanations for the 3611 Printer are listed in the Appendix D, “Terminal and Device Status Codes” on page D-1.

Output Translation Tables

The tables below are the standard output translation tables for the 3611. Characters that do not have corresponding hexadecimal values in these tables are nonstandard. To refer to these characters, you must either give each of them a unique hexadecimal value when it is specified in the OUTRTBL macro, or use a transparent write operation.

For 3611, 64-Character Set

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

Note: With the standard translation table, referring to position 128 results in a space.

For 3611, 96-Character Set

The first 64 characters of the 96-character set table for the 3610/3611/3612 are identical to the 64-character set. This table shows the final 32 characters beginning with relative position 64.

Position	*Hex Value*	*Character*
64	A2	s
65	83	c
66	96	o
67	97	p
68	89	i
69	94	m
70	A3	t
71	A8	y
72	85	e
73	82	b
74	81	a
75	86	f
76	95	n
77	87	g
78	99	r
79	A4	u

Position	*Hex Value*	*Character*
80	93	l
81	88	h
82	84	d
83	A6	w
84	A5	v
85	92	k
86	A7	x
87	79	`
88	A1	~
89	7F	"
90	98	q
91	91	j
92	A9	[illegible]
93	C0	{
94	D0	}
95	6A	¡

Note: With the standard translation table, referring to position 128 results in a space.

For 3611, 128-Character Katakana Set

Position	*Hex Value*	*Character*	
0		⊗	
1	98	ﾇ	Nu
2	A2	ﾍ	He
3	A6	ﾑ	Mu
4	99	ﾈ	Ne
5	AA	ﾕ	Yu
6	A9	ﾔ	Ya
7	A7	ﾒ	Me
8	BB	ﾛ	Ro
9	A3	ﾎ	Ho
10	9F	ﾌ	Fu
11	9E	ﾋ	Hi
12	A5	ﾐ	Mi
13	84	ｴ	E
14	AC	ﾖ	Yo
15	8F	ｾ	Se
16	89	ｹ	Ke
17	92	ﾁ	Chi
18	9D	ﾊ	Ha
19	90	ｿ	So
20	95	ﾄ	To
21	93	ﾂ	Tsu
22	9A	ﾉ	No
23	AE	ﾘ	Ri
24	94	ﾃ	Te
25	BA	ﾚ	Re
26	91	ﾀ	Ta
27	46	ｦ	Wo
28	8D	ｼ	Shi
29	5B	¥	Yen
30	86	ｶ	Ka
31	60	-	minus
32	82	ｲ	I
33	85	ｵ	O
34	BE	ﾞ	KanaSonant
35	8C	ｻ	Sa
36	5C	*	Ast'sk
37	8E	ｽ	Su
38	4B	.	Period
39	AD	ﾗ	Ra
40	F0	0	Zero
41	81	ｱ	A
42	F1	1	One
43	A8	ﾓ	Mo
44	F2	2	
45	56	ｯ	Sm. Tsu
46	F3	3	
47	A4	ﾏ	Ma
48	F4	4	
49	87	ｷ	Ki
50	F5	5	
51	AF	ﾙ	Ru
52	F6	6	
53	88	ｸ	Ku
54	F7	7	
55	96	ﾅ	Na
56	F8	8	
57	BC	ﾜ	Wa
58	F9	9	
59	83	ｳ	U
60	6B	,	comma
61	BD	ﾝ	N
62	97	ﾆ	Ni
63	8A	ｺ	Ko
64	54	ｭ	Sm. Yu
65	D3	L	
66	49	ｩ	Sm. u
67	C6	F	
68	55	ｮ	Sm. Yo
69	C9	]	
70	53	ｬ	Sm. Ya
71	D5	N	
72	BF	ﾟ	Sem.Son
73	E2	S	
74	58	ｰ	Choon
75	D6	O	
76	42	｢	Op.Quot
77	D9	R	
78	43	｣	Cl.Quot
79	64	D	
80	C2	B	
81	E3	T	
82	D7	P	
83	C3	C	
84	D4	M	
85	C1	A	
86	E4	U	
87	C5	E	
88	C7	G	
89	C8	H	
90	E8	Y	
91	E7	X	
92	61	/	
93	E5	V	
94	7E	=	
95	4E	+	
96	D2	K	
97	E0	$	
98	5F	¬	
99	41	｡	Kana Period
100	44	､	Kana Comma
101	5A	!	
102	7F	"	
103	A1	‾	Overline

For 3611, 128-Character Katakana Set (continued)

Position	*Hex Value*	*Character*	*Position*	*Hex Value*	*Character*
104	7D	'	116	45	. Center Per'd
105	6D	_ Underscore	117	4A	¢
106	6E	>	118	7C	@
107	4C	<	119	5D	)
108	5E	;	120	4D	(
109	4F	|	121	D6	O
110	6C	%	122	E9	Z
111	E6	W	123	6F	?
112	D1	J	124	52	ォ Small o
113	7B	#	125	48	ィ Small i
114	50	&	126	47	ァ Small a
115	7A	:	127	51	ェ Small e

Note: With the standard translation table, referring to position 128 results in a space.

Chapter 16. IBM 3612 Passbook and Document Printer

The IBM 3612 Passbook and Document Printer, used with a keyboard display, forms a passbook-handling work station. The 3612 printer (Figure 16-1) is available in three models to print on cut forms, passbooks, continuous forms, or journals:

- Model 1 prints cut forms and passbooks.
- Model 2 prints cut forms, a journal, and passbooks.
- Model 3 prints cut forms, continuous forms, and passbooks.

Two additional models that provide the same capability as Models 2 and 3 but have 128-character print sets are available for Japan only. A 128-character Kanji print set for passbook printing only is also available in Japan. These Japanese models are, respectively, Models 12 and 13.

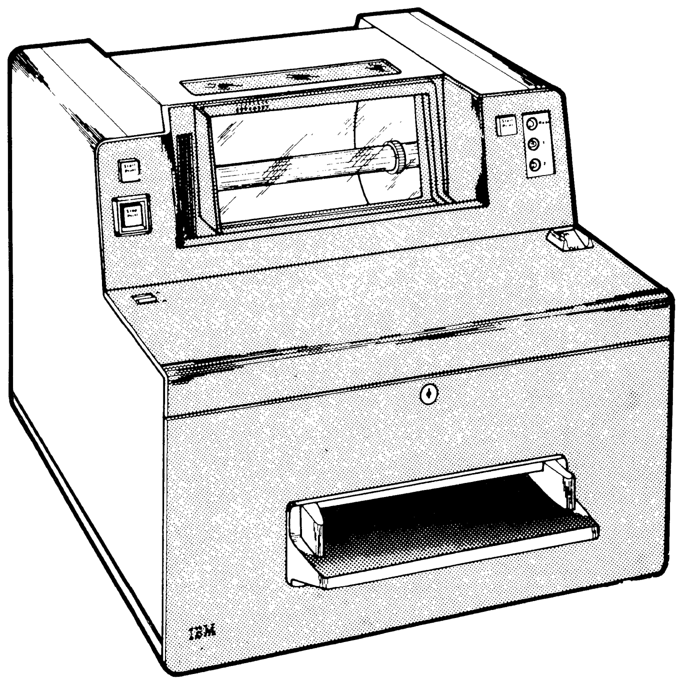


Figure 16-1. IBM 3612 Passbook and Document Printer

The basic 3612 printer prints a set of 64 characters at 15 characters per second. The passbook printing mechanism prints a line of up to 100 characters at 12 characters per inch and either 5 or 6 lines per inch. The document printing mechanism, which prints on cut forms, journal rolls, and continuous forms (depending on the model), prints a line of up to 80 characters at 10 characters per 2.54 cm (inch) and either 5 or 6 lines per inch. All three models contain a passbook printer, and all three models contain a document printer that can print a single line or multiple lines. The Model 2 can also print on a journal roll; the Model 3 can also print on continuous forms.

Warning: To avoid damaging the printing mechanism, do not print off the edge of the passbook or document, and do not print over holes, turned-over edges, folds (including the normal fold), stamps, staples, paper clips, or severely warped pages.

Two special features are available on all three models: (1) a set of 96 characters that is printed at up to 30 characters per second and (2) a shared printer with operator identifier keys that allow two logical work stations to share one 3612 printer. An additional special feature is available on Model 2: the ability to roll the single-part journal or the second part of the two-part journal inside the printer under a locked cover that limits access to the journal.

The character set can be Belgian, Brazilian/Portuguese, Danish/ Norwegian, American English, United Kingdom English, Finnish/Swedish, French, German, Italian, Japanese (Katakana), Spanish, or Spanish-speaking. Power can be 60 hertz (at 100 or 115 volts) or 50 hertz (at 100, 110, 123.5, 220, or 235 volts).

The two printing mechanisms (passbook and document) are similar, whereas the paper-handling mechanisms are different. Both printing mechanisms can have the 64-character print set, or both can have the 96-character print set. The printing mechanisms are never in use simultaneously, because they are individual output devices with distinct component addresses. (The printing mechanisms of the Models 12 and 13 can be in use simultaneously, because these models have two terminal address cards rather than one.) The teller can insert the passbook first and then the cut form, or vice versa, or possibly insert the cut form first, press the START PRINT key, and then insert the passbook.

The 3612 printer is specified by the DEV3612 configuration macro instruction with one physical device address (loop number, terminal address). (Two physical device addresses that form an even-odd pair must be specified for the document and passbook printers of the Model 12 or 13.) When referred to in the STATION configuration macro instruction or a data transmission instruction, the printer is given two logical addresses: one for the passbook printer and one for the document printer.

Controls and Indicators

Power On/Off Switch

This two-position rocker switch with graphic symbols turns power on and off. It should be used with care, because turning power on and off at one terminal can cause errors at other terminals on the same loop.

Start Print Key

This key activates the document and passbook mechanisms of the printer. It must be pressed before each printing operation that occurs either in cut-forms mode (unless autostart in cut-forms mode has been selected) or on a passbook. Pressing this key closes the platen of the document mechanism and turns on the drive motors, which starts the print wheels of the document and passbook mechanisms rotating. When a passbook is inserted, it is sensed by a physical stop and clamped by feed rolls. (If no passbook is inserted and there is no data transfer request to the document mechanism from the controller application program within 20 seconds, the platen is opened and the drive motors are turned off. If a passbook is inserted within 20 seconds after the key is pressed, this time-out does not occur, and both drive motors remain on.) If the printer is a shared terminal, there are two START PRINT keys (one for each teller) located on opposite sides of the printer.

Stop Print Key

This key stops any printing that is in progress. When it is pressed, the platen is opened or the passbook is ejected and the drive motors are turned off. When a passbook is ejected, the passbook feed rolls are opened so another passbook can be inserted. A START PRINT key must be pressed before printing resumes on either mechanism.

Ready Indicator

This light indicates the status of communications on the loop between the printer and the controller. If the Ready indicator is on continuously, communications are normal; if it is off or flashing off and on, there is a communications problem. When the indicator is off, signals from the controller are not arriving at the printer; when it is flashing off and on, signals from the terminals are not reaching the controller (the problem is at the terminal with the flashing indicator or on the loop downstream from that terminal).

Indicators 1 and 2

These lights are defined by the installation and can be set on and off by the controller application program by means of the SIGNAL instruction. For example, the lights on a shared printer could be used to indicate to the operator which logical work station has ownership of the printer.

Cut Forms Stop Selector

This two-position selector switch controls a cut-forms stop. When the cut-forms stop is selected, a mechanical stop within the 3612 limits the depth to which the form can be inserted. When the cut-forms stop is not selected, the depth to which the form can be inserted is not limited.

The 3612 printer can print on passbooks with the following dimensions:

Width:	102 to 221 mm (4.0 to 8.7 in)
Height:	121 to 210 cm (4.75 to 8.25 in)
Thickness:	0.279 to 1.574 mm (0.011 to 0.062 in) for vertical-fold passbooks and 0.279 mm to 1.27 mm (0.011 to 0.050 in) for horizontal-fold passbooks

Margins must be at least 3.2 mm (0.13 in) from the left and right edges of the center page of the passbook and at least 14.8 mm (0.6 in) standard stop or 22.3 mm (0.90 in) optional stop from the top, and 14 mm (0.55 in) above the bottom edge of the passbook. A vertical fold requires at least 4.6 mm (0.20 in) to the left of, and 6 mm (0.24 in) to the right of the center fold. A horizontal fold requires at least 4.2 mm (0.17 in) above and 6 mm (0.24 in) below the center fold. All dimensions are either to the horizontal center lines or to the vertical center lines of characters.

Note: Page size, as defined in the DEV3612 configuration macro or the DEVPARM instruction, determines the limit to which the controller allows a passbook to be indexed. (The passbook is ejected if an attempt is made to index past the last line, as defined by the page definition.) However, if a passbook of smaller size than the passbook defined to the controller is inserted into the 3612, it becomes possible to index off the page, and feeding problems result.

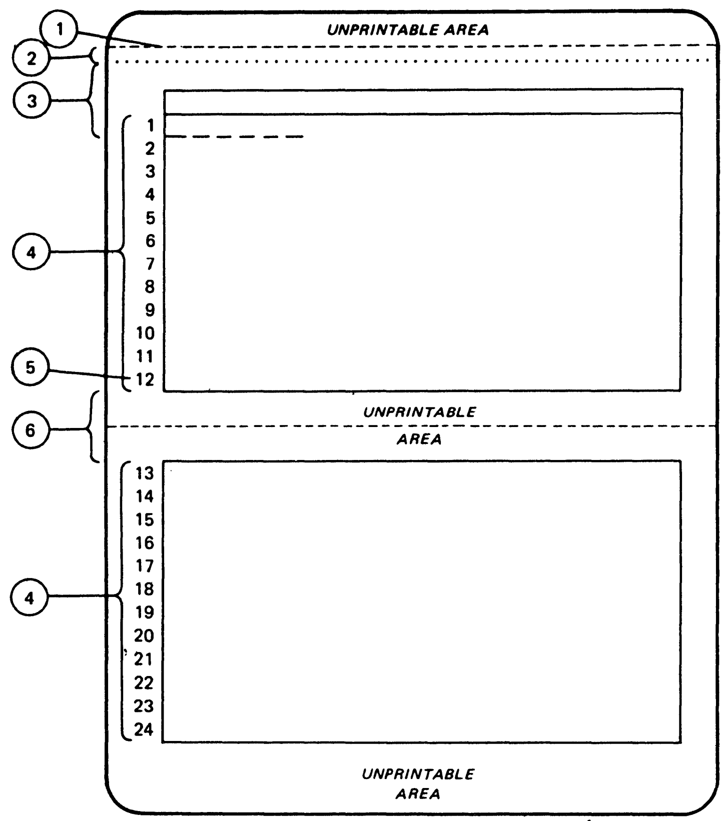
When passbooks are being printed, four factors must be considered: (1) line finding, (2) page definition, (3) end of page, and (4) center-fold detection. A typical passbook is shown in Figure 16-2 .

Line Finding: This is the procedure used to find the next available line in a passbook where printing can start. The controller application program can handle line finding in several ways. The next available line number can be stored as part of the customer's account record in the central processor's data base; or it can be part of the magnetic-stripe information encoded and read by the magnetic stripe encoder/reader if this feature is available; or the teller can determine the line number and enter it by means of the keyboard display. When the teller sets up the printer, the passbook is positioned by the printer to a known reference position (as noted in Figure 16-2). When the START key is pressed, the passbook is moved the number of offset steps and offset lines specified for that printer and becomes positioned at column 1 of logical print line 1. The application program can now position the passbook as required.

Page Definition: A page is defined as the total area that is available on an open passbook; a horizontal center-fold passbook has pages with an unprintable area above and below the center fold, and a vertical center-fold passbook has pages with an unprintable area to the left and right of the center fold.

End of Page: An end-of-page condition is an exceptional condition reported to the application program by means of the condition code (SMSCCD) and status bits (SMSDST). The controller ejects the passbook when requested to index past the last line as defined by the page definition.

Center-Fold Detection: The last print line before a horizontal center fold and the number of lines to be skipped for the center fold must be specified during controller configuration or during program execution using the DEVPARM instruction. These values must be specified correctly to avoid printing on the center fold. Vertical center-fold formatting is handled by the controller application program; characters must not be printed across a vertical center fold. The application program should use the position control character or should insert blanks in the output stream to avoid printing across a vertical center fold (at least 3 character positions are required).



Legend:
1. Clamping position
2. Offset steps to a physical print line
3. Offset lines to the first logical print line
4. Logical print lines that make up the page size
5. Center-fold begin
6. Center-fold skip (at least 6.35 mm (0.25 in.) from the center fold to the center of any character)

Note: The distance from the top of the passbook to the top of the first printable line (offset steps and offset lines set to 0) on a standard 3611 or 3612 is 13.462 mm (0.530 in.); with Feature 9650, the distance is 21.006 mm (0.827 in.).

Figure 16-2. Parameters of a Passbook with a Horizontal Center Fold

Document Printing

All models of the 3612 printer can print on cut forms. On Models 2 and 3, the cut form is inserted in front of the journal paper or pin-fed forms, respectively. Because the printer cannot detect the presence of a cut form in the printing mechanism, it must be defined as being in cut-forms or continuous-forms mode. The mode can be defined during the controller configuration procedure by means of the FT operand of the DEV3612 macro, or it can be set during execution of the application program by means of the DEVPARM instruction. Note that the journal paper and pin-fed forms are both treated as continuous forms.

If both the cut form and the backing paper (continuous forms) are to be printed on, the mode should be defined as cut forms. For this type of printing operation, the application program must take into account the width of the cut form, the width of the backing paper, and their relative horizontal positions. It must also consider the fact that a line index operation advances both the cut form and the backing paper.

Cut-Forms Printing

Cut forms are single-page forms that are inserted by the operator one at a time. They can be multipart forms to provide additional copies of the document. An example of a cut form is a deposit receipt. The printer can print on cut forms with the following dimensions:

Width:	102 to 235 mm (4.0 to 9.25 in)
Minimum Length:	71 mm (2.8 in)
Thickness:	single page 12 lb paper to card stock multipart 12 lb paper, maximum of four parts plus carbon, not to exceed 0.432 mm (0.017 in), including the thickness of the journal or pin-fed form

If the cut-forms stop is used, the cut form is positioned so that the print wheel is 19.05 mm (0.75 in) from the bottom edge of the cut form. If the cut-forms stop is not used, the cut form can be positioned between 279 mm (11 in) and 9.2 mm (0.36 in) from the bottom of the cut form.

Journal Printing

The journal feature (Model 2) is a friction-feed mechanism that enables printing on rolled stock contained within the printer. The paper can be one or two parts; the original exits from the printer past a tear bar after printing. The journal can be 102 mm (4.0 in) to 21.6 cm (8.5 in.) wide, and the roll can be up to 61 mm (2.4 in) in diameter.

The journal take-up feature provides the mechanism to store the single-part journal or the second part of the two-part printed journal on a take-up reel within the printer. The cover lock prevents access under the cover and access to the journal.

Pin-Fed Forms

The continuous-forms feature (Model 3) provides the mechanism to feed continuous pin-fed forms. The form is 241 mm (9.5 in) wide, with a width of 229 mm (9.0 in) between the centers of the pin-feed holes. The forms can have up to four parts; the maximum thickness (including a cut form, if used) that can be accommodated is 0.432 mm (0.017 in).

The document printer, Model 3, can detect an end-of-forms condition by means of a switch on the printer that is set when the end of the continuous forms passes it. At least 102 mm (4 in) of paper remain when the switch is first set. Until the operator inserts more paper (resetting the switch), intervention-required (out-of-forms) status is returned in SMSDST whenever the controller starts the motors and with every print operation completed.

Printing Modes

The printer operates in cut-forms, continuous-forms, or passbook mode, and can be in a ready or not-ready state. The mode of the printer determines its initial state, and the state of the printer determines how the printer is started. Certain actions cause the state of the printer to change. Figure 16-3 shows the changes that can occur in the state of a document printer and the reasons for the changes.

Cut-Forms Mode

In cut-forms mode, a printer is initially in a not-ready state. The operator must make it ready by inserting the document to be printed and pressing a START PRINT key, unless the printer is in cut-forms automatic-start mode. This action causes the platen mechanism to grip the document, start the printer motors, and signal the controller that the printer is ready.

The document is released and the motors are stopped when:

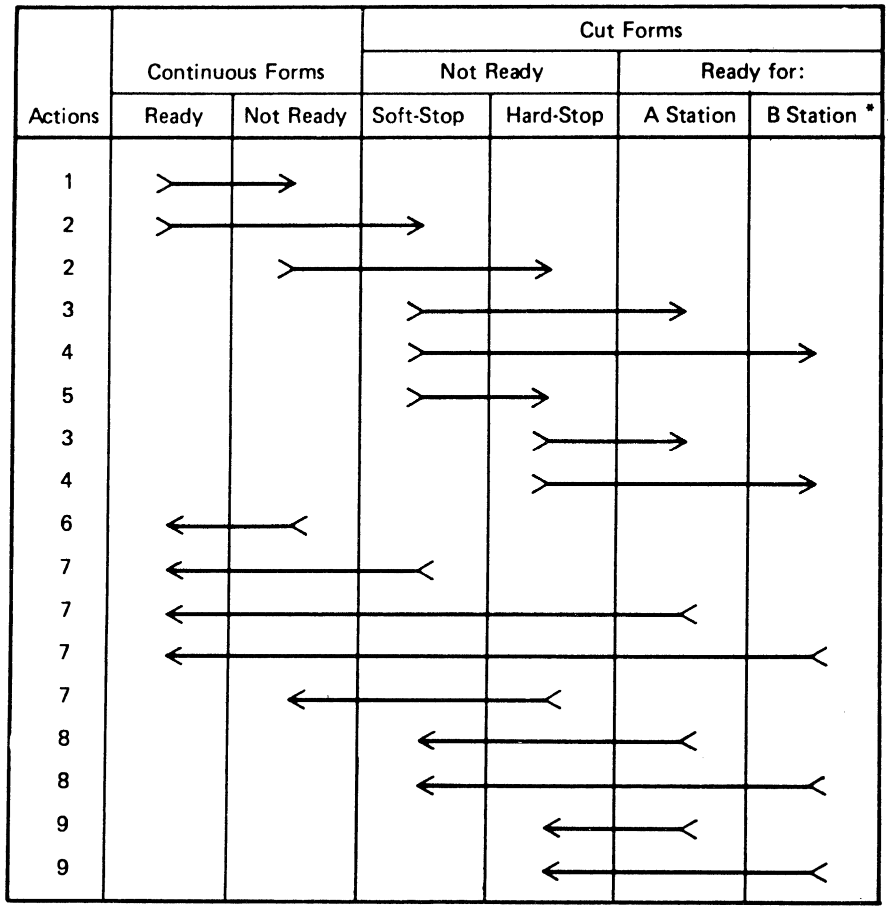
- The application program writes a form-feed control character.
- The application program attempts to index past the defined end of page.
- The application program issues a DEVPARM instruction to redefine the document characteristics.
- The operator presses the STOP PRINT key or opens the cover.
- The printer has been ready for a period of approximately 20 seconds and no write instructions have been issued to it.

When a form-feed control character is detected or an attempt is made to index past the defined end-of-page, the document is indexed 1 line before being released. Any of these actions returns the printer to the not-ready state.

The not-ready state of a printer in cut-forms mode can be further described as either soft-stopped or hard-stopped, depending on the way the printer entered the not-ready state. When the printer is soft-stopped, a START PRINT key must be pressed before further cut-forms printing can occur. If the mode is changed to continuous forms by means of a DEVPARM instruction, however, the printer can be started automatically by issuing a write instruction. When the printer is hard-stopped, a START PRINT key must be pressed before any further printing can occur even if the mode is changed by a DEVPARM instruction. The initial not-ready state is soft-stopped.

Continuous-Forms Mode

In continuous-forms mode, a printer is initially in a ready state, but its motors are stopped. The motors are started automatically whenever a write instruction is issued to the printer and stopped automatically if approximately 20 seconds elapse during which no write instructions are issued to the printer. In this mode, therefore, the printer can operate without an operator (for example, to print teller or administrative reports).



Actions:

1. The STOP PRINT key was pressed, the cover was opened, or the printer is out of forms (Model 3 only).
2. A DEVPARM instruction was issued to change to cut-forms mode.
3. The START PRINT or START PRINT A key was pressed.
4. The START PRINT B key was pressed.
5. The cover was opened.
6. Any START PRINT key was pressed.
7. A DEVPARM instruction was issued to change to continuous-forms mode.
8. A time-out occurred, or a DEVPARM instruction specifying cut-forms mode was issued.
9. The STOP PRINT key was pressed, the cover was opened, or a form feed control character was sent, or an attempt was made to index past the defined end-of-page.

*This state does not exist if the printer was not defined as shared.

Figure 16-3. Actions That Change the State of the Document Printer

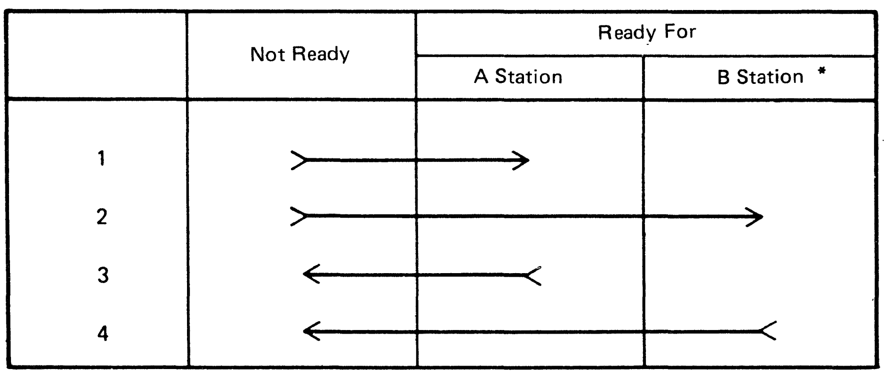
A printer in continuous-forms mode is put in a not-ready state if:

- The printer is stopped by pressing the STOP PRINT key or opening the cover.
- A Model 3 document printer detects the end of forms. The operator can allow successive lines to be printed, one at a time, by repetitively pressing a START PRINT key until the end of a page is reached. At that time, the operator can replace the forms and press the START PRINT key once more to resume normal printing.
- After a form feed control character has been sent to the printer in cut-forms mode to release the document, the mode is changed to continuous forms by means of a DEVPARM instruction. Pressing the START PRINT key is an indication that the cut form was removed from the printer.

A printer in continuous-forms mode in a not-ready state can be made ready by pressing a START PRINT key.

Passbook Mode

With respect to its ready and not-ready states, a passbook printer operates in a manner similar to that of a document printer in cut-forms mode. There is, however, no idle time-out on a passbook printer and no soft-stopped state. Figure 16-4 shows the changes that can occur in the state of a passbook printer and the reasons for the changes.



Conditions:

1. The START PRINT or START PRINT A key was pressed.
2. The START PRINT B key was pressed.
3. The STOP PRINT key was pressed, the cover was opened, a form feed control order was sent, an attempt was made to index past the defined end-of-page, or a DEVPARM instruction was issued. In all cases, the passbook is ejected from the machine.

*This state does not exist if the printer was not defined as shared.

Figure 16-4. Actions That Change the State of the Passbook Printer

Character Sets

There are 64-, 96-, and 128-character sets available with the passbook and document printer. The 64- and 96-character sets are available with any of the language groups offered. The 128-character set for Models 12 and 13 is for Japan only.

The printable characters are contained on a metal wheel; the wheel positions and characters are fixed. The 64-character set utilizes a print wheel that contains one 64-character array. The print speed is 15 characters per second. The 96-character set utilizes a print wheel containing two 48-character arrays. One array contains numerics, uppercase characters, and certain special characters, while the second array contains lowercase characters and additional special characters. Printing characters alternately from each array results in a print speed of approximately 20 characters per second; as the shift from one array to the other occurs less frequently, the print speed approaches 30 characters per second. If printing occurs from one array only, the print speed is 30 characters per second. The maximum print speed for the 128-character set is 22.5 characters per second.

The characters are represented in translation tables in the controller by the binary equivalent of their positions on the print wheel. The 64-, 96-, and 128-character sets are shown later in this chapter. The characters that can be substituted for international use on the 64- and 96-character sets are shown in Figure 16-5.

COUNTRY \ PRINT WHEEL POSITION	• 7	• 56	• 9	• 63	• 50	• 62	• 58	87	93	95	94	88
UNITED STATES	#	$	@	[	\	]	¬	`	{	¦	}	~
BELGIUM	#	$	à	[	¸	]	^	`	é	ù	è	¨
BRAZIL	Õ	Ç	Ã	É	\	$	^	ã	õ	¸	é	~
DENMARK/NORWAY	Æ	Å	Ø	#	\	¤	¬	`	æ	ø	å	ü
FRANCE	£	$	à	°	¸	§	^	`	é	ù	è	¨
GERMANY	Ä	Ü	Ö	#	§	$	¬	`	ä	ö	ü	ß
ITALY	£	$	§	°	¸	é	^	ù	à	ò	è	ì
JAPAN	#	¥	@	£	$	!	¬	`	{	¦	}	‾
PORTUGAL	Ã	$	Õ	[	Ç	]	^	`	ã	õ	´	¸
SPAIN	Ñ	₧	@	[	\	]	¬	`	{	ñ	}	¨
SPANISH SPEAKING	Ñ	$	@	[	\	]	¬	`	{	ñ	}	¨
SWEDEN/FINLAND	Ä	Å	Ö	§	É	¤	¬	é	ä	ö	å	ü
UNITED KINGDOM	£	$	@	[	\	]	¬	`	{	¦	}	‾

*Options available for the 64-character set only.

Figure 16-5. International Character Options

Translating Data and Handling Control Characters

All data and control characters provided by the application program (using WRTI or LWRITE) are translated by the controller using the translation table shown later in this chapter. The translated data is written to the printer from a buffer in the controller. If a control character is detected during translation, translation is suspended until the requested operation is performed. However, the controller does not suspend translation for control characters that result in no operation or in horizontal movement in the same line. If no status results, translation resumes. The controller reports any status to the application program.

Variable Parameters

A set of variable parameters describes the physical characteristics of the documents and passbooks to be printed. The parameters are specified during controller configuration. They can be changed during execution of the controller application program by means of the DEVPARM instruction.

For document printing, the variable parameters are:

- *Forms type:* A code designating the type of forms being printed: X'01' indicates cut forms with start key required; X'41' indicates cut forms and autostart; X'02' indicates continuous forms with no concurrent sharing; X'82' indicates continuous forms with concurrent sharing. Refer to Chapter 3, "Programming for Printers" on page 3-1 for a description of concurrent sharing.

- *Page size:* The number of print lines on a page. Indexing past the last line on a page causes unit-exception status to be set for the application program; the form is positioned to column 1 of line 1 of the next page. Indexing past the defined end-of-page of a cut form causes a 1-line index, the document to be released, and the motors stopped. This function can be bypassed by specifying a page size of 0. The maximum valid page size is 127 lines.

- *Warning line:* The number of any print line on the page. Indexing past this line causes unit-exception status to be set for the application program and the form to be positioned to the line following the warning line. This function can be bypassed by specifying a warning line or page size of 0. The application program can use the warning line to indicate when to print footings or when to skip to a new page.

- *Deblouse:* A parameter that specifies the number of lines to be printed on friction feed forms before deblousing (release tension on the feed rolls). The number of lines can be from 0 to 255. The more parts a form has, the more frequently it must be debloused. The default is 10. Too frequent deblousing reduces performance. No deblousing occurs when 0 is specified.

For passbook printing, a physical print line is any line that can be indexed to; a logical line is any line that can be printed on. A physical print line can not be a logical print line, depending on the gap at the top of the page and the gap over a horizontal fold. The application program refers only to logical print lines.

When a passbook is inserted, the 3612 printer clamps it in a fixed position. Because of variations in passbook format, this position is not necessarily at the first logical print line or even an integral number of physical print lines from the first logical print line. When the 3612 printer is activated, two index operations are performed: a fractional line index of so many steps that aligns the passbook on a physical print line and a full line index of so many physical lines that advances the passbook to the first logical print line.

If the passbook has a horizontal center fold, there is a gap at the center, which contains physical print lines that are not counted as logical print lines and which is indexed across automatically by the controller during printing operations. (See Figure 16-2 for an illustration of the parameters of a passbook with a horizontal fold.)

The variable parameters for the 3612 passbook printing mechanism are:

- *Forms type:* X'00' indicates passbook printing.

- *Page size:* The number of logical print lines on the passbook. The page size cannot be 0 or negative.

- *Center-fold begin:* The logical line number of the last line to be printed before the center fold (for horizontal-fold books only).

- *Center-fold skip:* The number of physical line indexes required to bypass a horizontal center fold. This number should not be less than 2.

- *Offset steps:* The number of steps (fractional lines) a passbook must be indexed on insertion to move it from the clamping position to an integral print line. (One step 0.410 mm (0.016 in.): up to 31 steps can be specified.)

- *Offset lines:* The number of full lines a passbook must be indexed to move it from the first integral line after clamping to the first logical print line. The maximum number is 31.

Note: The sum of the page size plus the offset lines plus the center-fold skip cannot exceed 42.

Terminal Control Characters

Control characters are embedded in the data stream and indicate operations to be performed at a terminal. The following are the control characters for the 3612:

Function	Control Character	Explanation
Carriage Return	X'0D'	The print element carriage in the designated printer is returned to the extreme left print position; the paper or passbook line is unchanged.
Line Feed	X'25'	The paper or passbook is spaced up one line; the print element position is unchanged.
New Line	X'15'	The paper or passbook is spaced up one line and the associated print element returns to the extreme left print position.

Function	Control Character	Explanation
Position	X'34'	This control character is followed by a flag byte and a 1-byte positioning value which control the repositioning of the print element and the paper or passbook. The flag byte has the following control: Bits 0-3 are ignored. Bit 4: 0 = Position is an absolute value. 1 = Position is relative to the present position. Bit 5: 0 = Position change is horizontal. 1 = Position change is vertical. Bits 6-7 of the flag byte are ignored. The positioning value indicates a line or column number. It contains an absolute value from 0 to 255 to use under the direction of the flag byte. A 0 results in no operation.
Form Feed	X'0C'	Release the document for cut-forms printing; skip to a new page for continuous-forms printing; release the passbook for passbook printing.
Transparent Write	X'35'	A transparent write (a write without translation) begins. The format of this control sequence is: Byte 0: The control character, X'35'. Byte 1: The number of wheel positions to be printed. Byte 2-Byte n: The wheel positions to be printed. The valid positions are X'00'-X'80' and X'A8' (no operation).
Select	X'04'	The byte following this control character is skipped (no operation takes place).

Programming Considerations

To improve performance for passbook and document printing, use these operational and programming procedures:

- The teller should press the START key on the passbook printer before inserting the passbook. This allows the printer motors to start and allows the signals generated by the printer, as a result of the teller pressing the START key, to be overlapped with the insertion of the passbook.

  A form feed control character should immediately follow the last character of the last line to be printed on the passbook. This inhibits the controller from indexing an unnecessary line and also eliminates the overhead associated with processing the form feed as a separate operation.

- A vertical positioning control character should be used, rather than multiple line-feed control characters, to index multiple lines. This eliminates the overhead incurred by issuing multiple control characters where one is sufficient.

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or they can be restarted, or "wrapped back," to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 3612:

Counter	Explanation
1	Loop error checks
2	Terminal address card unit checks
3	Intervention required
4	Emitter check
5	End of forms
6	Platen open
7	Time-out
8	Defective dashpot (missing left margin indication after carriage return)

Status Bits

The status bits and their explanations are in Appendix D, "Terminal and Device Status Codes" on page D-1.

Controller Configuration for The 3612

Before the 3612 can be operated by a controller program, controller configuration macro instructions must be written to define the physical and logical configuration of all devices attached to the controller. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. Refer to Volume 6 for macro coding information and descriptions of the DEV3612 and other configuration macros.

The DEV3612 macro instruction is used in conjunction with the DEFADDR and LDA macros to describe a 3612 document and passbook printer to the controller.

Translation Tables

The tables below are the standard output translation tables for the 3612. Characters that do not have corresponding hexadecimal values in these tables are nonstandard. To refer to these characters, you must either give each of them a unique hexadecimal value when it is specified in the OUTRTBL macro, or use a transparent write operation.

For 3612, 64-Character Set

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

Note: With the standard translation table, referring to position 128 results in a space.

For 3612, 96-Character Set

The first 64 characters of the 96-character set table for the 3610/3611/3612 are identical to the 64-character set. This table shows the final 32 characters beginning with relative position 64.

Position	*Hex Value*	*Character*	*Position*	*Hex Value*	*Character*
64	A2	s	80	93	l
65	83	c	81	88	h
66	96	o	82	84	d
67	97	p	83	A6	w
68	89	i	84	A5	v
69	94	m	85	92	k
70	A3	t	86	A7	x
71	A8	y	87	79	`
72	85	e	88	A1	~
73	82	b	89	7F	"
74	81	a	90	98	q
75	86	f	91	91	j
76	95	n	92	A9	z
77	87	g	93	C0	{
78	99	r	94	D0	}
79	A4	u	95	6A	¦

Note: With the standard translation table, referring to position 128 results in a space.

For 3612, 128-Character Katakana Set

Position	*Hex Value*	*Character*	
0		⊗	
1	98	ヌ	Nu
2	A2	ヘ	He
3	A6	ム	Mu
4	99	ネ	Ne
5	AA	ユ	Yu
6	A9	ヤ	Ya
7	A7	メ	Me
8	BB	ロ	Ro
9	A3	ホ	Ho
10	9F	フ	Fu
11	9E	ヒ	Hi
12	A5	ミ	Mi
13	84	エ	E
14	AC	ヨ	Yo
15	8F	セ	Se
16	89	ケ	Ke
17	92	チ	Chi
18	9D	ハ	Ha
19	90	ソ	So
20	95	ト	To
21	93	ツ	Tsu
22	9A	ノ	No
23	AE	リ	Ri
24	94	テ	Te
25	BA	レ	Re
26	91	タ	Ta
27	46	ヲ	Wo
28	8D	シ	Shi
29	5B	¥	Yen
30	86	カ	Ka
31	60	−	minus
32	82	イ	I
33	85	オ	O
34	BE	゛	KanaSonant
35	8C	サ	Sa
36	5C	*	Ast'sk
37	8E	ス	Su
38	4B	.	Period
39	AD	ラ	Ra
40	F0	0	Zero
41	81	ア	A
42	F1	1	One
43	A8	モ	Mo
44	F2	2	
45	56	ッ	Sm. Tsu
46	F3	3	
47	A4	マ	Ma
48	F4	4	
49	87	キ	Ki
50	F5	5	
51	AF	ル	Ru
52	F6	6	
53	88	ク	Ku
54	F7	7	
55	96	ナ	Na
56	F8	8	
57	BC	ワ	Wa
58	F9	9	
59	83	ウ	U
60	6B	,	comma
61	BD	ン	N
62	97	ニ	Ni
63	8A	コ	Ko
64	54	ュ	Sm. Yu
65	D3	L	
66	49	ゥ	Sm. u
67	C6	F	
68	55	ョ	Sm. Yo
69	C9	I	
70	53	ャ	Sm. Ya
71	D5	N	
72	BF	゜	Sem.Son
73	E2	S	
74	58	ー	Choon
75	D6	O	
76	42	「	Op.Quot
77	D9	R	
78	43	」	Cl.Quot
79	64	D	
80	C2	B	
81	E3	T	
82	D7	P	
83	C3	C	
84	D4	M	
85	C1	A	
86	E4	U	
87	C5	E	
88	C7	G	
89	C8	H	
90	E8	Y	
91	E7	X	
92	61	/	
93	E5	V	
94	7E	=	
95	4E	+	
96	D2	K	
97	E0	$	
98	5F	¬	
99	41	.	Kana Period
100	44	,	Kana Comma
101	5A	!	
102	7F	"	
103	A1	‾	Overline

For 3612, 128-Character Katakana Set (continued)

Position	*Hex Value*	*Character*	
104	7D	'	
105	6D	_	Underscore
106	6E	>	
107	4C	<	
108	5E	;	
109	4F	|	
110	6C	%	
111	E6	W	
112	D1	J	
113	7B	#	
114	50	&	
115	7A	:	
116	45	.	Center Per'd
117	4A	£	
118	7C	@	
119	5D	)	
120	4D	(	
121	D6	Q	
122	E9	Z	
123	6F	?	
124	52	ｫ	Small o
125	48	ｨ	Small i
126	47	ｧ	Small a
127	51	ｪ	Small e

Note: With the standard translation table, referring to position 128 results in a space.

For 3612 Passbook Printer, 128-Character Katakana-Kanji Set

This output table is identical to the table previously shown for the 128-character Katakana table, except for the characters shown at positions 98 through 109 displayed below.

Position	*Hex Value*	*Character*
98	5F	繰 Kuri
99	41	新 Shin
100	44	解 Kai
101	5A	約 Yaku
102	7F	越 Koshi
103	A1	現 Gen
104	7D	振 Furi
105	6D	他 Ta
106	6E	息 Soku
107	4C	利 Ri
108	5E	訂 Tei
109	4F	正 Sei

Note: With the standard translation table, referring to position 128 results in a space.

Chapter 17. IBM 3615 Administrative Terminal Printer

The IBM 3615 Administrative Terminal Printer (Figure 17-1 and Figure 17-2) is a medium-speed, tabletop, matrix printer that can be used to print a variety of cut forms and fan-fold continuous forms. The 3615 printer is available in two models:

- Model 1
  - Prints 60 characters per second
  - Has a tab speed of 228.6 mm (9 in) per second
  - Attaches to either local or remote loops
- Model 2
  - Prints 120 characters per second
  - Has a tab speed of 304.8 mm (12 in) per second
  - Attaches to the local loop only

The 3615 prints by means of an 8-wire matrix printing mechanism. Printing is done in both directions with the following print characteristics:

- Ten characters per inch
- Up to 132 characters per line
- Six lines per inch

Programmable indexing capability and end-of-forms detection are available on both Models 1 and 2.

One special feature is available: a variable-width forms tractor (Figure 17-2) that is required for use with continuous forms. A forms stand is also available as an accessory.

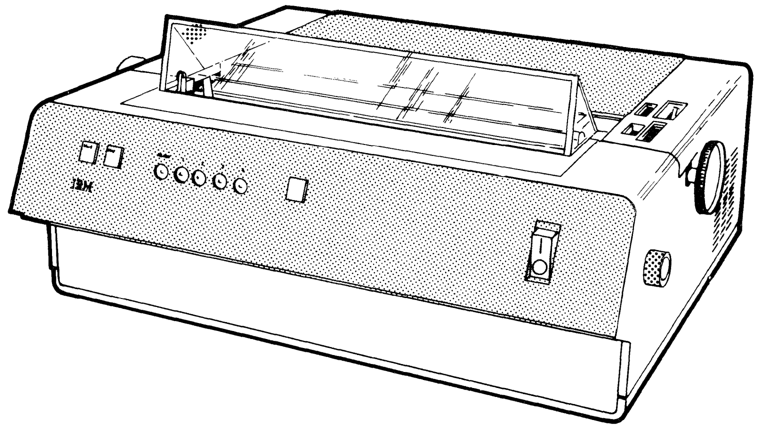


Figure 17-1. IBM 3615 Administrative Terminal Printer

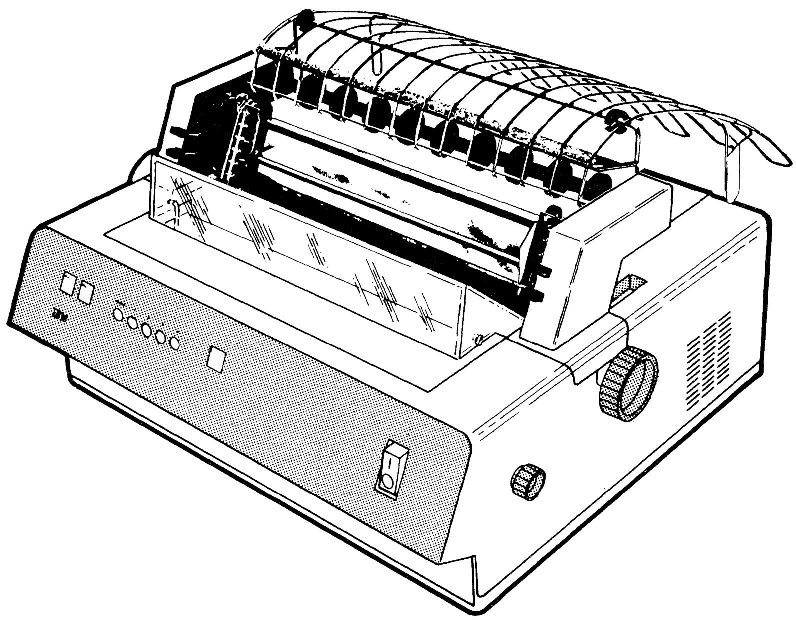


Figure 17-2. IBM 3615 with Forms Tractor Installed

The high-speed tab is performed by Model 1 for those positions in excess of 10. For up to 10 positions the tab speed equals the printing speed. Model 2 performs tab operations at the same speed as the printing speed.

The platen and forms mechanism are similar to the IBM Selectric typewriter and will handle individual forms in the same manner.

Warning: Multipart cut forms must be glued at the top. No staples or clips should be used, because they can damage the printer.

Crimped forms separate too easily and are therefore not recommended. Cut forms with up to 6 parts can be used; however, you should test forms with more than 4 parts for acceptable printing quality. For continuous forms, a maximum of four parts is recommended for best feeding and stacking.

Controls and Indicators

Power On/Off Switch

This two-position rocker switch with graphic symbols turns power on and off. It should be used with care, because turning power on and off at one terminal may cause errors at other terminals on the same loop.

Start Key

This key readies the 3615 printer for printing following intervention by the operator.

Stop Key

This key stops any printing after the current print line is completed. The Start key must be pressed before printing can begin again.

Ready Indicator

This light indicates the status of communications on the loop between the 3615 printer and the 4700 controller. If the READY indicator is on, communications are normal. If it is off or flashing off and on, there is a communications problem: signals from the printer are not reaching the controller. When the indicator is off, signals from the controller are not arriving at the 3615 printer.

Indicators 1-4

These lights are definable by the customer and can be controlled by the application program by means of the Program Operator Communication (POC) control character. For example, one of the lights could be used to indicate to the operator that intervention is required; this could be especially useful if no 3604 keyboard display is assigned to the logical work station controlling the 3615 printer.

Document Printing

Both models of the 3615 will print on forms with up to six parts, on single sheets, and on continuous forms consisting of an original and up to three carbon copies. Five- and six-part forms, with a total thickness of 0.457 mm (0.18 in.), should be tried on an individual basis for acceptable feeding, registration, and print quality. A forms tractor is required to handle continuous forms. Friction-fed cut forms may also be used without removing the forms tractor. In this case, printing is limited to 127 print positions per line.

The physical characteristics of the documents used are described during configuration. They can be changed during program execution by the DEVPARM instruction. Such characteristics are page size, warning line, and forms type.

Cut Forms

Cut forms are single-page, friction-fed forms that are inserted one at a time. They may be of the following dimensions:

Length:	7.0 to 35.6 cm (2.75 to 14.0 in)
Width:	15.25 to 35.6 cm (6.0 to 14.0 in)
Thickness:	Single-part forms, 0.08 to 0.2 mm (0.003 to 0.0075 in) Multipart forms, 0.08 to 0.46 mm (0.003 to 0.018 in)

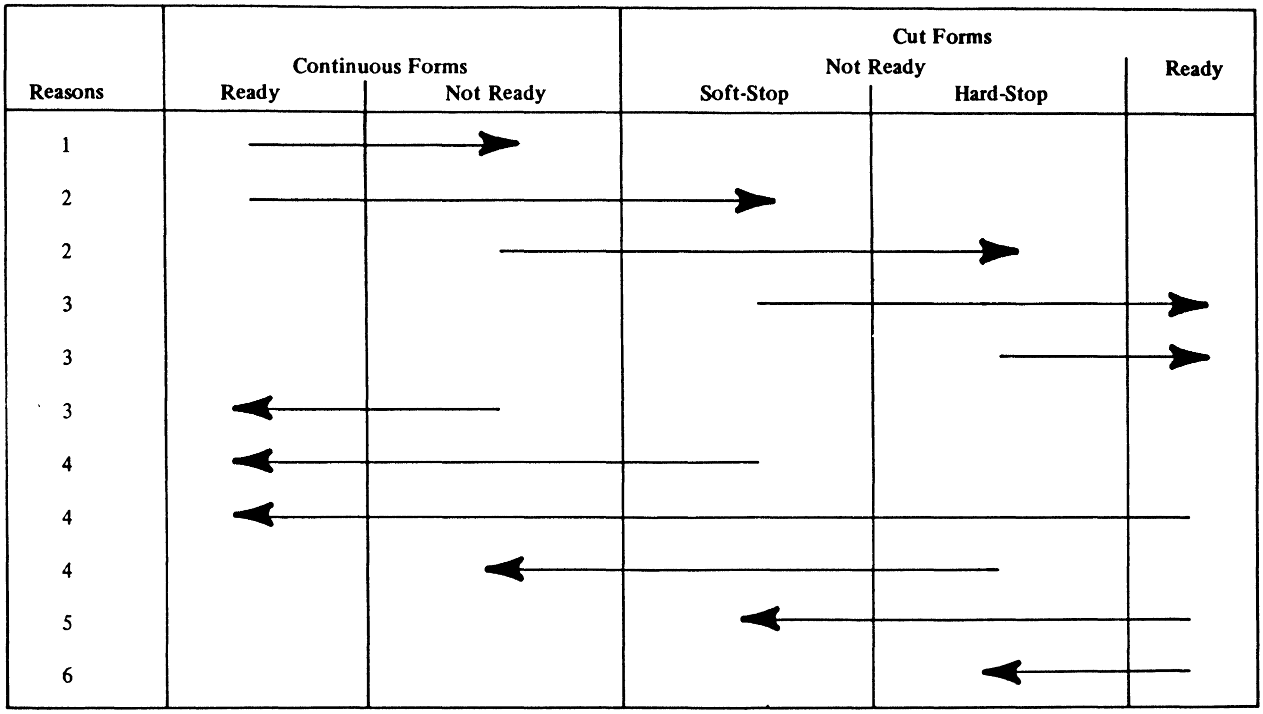
Continuous Forms

Continuous forms may be of the following dimensions:

Width:	76.2 to 381 mm (3.0 to 15.0 in)
Distance between folds:	7.6 to 35.6 cm (3.0 to 14.0 in)
Thickness:	Single-part forms, 0.08 to 0.13 mm (0.003 to 0.005 in) Multipart forms, 0.08 to 0.46 mm (0.003 to 0.018 in)

Printing Modes

The printer operates in cut-forms or continuous-forms mode and may be in a ready or not–ready state. The mode of the printer determines its initial state, and the state of the printer determines how the printer is started. Certain actions cause the state of the printer to change. Figure 17-3 shows the changes that can occur in the state of a document printer and the reasons for the changes.



Notes:

1. The Stop key was pressed, or the cover was opened.
2. A DEVPARM instruction was issued to change to cut-forms mode.
3. The Start key was pressed.
4. A DEVPARM instruction was issued to change to continuous-forms mode.
5. A time-out occurred, or a DEVPARM instruction specifying cut-forms mode was issued.
6. The Stop key was pressed, or a form-feed control character was sent, or an attempt was made to index past the defined end of page.

Figure 17-3. Actions That Change the State of the 3615

Cut-Forms Mode

In cut-forms mode, a printer is initially in a not–ready state. The operator must make it ready by inserting the document to be printed and pressing the Start key, unless the printer is in cut-forms automatic-start mode. This action signals the controller that the printer is ready.

Printing on the document ends and the print head is returned to the left margin of the printer when:

- The application program writes a form-feed control character.
- The application program attempts to index past the defined end of page.
- The application program issues a DEVPARM instruction to redefine the document characteristics.
- The operator presses the Stop key. If a printing operation is in progress when the Stop key is pressed, the current line completes printing and Intervention Required status is posted.

- The printer has been ready for approximately 10 seconds and no write instruction has been issued.

When a form-feed control character is detected or an attempt is made to index past the defined end of page, the document is indexed one line before being released. Any of these actions returns the printer to the not-ready state.

A printer in the not-ready state is either semi-stopped or fully stopped, depending on the way the printer entered the not-ready state. When the printer is semi-stopped, a Start key must be pressed before further cut-forms printing can occur. If the mode is changed to continuous forms by means of a DEVPARM instruction, however, the printer can be started automatically by a write instruction. When the printer is fully stopped, the operator must press the Start key before any more printing can occur, even if the mode is changed by a DEVPARM instruction. The initial not-ready state is semi-stopped.

Continuous-Forms Mode

In continuous-forms mode, a printer is initially in a ready state and its print head is at the left margin. When messages are sent to the printer, the print head moves and the line is printed. The print head is returned automatically to the left margin if approximately 10 seconds elapse during which no write instructions are issued to the printer. In this mode, therefore, the printer can operate without an operator (for example, to print teller or administrative reports).

A printer in continuous-forms mode is put in a not-ready state if:

- The printer is stopped by pressing the Stop key. If a printing operation is in progress when the Stop key is pressed, the current line completes printing and Intervention Required status is posted.

- The 3615 printer detects the end of the forms. The operator can allow successive lines to be printed, one at a time, by repetitively pressing a Start key until the end of a page is reached. At that time, the operator can replace the forms and press the Start key once more to resume normal printing.

- After a form-feed control character is sent to the printer in cut-forms mode to release the document, the mode is changed to continuous forms by means of a DEVPARM instruction. Pressing the Start key is an indication that the cut form was removed from the printer.

A printer in continuous-forms mode in a not-ready state can be made ready by the pressing of a Start key.

Character Set

The 3615 has a 128-character set which includes the International, Canadian French, and EBCDIC (USA) options.

Unlike the 3610, 3611, and 3612 printers, which have a print wheel, the 3615 has an 8-wire matrix printing mechanism.

The 128-character set is defined in the 3615 as a set of 4-out-of-7 wide by 8 high dot matrixes which correspond to position numbers 0 through 127. The characters are represented in a translation table in the controller as a group of 128 entries, each associating an EBCDIC value with a position number. The controller has a default output translation table for the 3615 character set, which is defined later in this chapter.

Translating Data and Handling Control Characters

All data and control characters provided by the application program (using WRTI or LWRITE) are translated by the controller using the translation table shown later in this chapter. The translated data is written to the printer from a buffer in the controller. If a control character is detected during translation, translation is suspended until the requested operation is performed. However, the controller does not suspend translation for control characters that result in no operation or in horizontal movement in the same line. If no status results, translation resumes. The controller reports any status to the application program.

Variable Parameters

A set of variable parameters describes the physical characteristics of the documents to be printed. The parameters are specified during controller configuration. They can be changed during execution of the controller application program by means of the DEVPARM instruction. The variable parameters for the 3615 printer are:

- Page Size: The number of physical print lines on a page. A page size of zero results in a page of infinite length.

- Warning Line: The line number of any physical print line on the printer. Passing this line during relative vertical skipping causes unit exception status to be posted to the program. This function can be bypassed by specifying a warning line or page size of zero.

- Forms Type: A code in the flag byte of DEVPARM indicates the type of forms being used:

  X‘01’ Cut forms, Start key required

  X‘41’ Cut forms, Autostart

  X‘02’ Continuous Forms, no concurrent sharing

  X‘82’ Continuous Forms, concurrent sharing

Terminal Control Characters

Control characters are embedded in the data stream and indicate operations to be performed at a terminal. The following are the control characters for the 3615:

Function	Control Character	Explanation
Forms Feed	X'0C'	For continuous forms, lines are skipped until the paper is positioned to print the first print line of a new page. For cut forms, the document is released.
Select	X'04'	The byte following this control character is ignored; automatic new line is suppressed.
New Line	X'15'	The paper is spaced up one line, and the print element position is set to the extreme left print position.
Program Operator Communication	X'17'	This character, followed by two indicator control bytes, turns one or more 3615 indicators on or off. Control byte 1 is an on/off control, and control byte 2 selects individual indicators (using the right hexadecimal character), or acts as a mask (using the left hexadecimal character) to turn on one or more indicators simultaneously. The following table and chart show how to code the control character and control bytes to control either individual indicators, or to turn on one or more indicators simultaneously:

Controlling individual indicators:

3615

Indicator:	On:	Off:
1	17 C1 01	17 C2 01
2	17 C1 02	17 C2 02
3	17 C1 03	17 C2 03
4	17 C1 04	17 C2 04

Function	Control Character	Explanation

Controlling all indicators simultaneously:

	Control Byte 2	3615 Indicator: 1	2	3	4
17 C3 . . .	0 0				

Turns all indicators off*

	Control Byte 2	3615 Indicator: 1	2	3	4
17 C3 . . .	1 0				•
	2 0			•	
	3 0			•	•
	4 0		•		
	5 0		•		•
	6 0		•	•	
	7 0		•	•	•
	8 0	•			
	9 0	•			•
	A 0	•		•	
	B 0	•		•	•
	C 0	•	•		
	D 0	•	•		•
	E 0	•	•	•	
	F 0	•	•	•	•

Turns indicators marked • on

*Unwanted indicators must first be turned off before desired indicators are turned on.

Position	X'34'	This control character is followed by a flag byte and a 1-byte positioning value that control the repositioning of the print element and paper. The flag byte has the following control:

Bits 0-3 are ignored.

Bit 4: 0 = Position is an absolute value.
1 = Position is relative to the present position.

Bit 5: 0 = Position change is horizontal.
1 = Position change is vertical.

Bits 6-7 of the flag byte are ignored. The positioning value indicates a line or column number. It contains an absolute value from 0 to 255 for horizontal positioning and from 0 to 127 for vertical positioning to use under the direction of the flag byte. A 0 results in no operation.

Function	Control Character	Explanation
Transparent Write	X'35'	A transparent write (without translation) begins. The format of this control sequence is: Byte 0: The control character, X'35'. Byte 1: The number of belt positions to be printed. Byte 2-Byte n: The belt positions to be printed. The valid positions are X'00'-X'7F'.
Line Feed	X'25'	The paper is spaced up one line; the print position is unchanged.
Carriage Return	X'0D'	The print element is set to the extreme left print position; the paper line is unchanged; the print head is not moved.

Programming Considerations

Each separate data transmission to the 3615 incurs some loop overhead. All control characters that require vertical forms control, either actual (new line, carriage return, line feed, form feed, vertical position, or selection) or implied (horizontal position where the new position is less than the current position, or the relative position plus the current position exceeds the right margin of the printer), cause the 4700 controller to divide the message into submessages for transmission. Therefore, data transmission to the 3615 may be faster when fewer of these control characters are used.

Another method of providing a more efficient data transfer is to use horizontal positioning, rather than blanks, to space between print fields on the 3615. Horizontal positioning requires fewer bytes to be transmitted than blanks for equivalent functions and does not cause message division unless it implies vertical positioning. For example, to print:

```
    001234      J. W. Customer
```

the following data string could be sent:

```
FO FO F1 F2 F3 F4 34 00 14 D1 4B E6...99
                  |________|
                           Horizontal
                           Positioning
```

Note: The data string ensures correct horizontal positioning.

Statistical Counters

The statistical counters reach a maximum of 255. The counters will either stop at 255, or they may be restarted, or "wrapped back," to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 3615:

Counter	Explanation
1	Loop error checks
2	Terminal address card unit checks
3	Printer check
4	End of forms
5	Not ready
6	Time-out
7	Halt on check
8	Forms emitter check

Controller Configuration Macro Instructions

Controller configuration macro instructions are used to specify the physical and logical configuration of each 4701 and its associated terminals. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration.

Translation Tables

The tables below are the standard output translation tables for the 3615. Characters that do not have corresponding hexadecimal values in these tables are nonstandard. To refer to these characters, you must either give each of them a unique hexadecimal value when it is specified in the OUTRTBL macro, or use a transparent write operation.

For 3615, 128-Character Set

<table>
<tr><th>Position</th><th>Hex Value</th><th>Character</th><th>Position</th><th>Hex Value</th><th>Character</th></tr>
<tr><td>0</td><td>F1</td><td>1</td><td>28</td><td>D5</td><td>N</td></tr>
<tr><td>1</td><td>F2</td><td>2</td><td>29</td><td>D6</td><td>O</td></tr>
<tr><td>2</td><td>F3</td><td>3</td><td>30</td><td>D7</td><td>P</td></tr>
<tr><td>3</td><td>F4</td><td>4</td><td>31</td><td>D8</td><td>Q</td></tr>
<tr><td>4</td><td>F5</td><td>5</td><td>32</td><td>D9</td><td>R</td></tr>
<tr><td>5</td><td>F6</td><td>6</td><td>33</td><td>60</td><td>-</td></tr>
<tr><td>6</td><td>F7</td><td>7</td><td>34</td><td>5B</td><td>$</td></tr>
<tr><td>7</td><td>F8</td><td>8</td><td>35</td><td>5C</td><td>*</td></tr>
<tr><td>8</td><td>F9</td><td>9</td><td>36</td><td>C1</td><td>A</td></tr>
<tr><td>9</td><td>F0</td><td>0</td><td>37</td><td>C2</td><td>B</td></tr>
<tr><td>10</td><td>7B</td><td>#</td><td>38</td><td>C3</td><td>C</td></tr>
<tr><td>11</td><td>7C</td><td>@</td><td>39</td><td>C4</td><td>D</td></tr>
<tr><td>12</td><td>61</td><td>/</td><td>40</td><td>C5</td><td>E</td></tr>
<tr><td>13</td><td>E2</td><td>S</td><td>41</td><td>C6</td><td>F</td></tr>
<tr><td>14</td><td>E3</td><td>T</td><td>42</td><td>C7</td><td>G</td></tr>
<tr><td>15</td><td>E4</td><td>U</td><td>43</td><td>C8</td><td>H</td></tr>
<tr><td>16</td><td>E5</td><td>V</td><td>44</td><td>C9</td><td>I</td></tr>
<tr><td>17</td><td>E6</td><td>W</td><td>45</td><td>4E</td><td>+</td></tr>
<tr><td>18</td><td>E7</td><td>X</td><td>46</td><td>4B</td><td>.</td></tr>
<tr><td>19</td><td>E8</td><td>Y</td><td>47</td><td></td><td>û</td></tr>
<tr><td>20</td><td>E9</td><td>Z</td><td>48</td><td>4D</td><td>(</td></tr>
<tr><td>21</td><td>4C</td><td><</td><td>49</td><td></td><td>ç</td></tr>
<tr><td>22</td><td>6B</td><td>,</td><td>50</td><td></td><td>[</td></tr>
<tr><td>23</td><td>6C</td><td>%</td><td>51</td><td></td><td>|</td></tr>
<tr><td>24</td><td>D1</td><td>J</td><td>52</td><td>5D</td><td>)</td></tr>
<tr><td>25</td><td>D2</td><td>K</td><td>53</td><td></td><td>]</td></tr>
<tr><td>26</td><td>D3</td><td>L</td><td>54</td><td>5E</td><td>;</td></tr>
<tr><td>27</td><td>D4</td><td>M</td><td>55</td><td></td><td>¬</td></tr>
</table>

For 3615, 128-Character Set (continued)

Position	*Hex Value*	*Character*
56		\
57	6D	_
58	6E	>
59	6F	?
60	7A	:
61	7E	=
62	7F	"
63	7D	'
64		¦
65	81	a
66	82	b
67	83	c
68	84	d
69	85	e
70	86	f
71	87	g
72	88	h
73	89	i
74		{
75		}
76		¨
77	91	j
78	92	k
79	93	l
80	94	m
81	95	n
82	96	o
83	97	p
84	98	q
85	99	r
86	50	&
87	79	`
88	A2	s
89	A3	t
90	A4	u
91	A5	v
92	A6	w
93	A7	x
94	A8	y
95	A9	z
96		
97		¢
98		!
99		ë
100		ï
101		Ü
102		â
103		ê
104		ü
105		ô
106		ˆ
107		è
108		é
109		ç
110		À
111		È
112		Ù
113		Ë
114		Ï
115		´
116		¸
117		î
118		Â
119		Ê
120		Î
121		Ĉ
122		Û
123		É
124		ù
125		à
126		~
127		

Note: With this standard translation table, referring to position 96 results in a space.

Chapter 18. IBM 3616 Passbook And Document Printer

The IBM 3616 Passbook and Document Printer (Figure 18-1) has two logically independent matrix print stations, document and journal, combined to form a single terminal. The document print station (DPS) handles several sizes of horizontal and vertical fold passbooks as well as cut forms requiring multi-line printing. The journal print station (JPS) allows the use of journal forms and cut forms requiring single-line validation printing. The terminal prints in either direction at 120 characters per second. Program selected character spacing is either 10 or 12 characters per inch. Indexing at the document print station occurs at program selected five or six lines per inch, and at approximately six lines per inch at the journal print station.

The 3616 has a basic set of 96 alphameric characters and up to 13 customer-defined special characters. The special character set can be Belgian, Brazilian, Portuguese, Danish/Norwegian, International, American English, United Kingdom English, Finnish/Swedish, Canadian French, French, German, Italian, Japanese (Katakana), Spanish, Spanish speaking, or Yugoslavian. Printing can be in a wide, bold manner at five or six characters per inch. Power can be 60 hertz (at 100, 110, 120, 200, 220, or 240 volts) or 50 hertz (at 100, 110, 200, 230, or 240 volts).

The document print station prints a line up to 100 characters long at 12 characters per inch or 83 characters long at 10 characters per inch. The journal print station prints a line up to 57 characters long at 12 characters per inch or 47 characters long at 10 characters per inch. The two print stations have separate, unique paper handling mechanisms with a common 8-wire matrix printing mechanism.

Controls and Indicators

Power On/Off Switch

This two-position rocker switch, with graphic symbols, turns terminal power on and off. It should be used with care, because turning power on and off at one terminal may cause errors at other terminals on the same loop.

Start Print Key

Two start print keys activate the print mechanism. For a nonshared terminal, pressing either key initiates platen closure and print transaction for a manually inserted passbook or cut form in the document print station, or validation slip in the journal print station. For a nonshared terminal set for autostart, pressing the passbook or cut form against the DPS stop bail initiates platen closure and print transaction. For a shared terminal with a start key required, pressing Start A or Start B initiates all printing transactions.

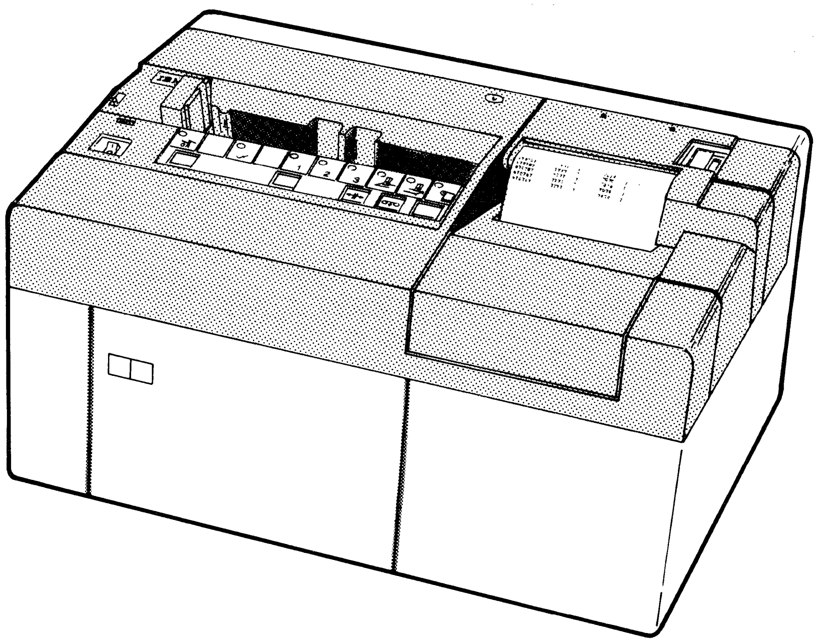


Figure 18-1. IBM 3616 Passbook and Document Printer

Stop Print Key

Pressing this key causes printing to stop at the end of the current print line and ejection of a document at the DPS. The JPS platen opens and paper advances one line. The DPS remains in its current state (cut-form or passbook).

Cut-Forms Insert

This switch can be pressed, before loading a long cut-form into the DPS, to select the first print line manually. An automatic 'Cutform Present' condition is set in 3616 control. The operator must position the document vertically.

Ribbon Change Control

Pressing this button moves the print head to the ribbon change access door. The operator opens the access door to change the ribbon. Note that power is still on after the door is opened. The operator should turn off the 3616 before changing the ribbon. Safety interlocks open when the print head is in the correct position.

Both the Stop Print key and the Ribbon key return the same status: interfaces A and B inactive for both stations, and the Stop Key status set. However, the Ribbon key moves the print head to the journal printer.

Journal Advance

This is a manually operated thumb-wheel control used to advance the journal take-up roll.

Test Switches

The two-position test switches determine which tests are to run. When the switches are off, no test is run. The switches are behind the front panel door.

Ready Indicator

This light provides the operator with loop status. When it is on, the loop and terminal loop adapter are operating satisfactorily. When the light is flashing, there is a failure between this terminal and the input loop adapter. When it is off, one of the following conditions exists.

- Terminal power is off.
- The indicator has failed.
- Terminal loop adapter has failed.
- The loop failed between the output loop adapter and this terminal.

Check

This light indicates an error in the printing. Such error conditions are:

- Translate check
- DEVPARM format check
- Overlength message check

This light is turned off when the error is corrected and an operation is started.

End-of-Journal-Roll Indicator

This light turns on when journal forms is reduced to less than 50.8 mm (2 inches). Install a new roll of paper and turn the 3616 off and on again to reset this condition.

Load DPS

This light turns on under program control to indicate a need to insert a document into the DPS.

Load JPS

This light turns on under program control when the transaction calls for a Validation print.

Indicators 1, 2, and 3

The functions of these indicators are user defined.

Special Features

- A special feature allows the document print station to accept a single-size vertical fold passbook.
- Two-part journal paper and tear bar.
- White face covers.

3616 Forms

The 3616 prints on passbooks, cut forms, and journal-roll forms with the following dimensions:

Horizontal Fold Passbooks

Width: 101.6 to 190.5 mm (4.00 to 7.50 in)

Height: 120.7 to 190.5 mm (4.75 to 7.50 in)

Thickness: 0.28 to 1.12 mm (.011 to .044 in)

Vertical Fold Passbooks

Width: 101.6 to 215.9 mm (4.0 to 8.5 in)

Height: 120.7 to 184.2 mm (4.75 to 7.25 in)

Thickness: 0.28 to 1.27 mm (.011 to .050 in)

Passbook Cut Forms

Width: 101.6 to 221.0 mm (4.0 to 8.70 in)

Height: 69.9 to 377.7 mm (2.75 to 14.87 in)

Thickness: 0.10 to 0.43 mm (.004 to .017 in)

Journal Roll Forms

Width: 12.70 @ 0.38 mm (5.00 @ 5.015 in)

Length: 12.24 m (50 ft.) single- or two-part forms. Maximum roll diameter is 61.0 mm (2.4 in).

Journal Cut Forms

Width: 13.97 to 30.48 mm (5.5 to 12.0 in)

Height: 69.85 mm to 152.4 mm (2.75 to 6.0 in)

Thickness: 0.10 to 0.25 mm (.044 to .010 in)

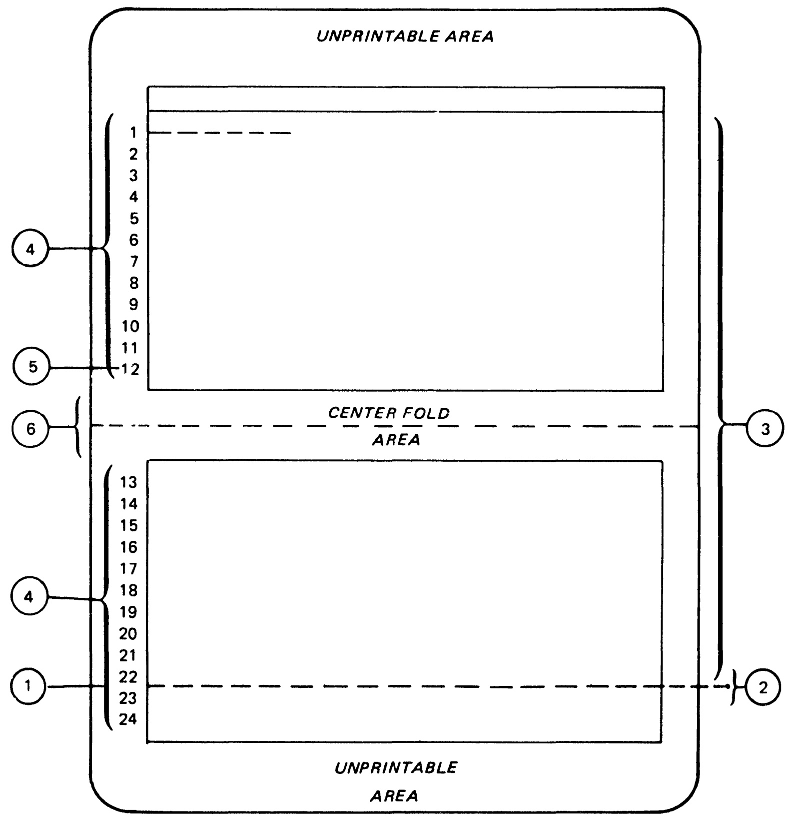
Detailed information concerning forms and paper stock requirements is given in *Forms Design Reference Guide for Printers*, GA24-3488.

Passbook Printing

The 3616 document print station can print on passbooks with the following dimensions. Margins must be at least 4.0 mm (0.156 in) from either edge of the passbook, at least 17.8 mm (.7 in) from the top of the cover or page, and 12.7 mm (0.500 in) above the bottom edge of the passbook. A vertical fold requires at least 4.75 mm (0.187 in) to the left and right of the center fold. A horizontal fold requires at least 6.35 mm (.25 in) above and below the center fold. All dimensions are either to the horizontal center lines or to the vertical center lines of characters.

Note: Page size, as defined in the DEV3616 configuration macro or the DEVPARM instruction, is the limit to which the printer allows a passbook to be indexed. However, if a passbook of smaller physical size than the passbook defined to the printer is inserted into the 3616, it becomes possible to index off the page, and feeding problems result.

When passbooks are being printed, four factors must be considered: (1) line finding, (2) page definition, (3) end of page, and (4) center-fold detection. A typical passbook is shown in Figure 18-2 .



Legend:
1. Clamping position
2. Offset steps to a physical print line, toward bottom of page
3. Offset lines to the first logical print line, toward top of page
4. Logical print lines that make up the page size
5. Center-fold begin
6. Center-fold skip (at least 6.35 mm [0.25 in.] from the center fold to the center of any character)

Figure 18-2. Parameters of a Passbook with a Horizontal Center Fold

Line Finding

This is the procedure used to find the next available line in a passbook where printing may start. The application program can find lines in two ways. The next available line number can be stored as part of a customer's account record in the central processor's data base, or the teller can determine the line number and enter it by means of the keyboard display.

Page Definition

A page is defined as the total area that is available on an open passbook. A horizontal center-fold passbook has unprintable areas above and below the center fold. A vertical fold passbook has unprintable areas to the left and right of the center fold.

End of Page

An end-of-page condition is reported to the application program by means of the condition code and status bits.

Center-Fold Detection

The last print line before a horizontal center fold and the number of lines to be skipped for the center fold must be specified during printer configuration by DEV3616, or during program execution using the DEVPARM instruction. These values must be specified correctly to avoid printing on the center fold. Vertical center-fold formatting is handled by the printer; characters must not be printed across a vertical center fold. The application program should use the center–fold device characteristics of the DEVPARM parameter (center-fold column number and number of columns to skip) to avoid printing across a vertical center fold; a skip of at least three characters must be specified.

Document Printing

The 3616 document print station can print on cut forms. The 3616 journal print station can print a one-line validation.

Cut Forms

Cut forms are inserted one at a time by the operator. An example of a cut form is a deposit slip. Forms printed at the DPS are printed using the cut forms parameters. Forms printed at the JPS are printed using the continuous forms parameters.

Journal Printing

The journal is a friction-feed mechanism that enables printing on rolled stock contained within the printer. The paper can be one or two parts; the original exits from the printer past a tear bar after printing.

The journal take-up provides the mechanism to store the single-part journal or the second part of the two-part printed journal on a take-up reel within the printer. The cover lock prevents access under the cover and access to the journal.

The journal printer can detect an end-of-forms condition by means of a switch on the printer that is set when the end of the continuous forms passes it. At least 102 mm (4 in) of paper remain when the switch is first set. Until the operator loads more paper (resetting the switch), intervention-required (out-of-forms) status is returned in SMSDST whenever the application program attempts a print operation.

Printing Modes

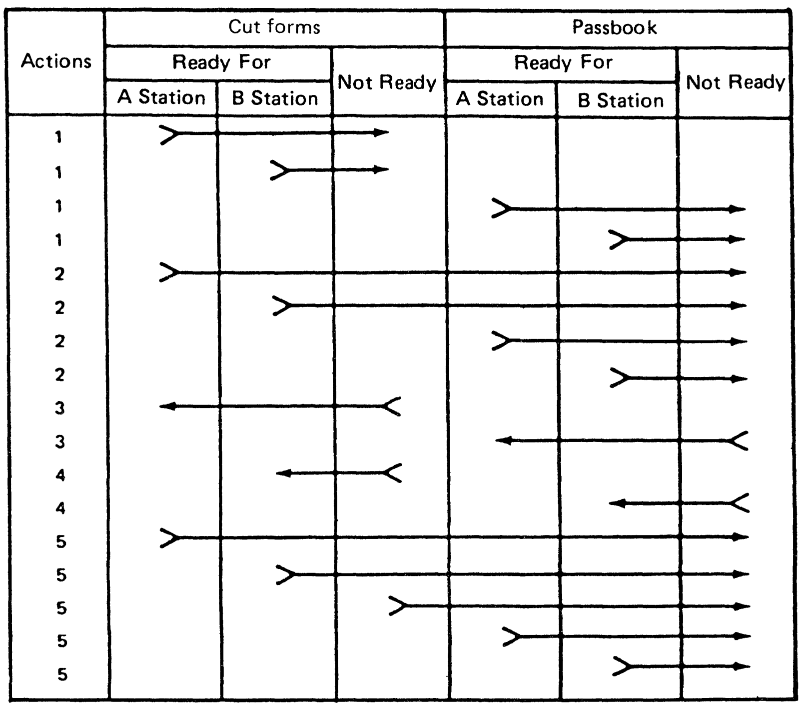
The printer operates in cut-forms, continuous-forms, or passbook mode, and may be in a ready or not-ready state. The application program can set the mode using the DEVPARM macro. The mode of the printer determines its initial state, and the state of the printer determines how the printer is started. Certain actions cause the state of the printer to change. Figure 18-3 shows the changes that can occur in the state of a printer and the reasons for the changes.

Cut-Forms Mode

In cut-forms mode, the printer is initially in a not-ready state. The operator must make it ready by inserting the document to be printed and pressing a Start key, unless the printer is in cut-forms automatic-start mode. The printer signals the application program that the printer is ready.

The document is released when:

- The application program writes a form-feed control character.
- The operator presses the Stop key.



Actions:

1. The STOP Print Key was pressed or an attempt was made to index past the defined end-of-page. The PASSBOOK or CUT FORM is ejected from the machine.
2. A form-feed control order was sent. The PASSBOOK or CUT FORM is ejected from the machine.
3. The START PRINT or START PRINT A Key was pressed.
4. The START PRINT B Key was pressed.
5. The cover was opened. The PASSBOOK or CUT FORM is ejected from the machine.

Figure 18-3. Actions That Change the State of the Document Printer

When a form-feed control character is detected, or an attempt is made to index past the defined end of the page, status is returned to the application program. Any of these actions returns the printer to the not-ready state.

After a X'0C' (forms-feed) command, printer reverts to passbook mode.

Continuous-Forms Mode

In continuous-forms mode, a printer is initially in a ready state. In this mode, the printer can operate without an operator (for example, to print teller or administrative reports). Figure 18-4 shows changes that can occur in the state of the journal printer and the reasons for the changes.

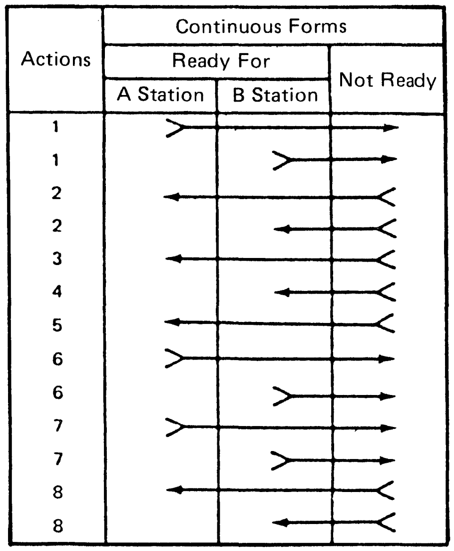
A printer in continuous-forms mode is put in a not-ready state if:

- The printer is stopped by pressing the Stop key or the cover is opened.
- A DEVPARM is issued requesting Start-key control.
- A form-feed control character is detected and the current device parameters indicate that a Start key is required.

A printer in continuous-forms mode in a not-ready state can be made ready by pressing a Start key (after replacing the cover, if it was removed).

Passbook Mode

With respect to its ready and not-ready states, a printer in passbook mode operates in a manner similar to one in cut-forms mode. Figure 18-3 shows the changes that can occur in the state of a passbook printer and the reasons for the changes.



Actions:
1. The STOP PRINT Key or the ribbon key was pressed, the cover was opened, or the printer is out of forms.
2. Any START PRINT Key was pressed while in autostart mode.
3. The START PRINT A Key was pressed while shared in start key required mode.
4. The START PRINT B Key was pressed while shared in start key required mode.
5. Any START PRINT Key was pressed while in start key required mode.
6. A form feed control order was sent while shared in start key required mode.
7. A DEVPARM was issued to change to start key required mode.
8. A DEVPARM was issued to change from start key required to autostart mode.

Figure 18-4. Actions That Change the State of the Journal Printer

Character Set

There is a 96-character set available for the document and journal printer. Characters are printed by a wire matrix print head at a speed of 120 characters per second. The characters are represented in translation tables within the printer.

Translating Data and Handling Control Characters

All data and control characters are sent directly from the application program to the printer; the controller does not translate the data.

Variable Parameters

A set of variable parameters describes the physical characteristics of the documents and passbooks to be printed. The parameters are specified during configuration. They can be changed during execution of the application program by the DEVPARM instruction.

Note: The SETRET=RETURN operand of DEVPARM is invalid for the 3616.

The variable parameters for JPS printing are:

- Forms type: A code designating the type of form to be printed. For the 3616, X'02' (continuous forms) or X'82' (continuous forms with concurrent sharing).
- Page size: The number of print lines on a page. Indexing past the last line on a page causes unit-exception status to be set for the application program; the form is positioned to column 1 of line 1 of the next page. This function can be bypassed by specifying a page size of 0. The maximum valid page size is 127 lines.
- Warning line: The number of any print line on a page. Indexing past this line causes unit-exception status to be set for the application program and the form to be positioned to the line following the warning line. This function can be bypassed by specifying a warning line or page size of 0. The application program can use the warning line to indicate when to print footings or when to skip to a new page.
- Line length: The number of character positions in a print line. The maximum line length for 10 characters per inch is 47. The maximum line length for 12 characters per inch is 57. If line length specified is 0, the maximum is assumed.
- Character density: The number of characters printed per 2.54 centimeters of print line, specified as 10 or 12.
- The JPS always performs the automatic new line function.
- Shared: The continuous-forms shared indicator allowing for sharing the JPS among application programs.
- Automatic Start: The indicator specifying whether the start key is required to begin printing. You cannot specify automatic start for a shared printer.

For passbook printing, the 3616 measures a passbook in print lines, and in steps within lines. The size of a step is constant, but the number of steps in a line depends on the density as selected during configuration or by the application program.

When a passbook is inserted in the 3616 document printer, it is clamped in a predetermined position though not necessarily on a print line. The printer then indexes, using the step and line offset values, to the correct position for printing. The 3616 calculates the number of steps to move the passbook.

The 3616 document printer spaces across a passbook center fold. You specify, during configuration, the type of center fold, the beginning line, and the skip count (in characters for a vertical fold passbook, and in lines for a horizontal fold passbook).

The variable parameters for DPS printing are:

- Forms type: A code designating the type of form being printed. X'00' indicates a passbook, X'01' indicates a cut form.

- Page size: The number of logical print lines on the passbook or cut form. The page size cannot be 0 or negative.

- Center-fold begin: The logical line number of the last line to be printed before the center fold (horizontal-fold books only) or the column number of the last character to be printed before the center fold (vertical-fold books only).

- Center-fold skip: The number of physical line indexes required to bypass a horizontal center fold or the number of columns to be skipped to bypass a vertical center fold. This number should not be less than 2.

- Offset steps: The number of steps (fractional lines), toward the bottom, a passbook or cut form must be indexed on insertion to move it from the clamping position to an integral print line. The maximum number is nine steps if a 5-mm line width (5 lines per inch) is specified; 11 steps if a 4-mm line width (6 lines per inch) is specified.

- Offset lines: The number of full lines a passbook or cut form must be indexed, toward the top of the document, to move it from the first integral line after clamping to the first logical print line. The maximum number is 60 lines for a 5-mm line width (5 lines per inch) or 72 lines for a 4-mm line width (6 lines per inch).

- Margin: The number of columns to be automatically skipped on a passbook from the leftmost physical print position. This number, plus the line length must not exceed the maximum number of print positions on the line.

- Line length: The number of character positions in a cut form or passbook print line. Maximum number is 83 positions if 10 characters per inch is specified or 100 positions if 12 characters per inch is specified. If line length specified is 0 for the passbook, the document printer skips lines without printing. Do not specify 0 for cut forms.

- Passbook fold: Passbook fold type specified as vertical or horizontal.

- Shared: The passbook or cut forms shared indicator allowing for application sharing of the DPS.

- Autostart: The passbook or cut-forms indicator specifying whether the start key is required to begin a print operation or only insertion of the document to perform the start key function, cannot be specified on a shared printer.

- Auto newline: The passbook or cut-forms indicator that specifies whether the printer wraps a print line or reports line length exceeded status if the line length is exceeded.

- Line density: The passbook or cut-forms indicator that specifies whether the 3616 prints 5-mm lines (5 lines per inch) or 4-mm lines (6 lines per inch).

- Character density: The passbook or cut-forms indicator that specifies whether printing is to occur at 10 characters per inch or 12 characters per inch.

- Data Chaining: For cut-form printing only, the overlap of data transfer operations with printing operations.

Controller Configuration for The 3616

Before the 3616 can be operated by a controller program, controller configuration macro instructions must be written to define the physical and logical configuration of all devices attached to the controller. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. Refer to Volume 6 for macro coding information and descriptions of the DEV3616 and other configuration macros.

Terminal Control Characters

Control characters are embedded in the data stream and indicate operations to be performed at a terminal. The following are the control characters for the 3616:

Function	Control Character	Explanation
Carriage Return	X'0D'	The logical print element is returned to the extreme left print position; the paper line is unchanged.
Line Feed	X'25'	The paper is spaced up one line; the print position is unchanged.
New Line	X'15'	The paper is spaced up one line and the logical print element returns to the extreme left print position.
Position	X'34'	This control character is followed by a flag byte and a 1-byte positioning value that control the repositioning of the print element and the paper. The flag byte has the following control: Bits 0-3 are ignored. Bit 4: 0 = Position is an absolute value. 1 = Position is relative to the present position. Bit 5: 0 = Position change is horizontal. 1 = Position change is vertical. Bits 6-7 of the flag byte are ignored. The positioning value indicates a line or column number. It contains an absolute value for use under the direction of the flag byte. A 0 results in no operation.

Function	Control Character	Explanation
Form Feed	X'0C'	For continuous forms, lines are skipped until paper is positioned to print the first line of a new page. For passbook and cut forms, the document is released. In cut-form mode, the DPS reverts to passbook mode.
Select User Defined Characters	X'0E'	Characters in the range X'B0' through X'BF' are found in the user-defined table.
Deselect User Defined Characters	X'0F'	Character selection returns to the base translation table.
Select Bold Print	X'11'	Print the following characters in bold print.
Deselect Bold Print	X'12'	Return print density to normal density.

Programming Considerations

The LWRITE, WRTI, SIGNAL, and DEVPARM actually send data to the 3616. You can perform all operations *without* the WAIT operand or subsequent LCHECK instruction. However, the application program must refrain from using the output area until operations requested with NOWAIT are complete.

To improve performance, use the following procedures:

- A form-feed control character should immediately follow the last character of the last line to be printed on a passbook or cut form. This inhibits indexing an extra line and eliminates the time associated with processing the form feed as a separate operation.
- A vertical positioning control character should be used, rather than multiple line feed control characters, to index multiple lines. This eliminates the time used issuing multiple control characters where one is sufficient.
- In cut-forms mode, use DEVPARM to specify data chaining, which overlaps printing with data transfer. Each LWRITE is posted complete before actual printing is complete, allowing one LWRITE to be transmitted to the printer while the previous LWRITE is still printing. To ensure that an LWRITE is completed, issue a form-feed command. To recover from an error, reprint the document.

A DEVPARM to the JPS causes a line space. You can override the automatic new-line operation on the previous LWRITE by using a no-operation control character such as X'0F12'.

The 3616 does not support DEVPARM macro instructions with the SETRET=RETURN operand coded.

Upon receipt of X'0201' status from a 3616, the application program should wait 10 seconds before retrying the command. Or, the application can issue an LCHECK for status 0808 to await completion of a power-on reset.

The X'34' command can produce unpredictable results. Be sure to use the correct 3-byte command information, and do not cross the warning line. Note that, following a X'34' command, the next line to be printed is the target line, not the warning line+1.

Residual Count: Status X'0101' can be returned for two reasons. In one case, the operator attempted to write more characters than defined in the line-length parameter of DEVPARM. The X'0101' status is returned with a residual count of the excess characters. The number of characters specified in DEVPARM are printed.

In the second case, X'0101' is returned when the 3616 tries to print over the right side of the document. The residual count is 0.

The application program can use the residual count to determine what message to send to the operator.

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or they may be restarted, or "wrapped back," to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 3616:

Counter	**Explanation**
1	Loop error check
2	Terminal address card check
3	Time-out
4	Terminal protocol violation
5-8	Unused
9	Power-on reset
10	Ready indicator
11	A interface activated
12	B interface activated
13	Device processor protocol violation
14	DEVPARM request rejected
15	Incorrect message length
16	Stop key pressed
17	Line length exceeded
18	Warning line
19	End of forms
20	Print emitter checks
21	Print wire checks
22	Translate checks
23	End of page
24	Not used

Status Bits

The status bits and their explanations are shown in Appendix D, "Terminal and Device Status Codes" on page D-1.

Translation Tables

The controller does not translate standard characters for the 3616. Data and control characters are sent directly from the application program to the printer.

This section describes the translation tables that are used with the 3616. The National Use special characters are selected via OUTBHDR and OUTSPEC macros. User-defined characters are selected via the OUTBHDR and CHARDEF macros.

For 3616 Standard Character Set

Hex Value	*Character*	*Hex Value*	*Character*
F1	1	98	q
F2	2	99	r
F3	3	A2	s
F4	4	A3	t
F5	5	A4	u
F6	6	A5	v
F7	7	A6	w
F8	8	A7	x
F9	9	A8	y
F0	0	A9	z
C1	A	40	
C2	B	42	â
C3	C	48	ç
C4	D	4B	.
C5	E	4C	<
C6	F	4D	(
C7	G	4E	+
C8	H	50	&
C9	I	52	ê
D1	J	53	ë
D2	K	56	î
D3	L	57	ï
D4	M	5C	*
D5	N	5D	)
D6	O	5E	;
D7	P	60	-
D8	Q	61	/
D9	R	62	Â
E2	S	64	À
E3	T	68	Ç
E4	U	6B	,
E5	V	6C	%
E6	W	6D	_
E7	X	6E	>
E8	Y	6F	?
E9	Z	71	É
81	a	72	Ê
82	b	73	Ë
83	c	74	È
84	d	76	Î
85	e	77	Ï
86	f	7A	:
87	g	7D	'
88	h	7E	=
89	i	7F	"
91	j	CB	ô
92	k	DB	û
93	l	DC	ü
94	m	EB	Ô
95	n	FB	Û
96	o	FC	Ü
97	p	FD	Ù
		FF	█

For 3616 Katakana Character Set

Hex Value	*Character*	*Hex Value*	*Character*
F1	1	8D	シ
F2	2	8E	ス
F3	3	8F	セ
F4	4	90	ソ
F5	5	91	タ
F6	6	92	チ
F7	7	93	ツ
F8	8	94	テ
F9	9	95	ト
F0	0	96	ナ
C1	A	97	ニ
C2	B	98	ヌ
C3	C	99	ネ
C4	D	9A	ノ
C5	E	9D	ハ
C6	F	9E	ヒ
C7	G	9F	フ
C8	H	A2	ヘ
C9	I	A3	ホ
D1	J	A4	マ
D2	K	A5	ミ
D3	L	A6	ム
D4	M	A7	メ
D5	N	A8	モ
D6	O	A9	ヤ
D7	P	AA	ユ
D8	Q	AC	ヨ
D9	R	AD	ラ
E2	S	AE	リ
E3	T	AF	ル
E4	U	BA	レ
E5	V	BB	ロ
E6	W	BC	ワ
E7	X	BD	ン
E8	Y	BE	゛
E9	Z	BF	•
81	ア	40	
82	イ	41	•
83	ウ	42	「
84	エ	43	」
85	オ	44	,
86	カ	45	.
87	キ	46	ヲ
88	ク	47	ァ
89	ケ	48	ィ
8A	コ	49	ゥ
8C	サ	4B	•

For 3616 Katakana Character Set (continued)

Hex Value	*Character*	*Hex Value*	*Character*
4C	<	5E	;
4D	(	60	–
4E	+	61	/
50	&	6B	,
51	ｴ	6C	%
52	ｵ	6D	_
53	ﾔ	6E	>
54	ｭ	6F	?
55	ｮ	7A	:
56	ｯ	7D	'
58	ｰ	7E	=
5C	*	7F	"
5D	)	FF	▮

For 3616 National Use Graphics Table 5a Special Character Sets

Country \ Hex Value	4A	4F	5A	5B	5F	6A	79	7B	7C	A1	C0	D0	E0
US	¢	|	!	$	¬	¦	`	#	@	~	{	}	\
International	[	!	]	$	^	¦	`	#	@	~	{	}	\
Austria/Germany	Ä	!	Ü	$	^	Ö	`	#	§	ß	ä	ü	ö
Belgium	[	!	]	$	^	ù	`	#	à	¨	é	è	ç
Brazil	É	!	$	Ç	^	ç	ã	Õ	Ã	~	õ	é	\
Denmark/Norway	#	!	¤	Å	^	ø	`	Æ	Ø	ü	æ	å	\
Finland/Sweden	§	!	¤	Å	^	ö	é	Ä	Ö	ü	ä	å	É
France	°	!	§	$	^	ù	`	£	à	¨	é	è	ç
Italy	°	!	é	$	^	ò	ù	£	§	ì	à	è	ç
Japan	£	|	!	¥	¬	¦	`	#	@	–	{	}	$
Portugal	[	!	]	$	^	õ	`	Ã	Õ	ç	ã	'	Ç
Spain	[	|	]	Pt	¬	ñ	`	Ñ	@	¨	{	}	\
Spanish Speaking	[	|	]	$	¬	ñ	`	Ñ	@	¨	{	}	\
United Kingdom	$	|	!	£	¬	¦	`	#	@	–	{	}	\
Canadian French	à	!	'	$	^	ù	`	#	@	¨	é	è	,
Katakana	£	|	!	¥	¬	¦	`	#	@	–	{	}	$
Yugoslavia	$		ć	č				š	ž				Đ

Chapter 19. IBM 3287 Printer Models 1 and 2

The IBM 3287 (Figure 19-1) is a compact, movable, tabletop printer available in two models. It is a bidirectional wire matrix printer that can print up to 132 characters per line at up to 80 characters per second (Model 1) or up to 120 characters per second (Model 2). The printer is attached to the controller via the Device Cluster Adapter (DCA) feature. Refer to the *IBM 3287 Printer Models 1 and 2 Component Description* for detailed information regarding the 3287. The 3287 prints only on continuous forms. Detailed information about the form dimensions and paper stock requirements for the 3287 are also in that Component Description.

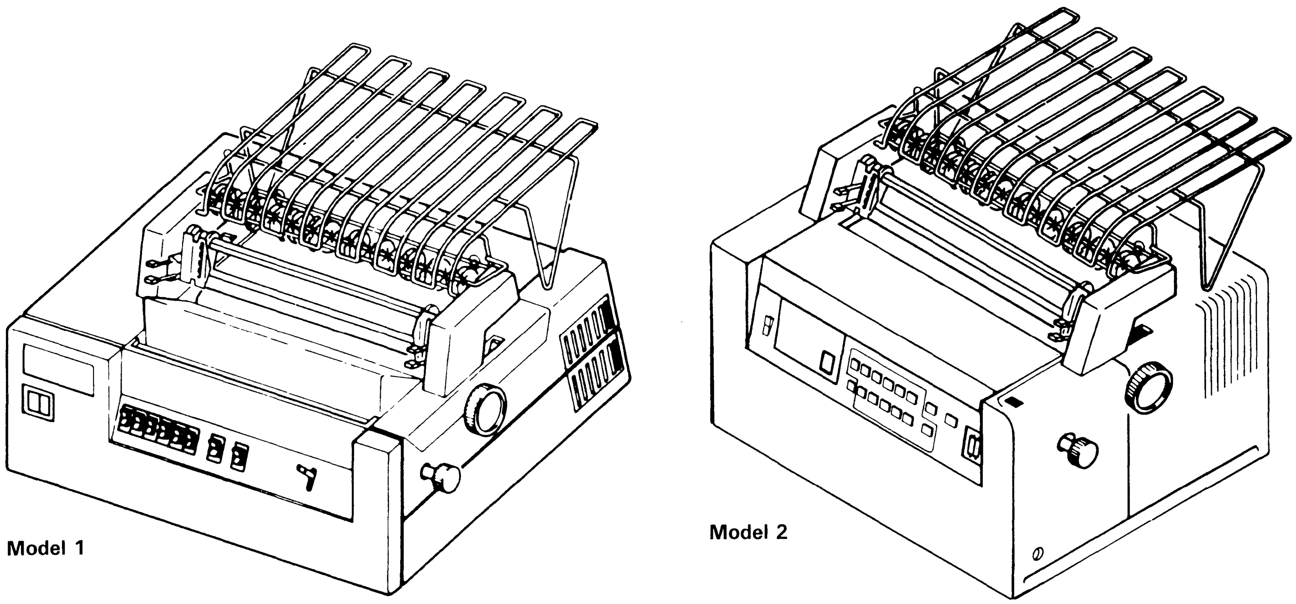


Figure 19-1. IBM 3287 Printer Models 1 and 2

Translating Data and Handling Control Characters

All data and control characters provided by the application program (using WRTI and LWRITE) are transmitted directly to the printer from the program-supplied area. The controller does not translate the data.

Variable Parameters

A set of variable parameters controls the print speed of the DCA-attached printers. You can change these parameters using the DEVPARM instruction. However, these parameters have no effect on the print speed of the 3287.

Forms Type: A code indicating continuous forms (X'02'). Concurrent sharing is not supported on the 3287.

Data Chaining: Allows printing at maximum rated speed, but has no effect on the print speed of the 3287.

Terminal Control Characters

Control characters are embedded in the data stream and indicate operations to be performed at a terminal. For the 3287, refer to the *3287 Printer Models 1 and 2 Component Description.*

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or they may be restarted, or "wrapped back," to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 3287:

Counter	Explanation
1-2	Reserved
3	Time-out
4-8	Not used
9	Power-on resets
10	Ready indications
11	Interface activated
12-13	Not used
14	DEVPARM request rejected
15	DCA not active
16	Operator intervention
17	Power off
18	Data check (parity)
19	No device on port
20	Equipment check
21	Device I/O error
22	Invalid Standard Character String (SCS) parameters
23	Invalid SCS control code
24	CANCEL key pressed
25	PA1 key pressed
26	PA2 key pressed
27	Eight (8) lines-per-inch key pressed
28	Six (6) lines-per-inch key pressed
29	Single-space key pressed
30	Double-space key pressed
31	Mono-case key pressed
32	Dual-case key pressed

Controller Configuration for The 3287

Before the 3287 can be operated by a controller program, controller configuration macro instructions must be written to define the physical and logical configuration of all devices attached to the controller. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. Refer to Volume 6 for macro coding information and descriptions of the DCA3287 and other configuration macros.

Translation Tables

There are no translation tables; data is sent directly from the application program to the printer.

Chapter 20. IBM 3262 Line Printer Models 3 and 13

The IBM 3262 Line Printer (Figure 20-1), available in two models, prints up to 132 characters per line. The 3262 Model 3 prints at up to 650 lines per minute; the 3262 Model 13 prints at up to 325 lines per minute. Both models of the 3262 attach to the controller through the Device Cluster Adapter (DCA) feature. For the 3262 printer to operate at its highest speed, the application program must specify data chaining with the DEVPARM instruction. For more information on the 3262 printers, refer to the appropriate publications listed in the *IBM System/370 Bibliography*, GC20-0001. Detailed information about forms and paper stock is also in the publications listed in that bibliography. The 3262 prints only on continuous forms.

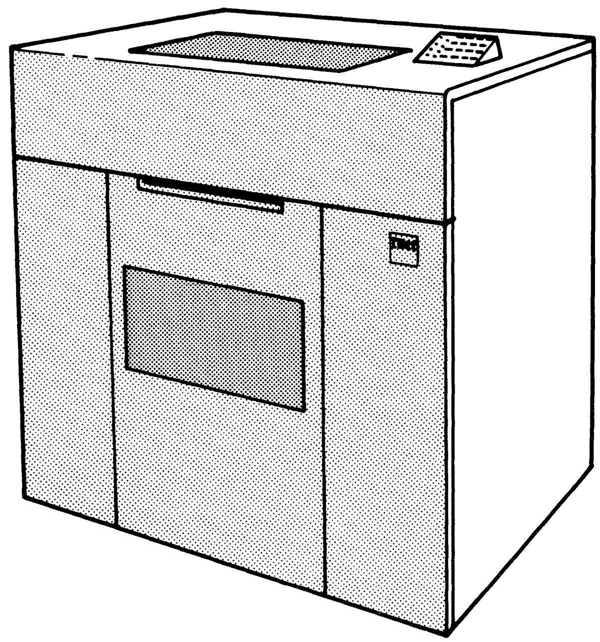


Figure 20-1. IBM 3262 Line Printer

Translating Data and Handling Control Characters

All data and control characters provided by the application program (using WRTI and LWRITE) are transmitted directly to the printer from the program-supplied area. The controller does not translate the data, and you do not specify translation tables.

Variable Parameters

A set of variable parameters controls the print speed of the DCA-attached printers. You can change these parameters using the DEVPARM instruction. However, these parameters have no effect on the print speed of the 3262.

Forms Type: A code indicating continuous forms (X'02'). Concurrent sharing is not supported on the 3262.

Data Chaining: Allows printing at maximum rated speed. Refer to "Programming Considerations" for information on specifying data chaining.

Terminal Control Characters

Control characters are embedded in the data stream and indicate operations to be performed at a terminal. For the 3262, refer to the *3262 Line Printer Models 3 and 13 Component Description.*

Programming Considerations

To permit the 3262 to operate at its highest speed, use DEVPARM to specify data chaining. When an application program prints data directly from the host to the 3262, the program should use DEVPARM to specify data chaining when the first-in-chain block is received. As data arrives from the host, the program issues a series of LWRITEs to print the first-in-chain block, and all the middle-in-chain blocks. When the last-in-chain block is received from the host, the program issues DEVPARM again to discontinue data chaining before writing the last-in-chain block to the printer. This ensures that printing will be complete when the LWRITE is complete.

When you specify data chaining for multiline LWRITEs, errors are reported as completion status on the LWRITE operation *after* the LWRITE in which the error actually occurred. Error recovery must begin with the data block before the one causing the LWRITE to report the error. For single-line LWRITEs with data chaining, error recovery must begin at the *second* block before the LWRITE reporting the error.

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or they may be restarted, or "wrapped back," to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 3262:

Counter	Explanation
1-2	Reserved
3	Time-out
4-8	Not used
9	Power-on resets
10	Ready indications
11	Interface activated
12-13	Not used
14	DEVPARM request rejected
15	DCA not active
16	Operator intervention
17	Power off
18	Data check (parity)
19	No device on port
20	Equipment check
21	Device I/O error
22	Invalid Standard Character String (SCS) parameters
23	Invalid SCS control code
24	CANCEL key pressed
25	PA1 key pressed
26	PA2 key pressed
27	Eight (8) lines-per-inch key pressed
28	Six (6) lines-per-inch key pressed
29	Single-space key pressed
30	Double-space key pressed
31	Mono-case key pressed
32	Dual-case key pressed

Controller Configuration for The 3262

Before the 3262 can be operated by a controller program, controller configuration macro instructions must be written to define the physical and logical configuration of all devices attached to the controller. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. Refer to *Controller Programming Library, Volume 6* for macro coding information and descriptions of the DCA3262 and other configuration macros.

Chapter 21. IBM 5210 Printer Models G01 and G02

The IBM 5210 Printers are desktop correspondence–quality printers for either hand–inserted cut forms or pin–fed continuous forms. Printing is in either proportional spaced or 10-, 12-, or 15-pitch lines ranging 132 to 198 characters long, depending on pitch. The line indexing speed is up to six inches per second. Print line density ranges 3.4 to 28 lines per inch, with half-line spacing for subscripts and superscripts. Depending on pitch and font, the maximum print speeds are 40 characters per second (cps) for Model G01, and 60 cps for Model G02. Both models of the 5210 attach to the controller through the Device Cluster Adapter (DCA) feature. For more information on the 5210 printers, refer to the appropriate publications listed in the *IBM System/370 Bibliography*, GC20-0001. Those 5210 references also describe the forms and paper stock needed for those printers.

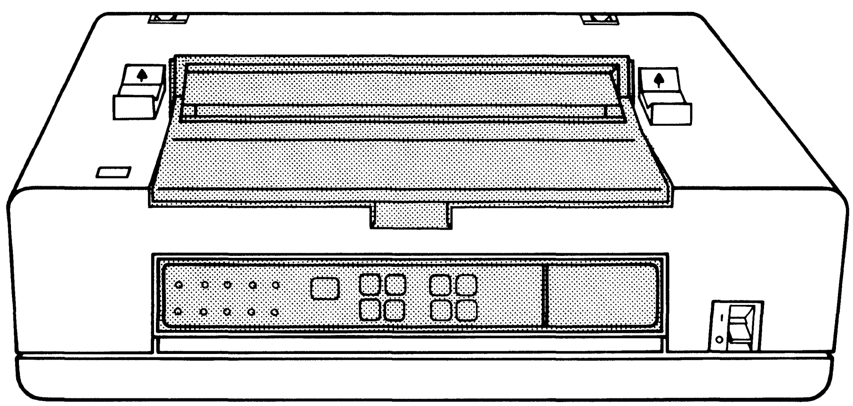


Figure 21-1. IBM 5210 Printer

Translating Data and Handling Control Characters

All data and control characters provided by the application program (using WRTI and LWRITE) are transmitted directly to the 5210 printer from the program's data area. The controller does not translate the data, and no translation tables have to be specified.

Variable Parameters

A set of variable parameters controls the print speed of the DCA-attached printers. You can change these parameters using the DEVPARM instruction.

Forms Type: A code indicating continuous forms (X'02'). Concurrent sharing is not supported on the 5210.

Data Chaining: Allows printing at maximum rated speed, but has no effect on the print speed of the 5210.

Terminal Control Characters

Control characters are embedded in the data stream and indicate operations to be performed at a printer. For the 5210, refer to the appropriate component description publications listed in the *IBM System/370 Bibliography*, GC20-0001.

Programming Considerations

The default settings for a 5210 after it is switched on are 10–pitch, six lines per inch.

The 4700 system does not support the data stream types allowing 5210 proportional spacing or drawer selection. When the 5210 attaches to the 4701 controller, programming should be identical to that for the DCA-attached 3287 printer.

Statistical Counters

The statistical counters reach a maximum of 255. The counters either stop at 255, or they may be restarted, or "wrapped back," to start counting at 128. The STATS parameter of the STARTGEN macro controls counter wrapping.

The following are the statistical counters for the 5210:

Counter	Explanation
1-2	Reserved
3	Time-out
4-8	Not used
9	Power-on resets
10	Ready indications
11	Interface activated
12-13	Not used
14	DEVPARM request rejected
15	DCA not active
16	Operator intervention
17	Power off
18	Data check (parity)
19	No device on port
20	Equipment check
21	Device I/O error
22	Invalid Standard Character String (SCS) parameters
23	Invalid SCS control code
24	CANCEL key pressed
25	PA1 key pressed
26	PA2 key pressed
27	Eight (8) lines-per-inch key pressed
28	Six (6) lines-per-inch key pressed
29	Single-space key pressed
30	Double-space key pressed
31	Mono-case key pressed
32	Dual-case key pressed

Controller Configuration for The 5210

Before the 5210 can be operated by a controller program, controller configuration macro instructions must be written to define the physical and logical configuration of all devices attached to the controller. When the configuration is performed, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration. Refer to Volume 6 for macro coding information and descriptions of the DCA5210 and other configuration macros.

Chapter 22. IBM Personal Computer and Personal Computer/XT

You can attach an IBM Personal Computer or Personal Computer/XT to the 4701 controller on the Device Cluster Adapter (DCA). The Personal Computer or Personal Computer/XT (referred to here as the PC or PC/XT) operates as a DCA–attached 3278 Model 2 (monochrome) or 3279 Model 2A (color) display. The PC or PC/XT, which can alternate between normal processing and DCA–attached terminal mode, must have the PC3278 control program installed.

When operating in terminal mode, the PC or PC/XT operates as described for a 3278 Model 2 in Chapter 9, "IBM 3278 Display Station Model 2." Refer to that information for additional operating details. For a more detailed description of the PC and PC/XT terminal mode features and requirements, refer to the *IBM Personal Computer 3278/79 Emulation Control Program: User's Guide.*

When processing PC or PC/XT keyboard scan codes, refer to the scan code table in Figure 22-1. These codes differ from the normal PC and PC/XT keyboard codes.

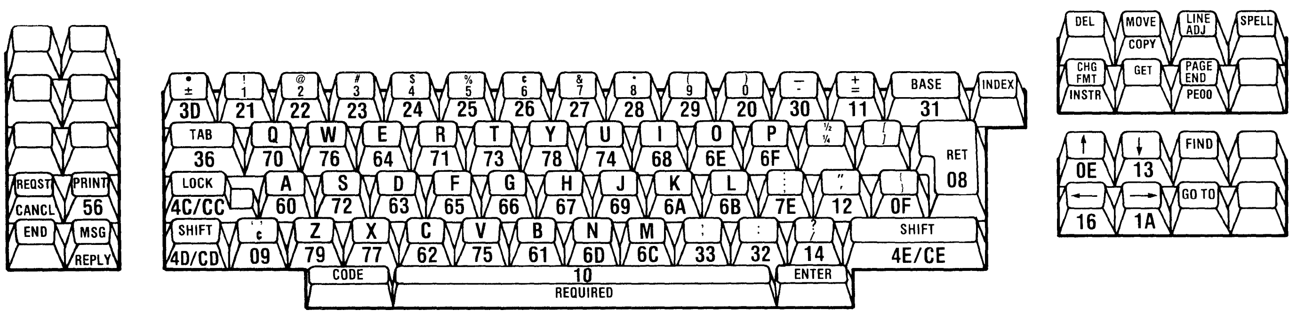


Figure 22-1. PC and PC/XT Scan Codes for 4700 Attachment

Chapter 23. IBM Displaywriter

With the 3270 Attached Work Station (3270 AW) Host Attachment control program installed, the IBM Displaywriter can be attached to the 4701 device cluster adapter (DCA) and operate as a 3278/3287 (display/printer) work station. If your 4700 system attaches to the Displaywriter/3270 AW, you should also refer to the following publications for that system:

1. *IBM Displaywriter 3270 AW Host Attach Programming Guide*, G544-2205, which gives detailed descriptions of the functions and differences described in this chapter. This manual also lists the other publications for the Displaywriter/3270 AW feature.

2. *IBM Displaywriter 3270 AW Operator's Guide*, S544-2202, which describes the differences in operation.

The rest of this chapter briefly summarizes the differences between 3278/3287 and Displaywriter/3270 AW features and functions.

Display and Keyboard Emulation

The Displaywriter/3270 AW attachment operates similarly to the 3278 Display Station, Model 2 as described in Chapter 9, "IBM 3278 Display Station Model 2." However, the following Displaywriter functions differ from normal 3278 operation:

Display Differences

Cursor display: Characters at the cursor position are displayed in high intensity.

Duplicate and Field Mark: The "*" and ";" are not combined with the "/" overstrike character.

Data Presentation: Screen data is presented a panel at a time rather than line by line.

Test Display: To display the 3278 test display issued by switching Normal/Test, press the Reset Display key. The Displaywriter Test key is the same function as the 3278 Test key.

Operator Information Line differs because of control unit differences, the ALA SNA/SDLC link, and logon status. Refer to the *IBM Displaywriter System, 3270 AW Operator's Guide*, S544-2202, for the symbols and their meanings.

The information display area indicates a connection to the 4700 by displaying "4700" in the Ready symbol position (left side).

Keyboard Differences

Keyboard Types Allowed: The Displaywriter/3270 AW supports the U.S. English 3278 87–key keyboard. Note that some keyboard scan codes, shown in Figure 23-1, differ from the normal Displaywriter scan codes when operating 3270 AW mode.

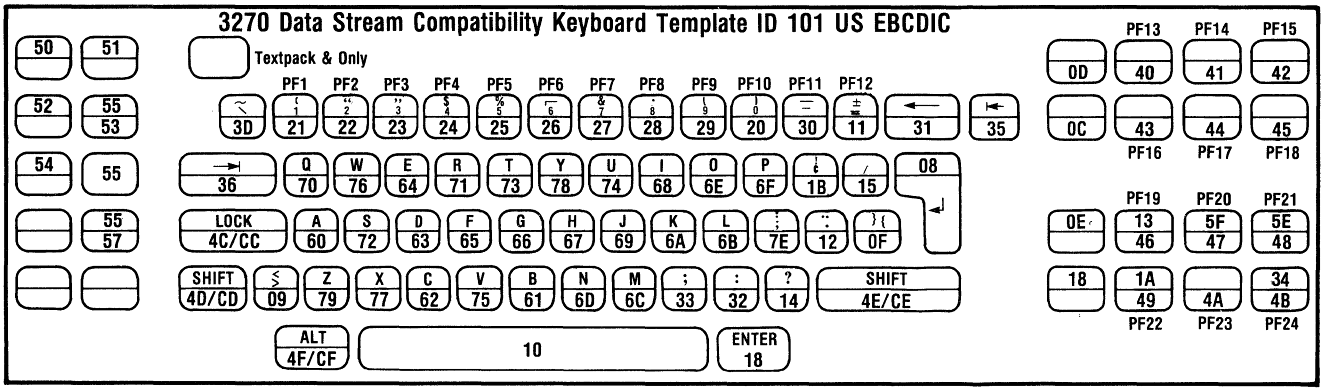


Figure 23-1. Displaywriter Scan Codes for 4700 Attachment

Disallowed National Keyboards: The following 3278 national keyboards are *not* supported by the Displaywriter/3270 AW:

- Brazilian/Portuguese
- Finnish
- International
- Japanese English, or Katakana
- World Trade EBCDIC

Disallowed 3278 Keyboard Functions: The following 3278 keys are not supported:

- Alternate cursor
- Audible key ("click")
- Cursor blink
- Entry assist
- Extended function keys (color, reverse video, and programmed symbols).

Disallowed Character Sets: The following 3274 character sets are not supported:

- Austrian/German alternate
- Brazilian
- Danish/Norwegian alternate
- English (US/ASCII)
- Finnish/Swedish alternate
- International (ASCII)
- Japanese English
- Portuguese
- Spanish alternate

Device Cancel (DEV CNCL): This key cancels a "save" or document transfer operation.

Keyboard Arrangement: The Displaywriter/3270 AW attachment has a special keyboard overlay mask, and some function keys have special functions. Refer to the *IBM Displaywriter 3270 AW Host Attachment Programming Guide*, G544-2205.

Typematic Keys: Some Displaywriter 3270 AW graphics and PF keys are typematic. Refer to the *Displaywriter/3270 AW Host Attachment Programming Guide*, G544-2205.

Numeric Lock: The Displaywriter/3270 AW keyboard does not automatically shift to lowercase mode when the cursor enters a field defined as numeric.

Reset Display: The Displaywriter/3270 AW uses this key to emulate the 3278 normal/test switch, for reconnecting the communication link, and to perform the power—on reset function.

Operating The 4700 System Monitor

You cannot operate the 4700 system monitor from the Displaywriter/3270 AW.

Other Unsupported 3278 Functions

The following general functions are not supported:

- 3270 extended data streams
- Magnetic stripe operations
- Single/multiple-case switch (A/A,a)
- Selector pen functions

Printer Emulation

The Displaywriter/3270 AW uses either a 5218/A01, 5218/A02, or 5228/A12 print-wheel printer to emulate 3287 print operation. The 5218/28 printers have print capability similar to the 3287 dot- matrix printer. They offer added 12–, 15–, and variable–pitch printing. However, many of the 3287 operator functions are different on the Displaywriter printers, and some functions (underscoring, programmed symbols, and so on) are not available.

Printer Differences

The following 3287 functions are emulated differently by the Displaywriter 5218/28 printers:

3287 Switch/Indicator Functions

Stop Switch: Pressing the 5218/28 Stop switch during a printing operation causes a delayed return of X'4001' intervention required printer status (a delay ranging 30 seconds to 10 minutes). You must enter a 5218/28 printer request to perform the 3287 Stop switch function. Pressing the 5218/28 Start switch after pressing the Stop switch has no effect on a pending LWRITE operation to the printer. You must press the Start switch again, after the LWRITE is issued.

Check Light Emulation: The Displaywriter printer Print Exception light emulates the 3287 Check light.

Status Indication: The two-digit display on the Displaywriter printer emulates the 3287 Status indicator.

Audible Alarm: Displaywriter/3270 AW replaces the 3287 audible alarm with a panel message and the display alarm.

Emulating DEVPARM and General Printer Control.

The Displaywriter/3270 AW emulates the DEVPARM–controlled 3287 functions using panel ("menu") options. Regardless of 4701 program control, the 5218/28 printer assumes cut-forms mode unless changed during the IPL process using the following procedure:

1. On the first TASK SELECTION menu, select option C (PROGRAM DISKETTE TASKS).

2. On the next menu, select option C (CHANGE PRINTER DESCRIPTION).

3. On the description panel, select option B, for the forms type.

4. On the forms panel, select option B, choice 3 (CONTINUOUS PAPER).

5. End the menu process by pressing Enter.

6. On the next panel, select option O (TASK SELECTION).

7. On the task selection panel, select FEATURE TASKS.

When executing a reverse DEVPARM command, the Displaywriter prints the requested parameter list on the attached 5218/28 printer. However, the operator must first press Stop, then Start to print the parameters.

Reporting End of Forms

A 5218/28 end of forms condition occurs as a Displaywriter screen message requesting you to load paper and restart the printer.

Warning: The Displaywriter/3270 AW *does not* return X'4001' (intervention required) status for end of forms. As a result, printing beyond the end of forms can occur causing loss of print data.

Deferred and Compressed Printing

The Displaywriter can buffer printer output data on diskette for later printing, and allows the operator to select variable–pitch printing. Refer to the *Displaywriter 3270 AW Host Attachment Programming Guide*, G544-2205, for specific information.

Unsupported 3287 Functions

The following functions are either not available or are not required when the Displaywriter/3270 AW is attached to the 4700 system:

- Intervention required (X'4001') status for end of forms (see "Reporting end of forms", above)
- Mono-case/dual-case switch and light
- Test switch functions
- Setup switch
- Index switch
- CU signal light
- Underscoring
- Programmed symbols
- Extended highlighting
- SNA character string support for structured fields and attribute processing
- "X" print error indicator.

Appendix A. Machine Instruction Formats

This appendix describes the machine instruction formats for the 4700 assembler instructions included in this volume. See *Volume 1 - General Controller Programming* for an explanation of the symbols used in this appendix.

ASSIGN

4C	0	S_2	D_2

0 8 12 16 31

ASSIGN

79	00	00	R_2	0

0 8 16 24 28 31

ASSIGN

79	10	00	R_2	0	D_2

0 8 16 24 28 32 47

DCACTL

4F	1	S_2	D_2

0 8 12 16 31

DCACTL

7A	27	00	R_2	0

0 8 16 24 28 31

DCACTL

7A	37	00	R_2	0	D_2

0 8 16 24 28 32 47

DEVPARM

4B	T	S_2	0	D_2

0 8 12 16 20 31

DEVPARM

7B	00	0	T	S_2	0	D_2
0	8	16	20	24	28	32 47

DEVPARM

7B	10	0	T	R_2	0
0	8	16	20	24	28 31

DEVPARM

7B	20	0	T	R_2	0	D_2
0	8	16	20	24	28	32 47

DPOOL

7A	4E	0	I_1	S_2	0	D_2
0	8	16	20	24	28	32 47

DPOOL

7A	5E	0	I_1	R_2	0
0	8	16	20	24	28 31

DPOOL

7A	6E	0	I_1	R_2	0	D_2
0	8	16	20	24	28	32 47

GETDMS

70	1	S_2
0	8	12 15

GETDMS

70	5	R_2
0	8	12 15

GETFLD

70	3	S_2
0	8	12 15

GETFLD

70	7	R_2
0	8	12 15

LCHECK (Loop Check-Terminal Device)

3A	T	0
0	8	12 15

LREAD (Loop or Device)

27	T	S_2
0	8	12 15

LREAD (Terminal-NOTRACK)

38	T	S_2
0	8	12 15

LREAD

78	64	0	T	S_2	0	D_2	L_2
0	8	16	20	24	28	32	48 63

LREAD

78	74	0	T	R_2	0
0	8	16	20	24	28 31

LREAD

78	84	0	T	R_2	0	D_2	L_2
0	8	16	20	24	28	32	48 63

LWRITE (Terminal)

39	T	S_2
0	8	12 15

LWRITE (Terminal)

3B	T	S_2	L_2	D_2	00
0	8	12	16	28	40 47

PUTDMS

70	0	S_2
0	8	12 15

PUTDMS

70	4	R_2
0	8	12 15

PUTFLD

70	2	S_2
0	8	12 15

PUTFLD

70	6	R_2
0	8	12 15

SIGNAL

48	T	S_2	L_2	D_2
0	8	12	16	20 31

SIGNAL

7B	03	0	I_1	S_2	L_2	D_2
0	8	16	20	24	28	32 47

SIGNAL

7B	13	0	I_1	R_2	0

0 8 16 20 24 28 31

SIGNAL

7B	23	0	I_1	R_2	L_2	D_2

0 8 16 20 24 28 32 47

STPLPS

4F	8	S_2	D_2

0 8 12 16 31

STPLPS

7A	23	00	R_2	0

0 8 16 24 28 31

STPLPS

7A	33	00	R_2	0	D_2

0 8 16 24 28 32 47

STRLPS

4F	0	S_2	D_2

0 8 12 16 31

STRLPS

7A	24	00	R_2	0

0 8 16 24 28 31

STRLPS

7A	34	00	R_2	0	D_2

0 8 16 24 28 32 47

SWAPTT

75	7	S_2	D_2

0 8 12 16 31

SWAPTT

79	05	00	R_2	0

0 8 16 24 28 31

SWAPTT

79	15	00	R_2	0	D_2

0 8 16 24 28 32 47

WRTI

3D	T	L	I_1	I_2

0 8 12 16 24 31

L - 1 or 2

Appendix B. 4700 Fields and Parameter Lists

This appendix gives detailed listings of the field and parameter lists created by the 4700 COPY assembler instruction for use with SNA/SDLC or BSC3 host communication. For definitions of other lists and fields created by the COPY instruction, refer to the COPY instruction description in Volume Two.

DEFDMS

```
DMSBEG      DEFxx   s,0,BDY=HALF BEGINNING OF DMS
DMSFLAG     DEFxx    DMSBEG,2    CONTROL BYTES
DMSFLG1     DEFxx    DMSFLAG,1   FIRST CONTROL BYTE
DMSFLG2     DEFxx   s,1          SECOND CONTROL BYTE
DMSSTAT     DEFxx   s,2          STATUS BYTES
DMSSTS1     DEFxx    DMSSTAT,1   FIRST STATUS BYTE
DMSSTS2     DEFxx   s,1          SECOND STATUS BYTE
DMSERCD     DEFxx   s,1          ERROR CODE BYTE
DMSEID      DEFxx   s,1          EOM/EID VALUE FROM SMSEID
DMSLSEQ     DEFxx   s,1          LINE SEQUENCE NUMBER
DMSDELM     DEFxx   s,1          USER DELIMITER FOR NATIVE CONTROL
DMSIML      DEFxx   s,2          VALUE FROM SMSIML
DMSFSEQ     DEFxx   s,2          FIELD SEQUENCE NUMBER
DMSFLG      DEFxx   s,2          DATSM WORK STATE FLAGS
DMSMOD      DEFxx   s,2          MODE FIELD
DMSHEND     DEFxx   s,0          END OF DMS HEADER FIELDS
DMSHSIZ     EQUATE (D:DMSHEND-D:DMSBEG)  SIZE OF THE HEADER AREA

*
*        END OF HEADER AREA
*
DMSPPF      DEFxx   s,2          PRESENTATION POSITION FIELD
DMSPPFR     DEFxx    DMSPPF,1    ROW POSITION
DMSPPFC     DEFxx   s,1          COLUMN POSITION
DMSACT      DEFxx   s,2          ACTUAL DEVICE DIMENSIONS
DMSACTR     DEFxx    DMSACT,1
DMSACTC     DEFxx   s,1
DMSACD      DEFxx   s,2          ACTUAL DEVICE COORDINATES
DMSACDR     DEFxx    DMSACD,1
DMSACDC     DEFxx   s,1
DMSBCD      DEFxx   s,2          IMAGE BUFFER COORDINATES
DMSBCDR     DEFxx    DMSBCD,1
DMSBCDC     DEFxx   s,1
DMSWIN      DEFxx   s,2          WINDOW DIMENSIONS (ROW,COL)
DMSWINR     DEFxx    DMSWIN,1
DMSWINC     DEFxx   s,1
DMSORG      DEFxx   s,2          ORIGINAL DEVICE DIMENSIONS
DMSORGR     DEFxx    DMSORG,1
DMSORGC     DEFxx   s,1
DMSRSA      DEFxx   s,1          RESERVED
DMSIBF      DEFxx   s,3          IMAGE BUFFER ADDRESS
DMSRSB      DEFxx   s,1          RESERVED
DMSFCT      DEFxx   s,3          FIELD CONTROL TABLE HEADER ADDR
DMSRS2      DEFxx   s,1          RESERVED
DMSDMS      DEFxx   s,3          DMS SELF LOCATE SECTION
DMSDMSEG    DEFxx    DMSDMS,1      SEGMENT NUMBER
DMSDMSPL    DEFxx   s,2            DISPLACEMENT
DMSLDA      DEFxx   s,1           DEVICE LDA
DMSRS3      DEFxx   s,3           RESERVED
DMSEND      DEFxx   s,0
DMSSIZ      EQUATE (D:DMSEND-D:DMSBEG)
DMSSECT     DEFxx    DMSBEG,DMSSIZ  FULL DMS DEFINED AS ONE FIELD
DMSHEAD     DEFxx    DMSBEG,DMSHSIZ FULL HDR DEFINED AS ONE FIELD
```

```
*
** PUTDMS CONTROL FLAG EQUATES
*
DMSPPFF  EQUATE X'8000'  PRESENTATION POSITION
DMSORGF  EQUATE X'4000'  ORIGINAL DEVICE SIZE
DMSACTF  EQUATE X'1000'  ACTUAL DEVICE SIZE
DMSACDF  EQUATE X'0400'  ACTUAL DEVICE COORDINATES
DMSBCDF  EQUATE X'0200'  IMAGE BUFFER COORDINATES FOR DISPLAY USE
DMSWINF  EQUATE X'0100'  WINDOW SIZE FOR DISPLAY USE

*
** GETFLD AND PUTFLD CONTROL FLAG EQUATES
*
DMSTYP   EQUATE    X'8000' DATASTREAM TYPE 0 = SCS, 1 = 3270
DMSIOT   EQUATE    X'4000' DATASTREAM BOUND  0 = DISP, 1 = PROCESSOR
DMSTIO   EQUATE    X'C000' DATASTREAM = 3270 PROCESSOR BOUND
*       00XX XXXX = SCS DISPLAY BOUND FLOW
*       01XX XXXX = SCS PROCESSOR BOUND FLOW
*       10XX XXXX = 3270 DISPLAY BOUND FLOW (INVALID FOR GETFLD)
*       11XX XXXX = 3270 PROCESSOR BOUND FLOW (INVALID FOR PUTFLD)
*

DMSSEG   EQUATE    X'0001' GETFLD SEGMENTS FIELDS LINE BY LINE
*
** GETFLD CONTROL FLAG EQUATES
*
DMSRMA   EQUATE    X'2000' READ MODIFIED ALL REQUEST
DMSDFO   EQUATE    X'1000' GET DATA ONLY FROM FIELD
DMSMVF   EQUATE    X'0800' GET ALL FIELDS OF REQUESTED CLASS
DMSMCF   EQUATE    X'0000' XXXX X000 = GET CURRENT FIELD
DMSMAD   EQUATE    X'0900' XXXX 1001 = GET ALL DISPLAY BOUND FIELDS
DMSMND   EQUATE    X'0100' XXXX 0001 = GET ONE DISPLAY BOUND FIELD
DMSMAP   EQUATE    X'0A00' XXXX 1010 = GET ALL PROCESSOR BOUND FIELDS
DMSMNP   EQUATE    X'0200' XXXX 0010 = GET ONE PROCESSOR BOUND FIELD
DMSMAE   EQUATE    X'0B00' XXXX 1011 = GET ALL USER FLAGGED FIELDS
DMSMNE   EQUATE    X'0300' XXXX 0011 = GET ONE USER FLAGGED FIELD
DMSMAU   EQUATE    X'0C00' XXXX 1100 = GET ALL UNPROTECTED FIELDS
DMSMNU   EQUATE    X'0400' XXXX 0100 = GET ONE UNPROTECTED FIELD
DMSMSQ   EQUATE    X'0500' XXXX X101 = GET FIELD OF SEQUENCE NUMBER
DMSMAL   EQUATE    X'0F00' XXXX 1111 = GET ALL FIELDS IN THE BUFFER
DMSMAA   EQUATE    X'0700' XXXX 0111 = GET ONE FIELD ANY TYPE
DMSGFB   EQUATE    X'0080' READ FULL BUFFER REQUIRED
DMSREI   EQUATE    X'0040' REISSUE GETFLD
DMSATT   EQUATE    X'0020' PROVIDE ATTRIBUTES FOR FIELD
DMSPRT   EQUATE    X'0010' OUTPUT IS FOR A PRINTER
DMSDEL   EQUATE    X'0008' INSERT USER DELIMITERS
DMSCAD   EQUATE    X'0004' GENERATE CURSOR ADDRESS FOR 3604
DMSPPC   EQUATE    X'0002' SET PRESENTATION POSITION TO THIS FIELD

*
** PUTFLD CONTROL FLAG EQUATES
*
DMSMDT   EQUATE    X'1000' TURN OFF MODIFIED DATA TAGS
DMSUSF   EQUATE    X'0800' TURN ON USER EXIT FLAG IN FLAGS BYTE
DMSSEQ   EQUATE    X'0400' USE SEQUENCE NUMBER TO FIND FIELD
DMSUDL   EQUATE    X'0200' DATASTREAM HAS USER DELIMITERS
DMSREP   EQUATE    X'0040' THIS IS A PUTFLD REISSUE
DMSSFP   EQUATE    X'0100' SET FIELD TO PROTECT
DMSFWT   EQUATE    X'2000' FORCE WRITE

*
** COMMON STATUS FLAG EQUATES
*
DMSMCR   EQUATE    X'0800' CURSOR POSITION OUT OF WINDOW
DMSERR   EQUATE    X'0200' AN EXCEPTION EXISTS.  CHECK THE ERROR CODE
DMSMPL   EQUATE    X'0100' THE OPERATION IS COMPLETED SUCCESSFULLY
DMSBOF   EQUATE    X'0010' BUFFER OVERFLOW
DMSFNF   EQUATE    X'0008' FIELD NOT FOUND
```

```
*
** GETFLD STATUS FLAG EQUATES
*
DMSNDS   EQUATE   X'8000' THE REQUESTED FIELD WAS NON-DISPLAYABLE
DMSUFL   EQUATE   X'0400' USER FLAG WAS ON FOR REQUESTED FIELD
DMSPRO   EQUATE   X'0002' THE REQUESTED FIELD WAS PROTECTED
DMSKJF   EQUATE   X'0001' THE REQUESTED FIELD WAS KANJI FIELD

*
** PUTFLD STATUS FLAG EQUATES
*
DMSRNT   EQUATE   X'8000' ISSUE LREAD KEYBOARD NOTRACK
DMSRTK   EQUATE   X'4000' ISSUE LREAD KEYBOARD TRACK
DMSRII   EQUATE   X'2000' REISSUE PUTFLD
DMSEOM   EQUATE   X'1000' EID RECD FROM KBD - PROCESSING REQUIRED
DMSHST   EQUATE   X'0800' ISSUE LWRITE TO HOST
DMSTRM   EQUATE   X'0400' ISSUE LWRITE TO DISPLAY
DMSRFB   EQUATE   X'0080' ISSUE GETFLD FOR FULL BUFFER READ
DMSUSP   EQUATE   X'0040' THIS FIELD HAD A USER FLAG TURNED ON
DMSRDM   EQUATE   X'0020' ISSUE GETFLD FOR PROCESSOR BOUND MODIFIED
DMSRML   EQUATE   X'0004' READ MODIFIED REQUEST RECEIVED FROM HOST

*
**  DATSM WORK STATE FLAGS
*
DMSFOR   EQUATE   X'4000' B'1' SCREEN FORMATTED, B'0' NOT FORMATTED
DMSUNP   EQUATE   X'2000' B'1' UNPROTECTED AREA ON SCREEN, B'0' NONE
DMSIRT   EQUATE   X'0200' B'1' INSERT MODE, B'0' NORMAL MODE

*
**  ERROR CODE EQUATES
*
DMSEC1   EQUATE   1       INVALID GETFLD REQUEST
DMSEC4   EQUATE   4       INVALID COMMAND OR ORDER IN DATA STREAM
DMSEC6   EQUATE   6       INVALID PUTFLD REISSUE
DMSEC7   EQUATE   7       NO PROTECTED FIELDS DEFINED IN IMAGE BUFFER
DMSEC8   EQUATE   8       DEVICE ERROR ON BUFFERED I/O
DMSEC10  EQUATE   10      STATION NOT DEFINED FOR DATSM USE
DMSEC11  EQUATE   11      RESERVED
DMSEC12  EQUATE   12      PUTDMS PARMETER EXCEEDS PERMITTED RANGE
DMSEC14  EQUATE   14      FIELD CONTROL TABLE OVERFLOW
DMSEC15  EQUATE   15      INVALID ADDRESS IN DATA STREAM

*
**  MODE FLAGS
*
DMSKJM   EQUATE   X'8000' KANJI (3270 EXTENDED) MODE
DMSEAB   EQUATE   X'4000' APL/EXTENDED ATTRIBUTE MODE
DMSBFM   EQUATE   X'2000' BUFFERED MODE
DMSNFF   EQUATE   X'1000' NO FORCED FIELD GEN FOR SCS
```

DEFPOL

```
* * *     'DPOOL' INSTRUCTION PARAMETER LIST DEFINITION

POLPAR   DEFxx   s,19        'DPOOL' PARAMETER LIST

*              DPOOL GET AND GETX PARAMETER LIST

POLGRS   DEFxx    POLPAR,1   RESERVED
POLGLA   DEFxx   s,1         LOGICAL DEVICE ADDRESS
POLGCI   DEFxx   s,8         COMPONENT IDENTIFIER
POLGCDT  DEFxx    POLGCI,2      ----- DEVICE TYPE
POLGCMN  DEFxx   s,3            ----- MODEL NUMBER
POLGCCT  DEFxx   s,1            ----- COMPONENT TYPE
POLGCUD  DEFxx   s,1            ----- USER DATA
POLGCOF  DEFxx   s,1            ----- OWNERSHIP FLAG
POLGPA   DEFxx   s,2         PHYSICAL DEVICE ADDRESS
POLGRC   DEFxx   s,1         RETURN CODE
POLGSP   DEFxx   s,6         RESERVED

*                 DPOOL RETURN PARAMETER LIST

POLRRS   DEFxx    POLPAR,1   RESERVED
POLRLA   DEFxx   s,1         LOGICAL DEVICE ADDRESS
POLRSS   DEFxx   s,10        RESERVED
POLRRC   DEFxx   s,1         RETURN CODE
POLRSP   DEFxx   s,6         RESERVED

*                  DPOOL QUERY PARAMETER LIST

POLQRS   DEFxx    POLPAR,1   RESERVED
POLQID   DEFxx   s,1         ID OF STATION WHOSE POOL IS
*                                     TO BE QUERIED
POLQIC   EQUATE    X'00'              --QUERY CURRENT STATION VALUE
POLQCI   DEFxx   s,8         COMPONENT IDENTIFIER
POLQCDT  DEFxx    POLQCI,2      ----- DEVICE TYPE
POLQCMN  DEFxx   s,3            ----- MODEL NUMBER
POLQCCT  DEFxx   s,1            ----- COMPONENT TYPE
POLQCUD  DEFxx   s,1            ----- USER DATA
POLQCOF  DEFxx   s,1            ----- OWNERSHIP FLAG
POLQPA   DEFxx   s,2         PHYSICAL DEVICE ADDRESS
POLQPB   EQUATE    X'0000'            --START AT BEGINNING OF DEV. POOL
POLQPE   EQUATE    X'FFFF'            --END OF DEVICE POOL WAS REACHED
POLQRC   DEFxx   s,1         RETURN CODE
POLQSI   DEFxx   s,1         STATION ID COMP. IS ASSIGNED TO
POLQNA   EQUATE    X'00'              --COMPONENT IS NOT ASSIGNED
POLQLA   DEFxx   s,1         LOGICAL DEVICE ADDRESS
POLQNR   DEFxx   s,1         NUMBER OF COMPONENTS ON TERMINAL
POLQUI   DEFxx   s,1         USE INDICATOR
POLQUA   EQUATE    X'00'              --ALL COMPONENTS OF TERM. AVAIL
POLQUS   EQUATE    X'FF'              --COMPONENT ASSIGNED SHARED TERM.
POLQSP   DEFxx   s,2         RESERVED
```

Appendix C. Program Check Codes

If the 4700 controller encounters an execution request that indicates a logic error, a program check results. The following are the hexadecimal codes and the explanations for possible program checks:

Code: **Explanation:**

01 Invalid segment specification: An operand specifies a segment that was not defined during controller configuration procedure, or segment 14 was specified in an instruction that will cause data to be stored or changed in segment 14.

02 Segment overflow: Completion of the instruction requires more storage than the specified segment provides.

03 (All instructions) Field length error:

1. An incorrect field was specified. The length is greater than 2 for an immediate operand.
2. A SETFPL instruction tried to set the field length indicator to a negative value.
3. A value is specified which, when added to the PFP, would be greater than the segment length.
4. The field length was greater than 255 for a PAKSEG instruction.

DATSM Instructions:

1. PUTDMS/GETDMS: The specified length for the DMS was less than the DMS size.
2. GETFLD: The data buffer is less than 50 bytes long.
3. PUTFLD: There are less than 50 bytes between the beginning of the data area and the end of the selected segment.

04 Return-address stack error: An LRETURN instruction was issued, but the return-address stack was empty; or a branch instruction was issued, but the stack was full.

06 Instruction count threshold: The number of instruction executions allowed per transaction has been exceeded.

08 No overlay name: The overlay name is not in the resident overlay directory.

09 Invalid operation or segment code: The instruction operation or segment selection code specified is invalid. Make sure that any required OPTMOD coding for the instruction was entered and that any parameter fields are properly coded.

Code:	Explanation:
0A	No entry point: There is no startup entry point specified.
0B	Instruction address error; an addressing error occurred. In the case of branch instructions, the program check address field of segment 1 will contain the address of the branch instruction. (DATSM instructions only): 1. GETDMS/GETFLD/PUTFLD: The field control table (FCT) was not initialized by a prior PUTDMS. 2. This station is not configured for DATSM operation.
0C	Instruction count exceeded: 65,535 instructions have been executed without a release of control.
0D	DEFDEL missing or incorrectly used: Either a delimiter request was made but no delimiter table was found or the table is not halfword aligned.
0E	EDIT mask error: The mask used with an EDIT instruction contains an error.
0F	Invalid link write control field: The link write control field or write options are invalid.
10	Communication link write length error: Data length exceeds 4095, data length during an LWRITE in batch mode was too long, command data length is incorrect; negative-response data length is incorrect, or there was a negative response to setting or testing sequence numbers.
11	Invalid parameter list, or parameter space is insufficient.
12	Indexing is not active.
20	Program check in called application program.
21	Called application program not found.
22	APCALL link stack full.
23	Recursive APCALL to an application program defined as USE=STATIC during configuration.
24	APCALL storage pool defined by MAXSTOR=was exceeded.
25	APCALL segment pool defined by MAXSEG=was exceeded.
26	APRETURN issued with no APCALL link stack entry - no calling application program.
27	Register address contains invalid segment space ID.

Code:	Explanation:
28	No transient pool: a transient pool was not defined for this station.
29	Transient application size error: the target transient application program will not fit in the largest transient area defined in the pool for this station.
FF	System error.

Appendix D. Terminal and Device Status Codes

The list below and table that follows contain information about two bytes of status bits that are set in SMSDST when an exceptional condition occurs (condition code=X'02'). The status bits in the first byte (SMSDS1) indicate the general condition:

Bits in SMSDS1	Condition
.......1 (X'01')	Incorrect length
......1. (X'02')	Unit check
.....1.. (X'04')	Command reject
....1... (X'08')	Attention
...1.... (X'10')	Prior operation
..1..... (X'20')	Data check
.1...... (X'40')	Unit exception
1....... (X'80')	Intervention required

The status bits in the second byte (together with those in the first) indicate the specific condition, as shown in the table.

The table applies to operations with terminals or terminal components; to use the table and list, find the status bits in the leftmost column of the table, the applicable instruction in the third column, and the applicable terminal or terminal component in the fourth column; read the explanation of the corresponding condition in the second column of the table.

The instruction has an appended letter code that indicates the component causing the exceptional condition. The codes are:

Code:	Meaning:
K	Keyboard
E	Encoder
D	Display
P	Printer

DCA-attached devices are either "DCA" followed by the component letter, such as DCA P for DCA-attached printers, or the model number of the device.

Note: Printers that attach to "3278 mode" DCA-compatible products do not return "DCA P" status. For example, the 4700 regards any status returned from the Displaywriter as DCA-attached display (DCA D) status, even though a Displaywriter-attached 5218 or 5228 printer caused the status condition.

Status Bits	Condition	Instruction	Terminal
.......1 .... (X'0100')	Incorrect length: The message length, as indicated by the first byte of the message, is less than the message length indicated in the LWRITE instruction. *Action:* Be sure that the controller application program and the implied message length are compatible.	LWRITE	3614/24
.......1 ....1 (X'0101')	Incorrect length: The message was longer than the space available in the segment (that is, the space between the PFP and the end of the segment). Or the message was longer than the value of the FLI when the FLI was nonzero and less than or equal to the length between the PFP and the end of the segment. *Action:* Change the segment so that enough space is available for the message. Or, if the end of the field was unexpected, change the FLI.	LREAD	DCA K 3604 K 3606 K 3608 K 3614 4704 K
	The data was longer than the maximum of 36 bytes that can be written to the magnetic stripe encoder. *Action:* Change the data so that its length is 36 bytes or less.	LWRITE LCHECK	3604 E 4704 E
	The data control characters for a print line caused the 3616 DPS, 4710 or 4720 cut-form, to attempt printing beyond the specified line length, the printer attempted to print off the form, the controlling program issued an LWRITE or DEVPARM with a data length of zero, or you requested bold printing on the 4720 with a defined line length of *1.* *Action:* Revise the data so it does not exceed the form width or is other than zero length, or change the specified form width with the DEVPARM instruction.	LWRITE LCHECK DEVPARM	3608 P 3616 DPS 4710 4720
	The data was longer than the maximum of 256 bytes that can be written to the 3614 terminal. Or a message in segment 14 was longer than the message length specified in the first byte of the message. *Action:* Change the data so that its length is 256 bytes or less. Or, change the message in segment 14 or the message length specified in the first byte.	LWRITE LCHECK	3614/24
.......1 ....0 (X'0200')	Unit Check: An error put the loop out of synchronization (possibly during a data transmission operation) and attempts to recover failed. The lights on the terminals should be off or flashing. A break in the loop probably caused the error. *Action:* Do not retry the operation for the current transaction. See *4700 System Problem Determination.*	DEVPARM LREAD LWRITE WRTI LCHECK SIGNAL	3604 K 3604 D 3604 E 3606 K 3606 D 3608 K 3608 D 4704 K 4704 D 4704 E 3608 P 3610 3611 3612 3614 3615 3616 4710

Figure D-1 (Part 1 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
. . . . . . .1 . . . . .0 (X'0200') (cont)	Unit Check: Either the Device Cluster Adapter (DCA) has been stopped or there was a DCA error. *Action:* If the DCA has been stopped, attempt to start the DCA with the DCACTL instruction or via the control operator. Otherwise, see *4700 System Problem Determination.*	LREAD LWRITE LCHECK	4720 DCA K DCA D DCA P 4704 E 4704 D 4704 K
	Unit Check: DCA error. *Action:* Do not retry the operation. See the *4700 System Problem Determination.*	DCACTL	DCA
. . . . . .1 . . . . .1 (X'0201')	Unit check: The controller received no response when it addressed a terminal address card (even though the loop was working). One of the following may have caused the condition: the device may have been turned off; the device's switches may have been set to the wrong address; more than one device may have been set to the same address; the device may have malfunctioned. *Action:* Do not retry the operation for the current transaction unless the device is a 3616 or 4710. See the *4700 System Problem Determination.* For the 3614, see the *3614 Programmer's Guide.*	DEVPARM LREAD LWRITE WRTI LCHECK SIGNAL	3604 K 3604 D 3604 E 3606 K 3606 D 3608 K 3608 D 3608 P 3610 3611 3612 3614/24 3615 3616 4704 K 4704 D 4704 E 4710
	Unit Check: The device may be powered off; the port may not have a device attached; the port may be stopped; there may be a hardware problem. *Action:* Be sure that a device is attached to the port and powered on. If the port had been stopped, attempt to start the port with the DCACTL instruction or via the control operator. Otherwise, see the *4700 System Problem Determination.*	LREAD LWRITE LCHECK	3262 DCA K DCA D 3287 4704 E 4704 D 4704 K
	Unit Check: DCA port error. *Action:* Do not retry the operation. See the *4700 System Problem Determination.*	DCACTL	DCA

Figure D-1 (Part 2 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
. . . . . . .1 . . . . . . .1 . (X'0202')	Unit check: An error occurred while enabling the magnetic stripe encoder or while encoding the magnetic stripe. If a stripe was encoded, the stripe may be incorrect. One of the following may have caused the condition: the encoder may have rejected the message, the stripe may have been passed through improperly, or the encoder may have malfunctioned. *Action:* Retry writing the message and passing the stripe through.	LCHECK	3604 E
	Unit check: The read operation did not complete in 1 minute and was ended by a timeout. This can occur by powering the terminal off/on after a message was keyed and the SEND key pressed. If the attention from the SEND key is received before the power off/on the system expects the device to be holding input, but the power off cleared the device buffers. *Action:* If attempting to read from a specific device, the operation may be retried until the message is received or until other status is posted. If the read was to service an attention, exit and wait for additional input.	LREAD	3606 K 3608 K
	Unit Check: An LWRITE was issued while the 4704 was still being initialized from the controller (IPL Mode). *Action:* Retry the operation when initialization is complete.	LWRITE LREAD	4704 D 4704 K
	Unit Check: The printer did not complete an operation in a specified time (that is, a time-out occurred), an emitter check occurred because the print wheel was not running at the proper speed, a print wire check occurred, or the platen opened unexpectedly. One of the following may have caused the condition: the printer may have been turned off and on; the STOP button may have been pressed; or the printer may have malfunctioned; for a 3611, Model 2, the last data character printed one position to the right of the last correct printing location as specified by the right sensor, and the following control character (carriage return) did not complete. If the STOP button was pressed while the printer was printing output, status X'8202' (X'0202' plus X'8000') is returned; some data may have been lost. *Action:* Do not retry the operation. Check the error counters to isolate the problem (See the *4700 System Problem Determination.*)	DEVPARM LCHECK LWRITE SIGNAL WRTI	3608 P 3610 3611 3612 3616 4710 4720
	Unit Check: The printer did not complete an operation in a specified time (that is, a time-out occurred); or the print belt did not come up to speed; or a device error, hammer fire check, or synch check occurred; or the printer overheated. One of the following may have caused the condition; the printer may have been turned off and on; the printer may have malfunctioned.	LCHECK	3615 3616 4710 4720

Figure D-1 (Part 3 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
. . . . . .1 . . . . . . .1 . (X'0202') (cont)	Unit Check: (cont) The data transfer on a physical read or write operation did not complete within 1 minute. *Action:* Ensure that the loop speed is the same as that specified by the BPS=operand of the LOOPS macro instruction, or check the statistical counters to isolate the problem. See the *4700 System Problem Determination.*	LREAD LWRITE	3614/24
. . . . . .1 . . . . . . .1 . (X'0204')	Unit Check: An input buffer overflow occurred while magnetic stripe data was being read. The keyboard is placed in purge mode to synchronize the input stream with the operator. This error can occur if data is being entered, but the AP read was delayed in being issued. It is not possible for the operator to continue entry from the point of overflow which is somewhere within the magnetic stripe data. The entire message should be re-entered. *Action:* The partial message received should be discarded and another read issued to get the whole message input.	LREAD	3604 K 4704 K
	Unit Check—Protocol Violation: A signal or data sequence between the printer and controller violated the defined communication protocol, or a hardware or loop error occurred. *Action:* Check the device statistical counters for hardware or loop errors. If the counters indicate a hardware error occurred, force a power-on reset sequence by opening and closing the front access door of the 3616 and 4720, or by switching power on and off on a 4710. If the counters indicate a loop error, retry the operation once again. If the retry fails, refer to the *4700 System Problem Determination.*	DEVPARM LWRITE LCHECK SIGNAL WRTI	3616 4710 4720
	Unit Check: A hardware or software buffer overflow occurred while PIN data was being received. The end-of-pin character may be lost. A terminal RESET is issued to clean up the input stream. This error may occur if the read is not issued for an excessive time after the PIN Keypad is enabled. *Action:* Prompt the operator, and reissue a read to get the message again.	LREAD	3604 K 4704 K
. . . . . .1 . . . . .1 . . . (X'0208')	Unit Check: While in PIN mode, the PIN indicator has been reset, disabling the PIN Keypad. This may have been caused by a SIGNAL instruction in the application program. A terminal reset is issued to clean up the input stream. *Action:* When PIN data is being received, only the hardware is permitted to terminate PIN mode. If a SIGNAL was issued by the application program, it should be removed and the application program reassembled.	LREAD	3604 K 4704 K

Figure D-1 (Part 4 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
. . . . . .1 . . . . .1 . . . (X'0208') (cont)	Unit Check—Incorrect Message Length: The controller tried to send the printer more data than the device was prepared to receive. *Action:* The device may have misread the data length because of a loop error. Retry the operation once. If retrying fails, refer to the *4700 System Problem Determination.*	DEVPARM LCHECK LWRITE SIGNAL WRTI	3616 4710 4720
	Device I/O Error: Repeated attempts to write to the printer have failed. *Action:* Do not retry the operation. See the *4700 System Problem Determination.*	LWRITE LCHECK	DCA P
. . . . . . .1 . . .1 . . . . (X'0210')	Equipment Check: One of these conditions has occurred: ● Thermal check ● Hammer-fire check ● Belt-synch check ● Repeatable parity error ● SCS feature not installed *Action:* Do not retry the operation.	LWRITE LCHECK	DCA P
	Unit Check: Magnetic stripe data was received while the terminal was still in PIN mode. The operator probably passed a magnetic stripe card in the reader before the PIN indicator on the terminal was turned off. A terminal RESET is issued to clean up the input stream. *Action:* Prompt the operator, and reissue the read to get the entire message entered again.	LREAD	3604 K 4704 K
. . . . .1 . . . . . . .1 . . (X'0404')	Command reject: The SMSTGU field is not set to a valid unit in the terminal group. *Action:* Verify that a valid address is being set in SMSTGU before an I/O operation, or that the address placed in SMSTGU on a non-specific read is not being cleared before the next write operation.	LWRITE WRTI	3606 D 3608 D 3608 P
. . . . .1 . . .1 . . . . . . (X'0440')	Command reject: No component was assigned to the logical device address specified for the operation. One of the following may have caused the condition: the logical device addresses may have been specified incorrectly during the controller configuration; the wrong logical device address may have been specified by the controller application program; the component may not have been assigned as expected or may have been reassigned (by means of an ASSIGN instruction or by the control operator.) *Action:* Check the controller configuration and the application program and make corrections if necessary.	LREAD LWRITE WRTI LCHECK SIGNAL DEVPARM	DCA K DCA D 3604 K 3604 D 3604 E 3606 K 3606 D 3608 K 3608 D 3608 P 3610 3611 3612 3614/24 3615 3616 DCA P 4704 K 4704 D 4704 E 4710 4720

Figure D-1 (Part 5 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
.....1.. 1....... (X'0480')	Command reject: Optional module was not loaded or an operation was requested that is invalid for the specified terminal (for example, a read operation from a write-only terminal). Or an invalid flag or parameter was specified for a DEVPARM instruction, DEVPARM incorrectly specified SETRET= RETURN, or a program tried to load a National Use table 5a that was not in the 3616, 4710, or 4720 configuration. One of the following may have caused the condition: the logical device addresses may have been specified incorrectly during the controller configuration; the wrong logical device address may have been specified by the controller application program; the component may not have been assigned as expected or may have been reassigned (by means of an ASSIGN instruction or by the control operator); or the parameter list for the DEVPARM instruction may have been incorrect. This status can also occur if you issue LWRITE or WRTI to a 4704-2/3 operating in local tracking mode.	LREAD	DCA K 3278 D 3604 D 3604 E 3608 P 3610 3611 3612 3614/24 3616 3262 3287 4704 K 4704 D 4704 E 4710
	Action: Check the controller configuration and the controller application program and make corrections if necessary. Or correct the parameter list if necessary.	LWRITE WRTI LCHECK	DCA K 3278 D 3604 K 3614/24 3616 DCA P 4704 K 4704 D 4710 4720
		DEVPARM	DCA D DCA K 3604 D 3604 E 3606 D 3608 D 3608 P 3614/24 3616 DCA P 3610 3611 3612 3615 4704 K 4704 D 4704 E 4710 4720

Figure D-1 (Part 6 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
.....1.. 1....... (X'0480') (cont)		DACTL	DCA
		SIGNAL	DCA P 3615 3616 4710 4720
.....1.. 1.....1 . (X'0481')	Command Reject: Port not 00-07 on enable/disable port. *Action:* Specify the correct port number.	DCACTL	DCA
.....1.. 1.....1 . (X'0482')	Command Reject: Port number not 00-07. *Action:* Specify the correct port number.	installation diskette only	DCA
.....1.. 1.....11 (X'0483')	Command Reject: Device type not 00-01. *Action:* Specify the correct device type.	installation diskette only	DCA
.....1.. 1....1 .. (X'0484')	Command Reject: Device currently assigned to station. *Action:* Make the device unassigned and retry the operation.	installation diskette cnly	DCA
.....1.. 1....1 .1 (X'0485')	Command Reject: No DCA devices on any port (invalid starter diskette). *Action:* See the *4700 System Problem Determination.*	installation diskette only	DCA
....1... ....0 (X'0800')	Attention: The operator signalled attention by pressing the reset key in succession. The operation was in a wait state with an undetermined end point (an attention does not affect a wait state with a determined end point). The wait state may have resulted from such conditions as: a read from a 3614/3624 terminal or the host; intervention required for a printer; failure to encode a magnetic stripe after the magnetic stripe encoder was enabled. *Action:* Prompt the operator to carry out the appropriate action (such as replacing the forms on a printer). Reset the magnetic stripe encoder if it was enabled.	LREAD	DCA K 3604 K 3614/24 4704 K
		LCHECK	3604 E 3610 3611 3612 3615 DCA P 4704 E 3616 4710 4720
		LWRITE WRTI	3610 3611 3615 DCA P 3616 4710 4720

Figure D-1 (Part 7 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
....1... ....1 (X'0801')	Attention: A write operation was attempted to a 3614 terminal that had data (status or a transaction) to be read. *Action:* Read the pending data before attempting another write operation.	LWRITE WRTI LCHECK	3614/24
	PA1 Switch: The PA1 switch was activated. *Action:* Defined by application program.	LWRITE LCHECK	DCA P
....1... ......1. (X'0802')	PA2 Switch: The PA2 switch was activated. *Action:* Defined by application program.	LWRITE LCHECK	DCA P
....1... .....1.. (X'0804')	Power Off: Power is turned off at the printer. *Action:* Set power on if required.	LWRITE	DCA P
....1... ....1... (X'0808')	Power On: The printer has completed its power-on/reset sequence: *Action:* Reissue LWRITE.	LWRITE WRTI DEVPARM SIGNAL	DCA P
	Power On—Attention: The 3616, 4710, or 4720 to which the operation was issued has been turned on and has completed its power-on reset sequence and has been initialized with the default configuration parameters. **Note:** This status is combined with the "prior operation-asynchronous" status, resulting in a X"1808" status code. *Action:* If the present device parameters are acceptable, reissue the instruction and continue; otherwise, use the DEVPARM instruction to set acceptable parameters.	DEVPARM LCHECK LWRITE SIGNAL WRTI	3616 4710 4720
....1... 1....... (X'0880')	Cancel Switch: The Cancel switch was depressed while printing; data was lost. *Action:* Defined by application program.	LWRITE LCHECK	DCA P
....1... .1...... (X'0840')	Printer Became Ready: A previously-reported intervention-required condition has been cleared and no application program was found waiting. (If a program is found waiting, this condition is not reported: the I/O is initiated). *Action:* Reissue the LWRITE.	LWRITE	DCA P

Figure D-1 (Part 8 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
...1.... .... (X'1000')	Prior operation: A previous read or write operation was not checked and was not completed successfully. The current operation was not attempted. The prior operation bit may be set in combination with any of the other status codes; the other bits indicate the status of the previous operation. For the 3616, 4710, 3262, and 3287, this bit also represents asynchronous status presented prior to the application initiated I/O. For 3616 or 4710, this status occurs only with "power on" status X"0808" as, X"1808." *Action:* Based on the other status bits, retry the previous operation or current operation or both.	LREAD	DCA K 3604 K 3606 K 3608 K 4704 K 3614/24
		LWRITE WRTI SIGNAL	DCA D DCA P 3604 D 3604 E 3606 D 3606 D 3608 D 3608 P 3610 3611 3612 3614/24 3615 3616 4704 D 4704 E 4710 4720
		DEVPARM	DCA P DCA D 3604 D 3604 E 3606 D 3608 D 3608 P 3610 3611 3612 3614/24 3615 3616 4704 D 4704 E 4710 4720

Figure D-1 (Part 9 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
..1..... .... (X'2000')	Data check: A character was not in the input translation table for the keyboard component or the output translation table or data/control stream for the display component. One of the following may have caused the condition: the translation table may have been specified incorrectly during the controller configuration; the wrong character may have been specified by the program; the address of the keyboard component may have been set incorrectly. *Action:* Check the translation table and character or the data/control stream and make corrections if necessary.	LREAD LCHECK	DCA K 3604 K 3606 K 3608 K 4704 K DCA D 3604 D 3606 D 3608 D 4704 D
	A character in a message for the magnetic stripe encoder was not a valid character (0-9, @, ', =) for the encoder. *Action:* Delete or replace the invalid character.	LWRITE WRTI	3604 E
	A character was invalid for the display component or printer (that is, an invalid control character or a data character not in the output translation table). The display component or printer displays or prints up to the invalid character and remains at that character position with the exception of the 3608 Printer. If an invalid character is contained in a 3608 message that is less than the average message length, as specified in the DEF3608 macro, no printing occurs. If the 3608 message length is greather than the average length, some printing may have occurred depending upon the location of the invalid character in the message. One of the following may have caused the condition: the translation table may have been specified incorrectly during the controller configuration; the wrong character may have been specified by the program, or the data contained an incomplete terminal control character. Device control 1 (X'11') occurred while performing quality printing on the 4720, an incomplete control character sequence was detected, the 4720 found a user-defined character while shifted in, or a regular character while shifted out. *Action:* Check the translation table and control or data character and make corrections if necessary.	LWRITE WRTI LCHECK	DCA D 3604 D 3606 D 3608 D 3608 P 3610 3611 3612 3615 3616 4704 D 4710 4720
..1..... ....1 (X'2001')	INVALID SCS Code: An invalid SCS control code is detected in an SCS character string. *Action:* Correct the program's use of the SCS control code.	LWRITE LCHECK	DCA P
..1..... ......1. (X'2002')	Invalid SCS Parameter: An invalid SCS control parameter is detected in the SCS data stream. *Action:* Correct the program's use of the SCS control parameter.	LWRITE LCHECK	DCA P
..1..... ....1... (X'2008')	Data Check—Parity Error: A parity error has been detected in the print buffer. *Action:* Reissue the LWRITE. If the problem persists, see the applicable *Component Description.*	WRITE LCHECK	DCA P

Figure D-1 (Part 10 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
.1...... .... (X'4000')	Unit exception: During a read with tracking operation from the keyboard component, the data could not be tracked on the display component. One of the following may have caused the condition: status from a previous write operation may have been pending; the display component may have malfunctioned. *Action:* Retry the read with tracking operation once.	LREAD	DCA K 3604 K 4704 K
.1...... ....1 (X'4001')	Unit exception: A read instruction was issued for a logical device which does not have message pending. Either the specific device requested has not entered a message, or if no specific device was requested, no device in the terminal group has a message entered. *Action:* Either exit the application program to wait on a message to present attention, repeat the read until other status is posted, or if polling logical devices, continue by reading the next in the list.	LREAD	3606 K 3608 K
	Operator Intervention Required: Operator intervention is required for one of the following conditions: • Cover or platen is open • End of forms • Paper jam • HOLD PRINT Left on for 10 minutes during printing. *Action:* Correct problem and reissue the LWRITE.	LWRITE LCHECK	DCA P
	The STOP key was pressed or a cover was opened while the printer was processing the subject instruction. *Action:* Find the residual count in SMSIML. If the count is zero (all information was processed), continue the operation. If the count is not zero (SMSIML=N), either cancel the operation or request the operator to press START to continue. Intervention required status continues to be reported on LWRITE/WRTI instructions until the START key is pressed.	DEVPARM LWRITE LCHECK SIGNAL WRTI	3616 4710 4720
.1...... ......1. (X'4002')	Dual Case Switch: The Dual Case switch was changed from MONO to DUAL case. *Action:* Defined by the controller application program.	LWRITE LCHECK	DCA P
..1..... .....1.. (X'4004')	6 LPI Switch: The LPI switch was changed from 8 to 6 lines per inch. *Action:* Defined by application program.	LWRITE LCHECK	DCA P
.1...... ....1... (X'4008')	8 LPI Switch: The LPI switch was changed from 6 to 8 lines per inch. *Action:* Defined by application program.	LWRITE LCHECK	DCA P

Figure D-1 (Part 11 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
.1. . . . . .1. . . . (X'4010')	Single Space Switch: The line spacing switch was changed from double spacing to single spacing. *Action:* Defined by application program.	LWRITE LCHECK	DCA P
.1. . . . . . . .1. . . . . (X'4020')	Mono Case Switch: The Mono Case Switch was changed from DUAL to MONO case. *Action:* Defined by the controller application program.	LWRITE LCHECK	DCA P
	Unit exception: The printer reached the logical end of page. For the 3610, 4710, and 3612, the cut form or passbook was released or ejected, or the continuous form was positioned at line 1 of the next page. The 3616 and 4720 do not eject a cut form or passbook. A multiple-line print or skip operation may have been truncated. *Action:* Defined by the controller application program.	LCHECK	3610 3611 3612 3615 3616 4710 4720
.1. . . . . . .1. . . . . . (X'4040')	Double Space Switch: The line spacing switch was changed from single to double spacing. *Action:* Defined by the application program.	LWRITE LCHECK	DCA P
.1. . . . . . 1. . . . . . . (X'4080')	Unit exception: The warning line was passed on the printer. The printer was positioned at the line after the warning line. A multiple-line print or skip operation may have been truncated. *Action:* Defined by the application program.	LCHECK LWRITE WRTI	3610 3612 3615 3616 4710 4720
1. . . . . .0 (X'8000')	Intervention required: The STOP PRINT key was pressed or the cover or throat was open (on any of the printers); the START PRINT key was not pressed (on the document printer with cut forms); either the START PRINT key was not pressed or the passbook was not inserted (on the passbook printer); a short form or no form was inserted; a form jam occurred (3608); the wrong START PRINT key was pressed (on a shared printer). This status may also be set after a write to a 3610 or 3612 in cut-forms mode, or a 3611 or 3612 in passbook mode, or a 3616, 4710, or 4720 is powered off or is not present on the loop. *Action:* Place the station in wait state (by reissuing the LWRITE instruction) until the operator intervenes appropriately.	LWRITE WRTI LCHECK	DCA P 3608 P 3610 3611 3612 3615 3616 4710 4720

Figure D-1 (Part 12 of 13). Status Bits for Terminals and Terminal Components

Status Bits	Condition	Instruction	Terminal
1. . . . . .1. . . . (X'8010')	Intervention required: The end of the continuous forms passed the End of Forms switch on the printer. *Action:* Place the station in wait state (by reissuing the LWRITE instruction) until the operator carries out the appropriate action. (Each time the START PRINT key on the 3610, 3612, 3616, or 4720 printer is pressed, one line is printed on the old forms; when the START PRINT key is pressed after inserting the new forms, normal printing resumes.) For cut forms or passbook printing, correct the program.	LCHECK	3610 3611 3612 3615 3616 4710 4720
		LWRITE WRTI	3616 4710 4720
	Intervention required: There was an attempt to print after end of page status was returned. *Action:* Correct the program.	LCHECK	3616 DPS 4710
1. . . . . .1. . . . . (X'8020')	Intervention required, unit exception: The printer tried to print past the end of the form or passbook. *Action:* Correct the program.	LWRITE LCHECK WRTI	4720
1. . . . . . . 1. . . . . .1 (X'8202')	Intervention required, unit check: The right edge of the document was sensed, and there was not enough room to print all of the data. The document is ejected and data is lost. *Action:* Correct the program.	LWRITE	3611-2

Figure D-1 (Part 13 of 13). Status Bits for Terminals and Terminal Components

Appendix E. Terminal and Device Statistical Counters

This appendix is a list of references to the terminal and device counter descriptions in each device description chapter.

The statistical counters record certain preselected events that occur on the loops, device cluster adapter (DCA), or on the devices themselves. For complete descriptions of the counters, how to read them with the system monitor, and their meanings refer to the *4700 Subsystem Operating Procedures*.

Device: ***Description Location:***

4704 Display Station, Model 1
"Statistical Counters" on page 6-10.

4704 Display Station, Models 2 and 3
"Statistical Counters" on page 7-12.

3604 Keyboard/Display
"Statistical Counters" on page 8-13.

3278 Display Station, Model 2
"Statistical Counters" on page 9-13.

3279 Display Station, Model 2
"Statistical Counters" on page 9-13, in Chapter 9, "IBM 3278 Display Station Model 2."

3606 and 3608 Financial Services Terminals
"Statistical Counters" on page 11-15.

4710 Receipt/Validation Printer
"Statistical Counters" on page 12-11.

4720 Forms/Passbook Printer, Models 1 – 4.
"Statistical Counters" on page 13-17.

3612 Passbook/Document Printer
"Statistical Counters" on page 16-15.

3610 Document Printer
"Statistical Counters" on page 16-15, in Chapter 16, "IBM 3612 Passbook and Document Printer."

3611 Passbook Printer
"Statistical Counters" on page 16-15, in Chapter 16, "IBM 3612 Passbook and Document Printer."

3615 Administrative Terminal Printer
"Statistical Counters" on page 17-11.

3616 Passbook/Document Printer
"Statistical Counters" on page 18-14.

3287 Printer, Models 1 and 2
"Statistical Counters" on page 19-2.

3262 Line Printer, Models 3 and 13
"Statistical Counters" on page 20-2.

5210 Printer, Models G01 and G02
"Statistical Counters" on page 21-2.

Personal Computer (PC and PC/XT)
"Statistical Counters" on page 9-13, in Chapter 9, "IBM 3278 Display Station Model 2."

Displaywriter
"Statistical Counters" on page 9-13, in Chapter 9, "IBM 3278 Display Station Model 2."

For counter descriptions of other attached devices or systems, refer to those device/system publications.

Appendix F. DATSM Sample Program and Error Codes

This appendix contains a sample DATSM application program, and defines the DATSM error codes that can be set in the DMSERCD field of the DATSM machine segment, DMS.

Sample Program

The following is a sample FCL program that illustrates the use of the DATSM facility. This sample program includes these steps:

1. Reads a data stream from the host.

2. Puts that data stream into the image buffer with the PUTFLD instruction.

3. Gets all fields from the image buffer with the GETFLD instruction.

4. Writes this output to the display.

5. Gets a current field from the image buffer, reads the keyboard to update that field, and returns the updated field to the image buffer.

6. Gets all fields from the image buffer, and sends these results to the host.

The sample application program is preceded by a skeleton program that illustrates the changes required or possible when a terminal is driven in terminal keytracking mode.

	Controller keytracking	**Terminal keytracking**
1	LREAD CP	LREAD CP
2		PUTDMS (LDA)
3	PUTFLD display-bd	PUTFLD (lock</unlock>)
4	GETFLD display-bd	<GETFLD display-bd>
5	LWRITE DS	
6	LREAD KB	
7	PUTFLD processor-bd	
8	GETFLD processor-bd	GETFLD
9	LWRITE CP	LWRITE CP

If an application is driving terminals in both controller and terminal keytracking mode, then the code may be left as in the right hand sketch, except that a PUTDMS must be inserted to cause the terminal to enter terminal keytracking mode and to communicate the LDA of the terminal to DATSM. The LDA will default to 1 if this value is not coded.

Also, in terminal keytracking mode DATSM may detect a device error. In this case, the status bit will be set X'2000' in the DMS status field, and the error field will contain the code X'08'. A test for this value should be added to the routine that handles error conditions.

```
                    TITLE 'SAMPLE APB'
**********************************************************************
*                     NOTES FOR SAMPLE APB                           *
*                                                                    *
*     PUTDMS MUST HAVE BEEN ISSUED BEFORE SAMPLE APB IS EXECUTED     *
*                                                                    *
*     ROUTINES FOR THE FOLLOWING LABELS HAVE NOT BEEN CODED IN       *
*     IN THIS EXAMPLE.                                               *
*                                                                    *
*   READHOST - READ DATA FROM HOST. BEFORE RETURN TO CALLER ROUTINE  *
*              WILL SET CONDITION CODE AS FOLLOWS:                   *
*               CC = 01 NO CHAIN - DATA ONLY                         *
*               CC = 02 FIRST IN CHAIN DATA                          *
*               CC = 04 LAST IN CHAIN DATA                           *
*               CC = 08 MIDDLE OF CHAIN                              *
*     WRHOST - WRITE CHAINED DATA TO HOST                            *
*     GETERR - STATUS OF 0200 RETURNED ON GETFLD                     *
*     PUTERR - STATUS OF 0200 RETURNED ON PUTFLD                     *
*    GETSTER - OTHER THAN TESTED FOR STATUS RETURNED ON GETFLD       *
*    PUTSTER - OTHER THAN TESTED FOR STATUS RETURNED ON PUTFLD       *
*     WRDERR - ERROR STATUS RETURNED ON WRITE TO DISPLAY             *
*     RDKERR - ERROR STATUS RETURNED ON READ KEYBOARD                *
*       EXIT - ROUTINE ENTERED AFTER ALL MODIFIED DATA SENT TO HOST  *
**********************************************************************
         LSPACE
*     EQUATES, CONSTANTS AND DEFLDS NEEDED FOR SAMPLE APB
         LSPACE
IOSEG    EQUATE    2
WKSEG    EQUATE    4
S4FLG2   DEFLD     4,,1               SEGMENTING FLAG BYTE
SGON     EQUATE    X'80'              SEGMENTING FLAG ON
SGOF     EQUATE    X'7F'              TURN OFF ABOVE FLAG
*
S4INSF   DEFLD     4,,2               INSERT MODE STATUS
S4INSRT  DEFLD     4,,2               INSERT MODE FLAG
S4INSWK  DEFLD     4,,2               INSERT MODE STATUS WORK FIELD
S4SIGC   DEFLD     4,,1               SIGNAL COMMAND OPERAND
S4SIGB   EQUATE    X'80'              TURN SIGNAL ON/OFF
S4SIG1   EQUATE    X'01'              USE INDICATOR LIGHT 1
*
DONLY    DEFCON    X'0110'            HOST WR CTL FIELDS - DATA ONLY
FINC     DEFCON    X'0D10'            HOST WR CTL FIELDS - FIRST IN CHAIN
MINC     DEFCON    X'0510'            HOST WR CTL FIELDS - MIDDLE OF CHAIN
LINC     DEFCON    X'0910'            HOST WR CTL FIELDS - LAST IN CHAIN
*
SWITCH   DEFLD     WKSEG,,1           BYTE USED FOR SWITCH
FSTON    EQUATE    X'80'              FIRST TIME SWITCH
FSTOFF   EQUATE    (FSTON-255)        TURN OFF FIRST TIME SWITCH
```

```
         LEJECT
         LSPACE
*********************************************************************
*                                                                   *
*        READ HOST AND PUT DATA RECEIVED INTO IMAGE BUFFER          *
*                                                                   *
*********************************************************************
         LSPACE
1    EXOR     DMSFLAG,DMSFLAG  MAKE SURE DMS CONTROL BYTES ZERO
         BRANL    READHOST        GO READ DATA FROM HOST
         JUMP     X'08',SAM010    JUMP IF MIDDLE OF CHAIN
         JUMP     X'02',SAM020    JUMP IF FIRST IN CHAIN
         JUMP     X'04',SAM030    JUMP IF LAST IN CHAIN
*            MUST BE DATA ONLY - NO CHAIN
         INORI    DMSFLAG,DMSTYP SET FOR 3270 DISPLAY BOUND
         BRANL    PUTRTN          GO TO PUTFLD SUBROUTINE
         JUMP     SAM100
*            MIDDLE OF CHAIN RECEIVED
SAM010   INORI    DMSFLAG,DMSREP SET CONTROL BYTE FOR PUTFLD REISSUE
         BRANL    PUTRTN          GO TO PUTFLD SUBROUTINE
         JUMP     1          GET HOST DATA
*            FIRST IN CHAIN RECEIVED
SAM020   INORI    DMSFLAG,DMSTYP SET FOR 3270 DISPLAY BOUND
         BRANL    PUTRTN          GO TO PUTFLD SUBROUTINE
         JUMP     1          GET HOST DATA
*            LAST IN CHAIN RECEIVED
SAM030   INORI    DMSFLAG,DMSREP SET REISSUE FLAG
         BRANL    PUTRTN          GO DO PUTFLD ON LAST FIELD
         JUMP     SAM100
*
*  PUTFLD SUBROUTINE
*
PUTRTN   LDFLD    1,SMSIML        IML WAS SET IN READ FROM HOST RTN
         SETFPL   IOSEG,0,(1)     SET FLI TO LGTH OF HOST DATA
         PUTFLD   IOSEG
         TSTMSKI  DMSSTAT,DMSMPL PUTFLD COMPLETED SUCCESSFULLY
         JUMP     ZO,SAM050       GO CHECK STATUS IF NOT
         LRETURN                  RETURN TO CALLER
SAM050   EXOR     SMSLSE,SMSLSE   ZERO LINK STACK COUNT
         TSTMSKI  DMSSTAT,DMSERR TEST IF EXCEPTION STATUS
         BRAN     MO,PUTERR       TO PUTFLD ERROR ROUTINE
         TSTMSKI  DMSSTAT,DMSRFB ISSUE GETFLD FOR FULL BUFFER READ
         JUMP     MO,SAM300       JUMP IF YES
         TSTMSKI  DMSSTAT,DMSRDM ISSUE GETFLD PROCESSOR BOUND MODIFIED
         JUMP     MO,SAM310       JUMP IF YES
         BRAN     PUTSTER         GO PROCESS OTHER STATUS
```

```
         LEJECT
         LSPACE
***********************************************************************
*                                                                     *
*      GETFLD ALL FIELDS FROM IMAGE BUFFER FOR DISPLAY WRITES         *
*                                                                     *
***********************************************************************
         LSPACE
SAM100   EXOR      DMSFLAG,DMSFLAG  MAKE SURE DMS CONTROL BYTES ZERO
         EXOR      DMSFSEQ,DMSFSEQ  MAKE SURE FIELD SEQ NUMBER ZERO
*   FLAG TO GET ALL FIELDS AND PROVIDE ATTRIBUTES FOR FIELD
MALATT   EQUATE    (DMSMAL+DMSATT)
         INORI     DMSFLAG,MALATT SET DMS CONTORL BYTES
SAM110   SETFPL    IOSEG,0,0      INIT SEG FOR FULL LGTH OF SEG
         GETFLD    IOSEG
         TSTMSKI   DMSSTAT,DMSMPL SUCCESSFUL COMPLETION
         JUMP      ZO,SAM120      JUMP IF NOT
         BRANL     WRDSP          GO DISPLAY DATA
         JUMP      SAM200         FINISH WITH DISPLAY WRITES
SAM120   TSTMSKI   DMSSTAT,DMSERR ERROR EXCEPTION
         BRAN      MO,GETERR      TO GETFLD ERROR ROUTINE
         TSTMSKI   DMSSTAT,DMSBOF BUFFER OVERFLOW CONDITION
         BRAN      MZ,GETSTER     TO ROUTINE TO HANDLE OTHER STATUS
*   BUFFER OVERFLOW CONDITION
         BRANL     WRDSP          GO DISPLAY DATA
         EXOR      DMSFLAG,DMSFLAG  ZERO DMS CONTROL BYTES
         INORI     DMSFLAG,DMSREI SET REISSUE GETFLD FLAG
         JUMP      SAM110         GO ISSUE GETFLD AGAIN
*
*  SUBROUTINE TO WRITE O/P TO DISPLAY
*
WRDSP    LWRITE    TDS,IOSEG      DISPLAY DATA
         JUMP      ST,WRDERR      JUMP IF ANY ERRORS
         LCHECK    TDS            WAIT FOR I/O COMPLETION
         JUMP      ST,WRDERR      JUMP IF ANY ERRORS
         LRETURN
*
*  SUBROUTINE TO READ I/P FROM KEYBOARD
*
RDKBD    LREAD     TKB,IOSEG,TRACK READ KEYBOARD FOR I/P DATA
RDKBD1   BRAN      ST,RDKERR      TO KEYBOARD READ ERROR ROUTINE
         MVFXD     DMSIML,SMSIML  SAVE I/P MESSAGE LGTH
         MVFXD     DMSEID,SMSEID  SAVE EID FIELD
         LRETURN
         LEJECT
         LSPACE
```

```
*******************************************************************
*                                                                 *
*    GET CURRENT FIELD, READ KEYBOARD AND RETURN TO IMAGE BUFFER  *
*                                                                 *
*******************************************************************
         LSPACE
SAM200   EXOR      DMSFLAG,DMSFLAG  ZERO DMS CONTROL BYTES
         INORI     DMSFLAG,DMSDFO GET CURRENT FIELD - DATA ONLY
         TSTMSKI   S4FLG2,SGON    TEST IF SEGMENTING
         JUMP      MZ,SAM205      IF NOT JUMP AROUND SETTING BIT
         INORI     DMSFLAG,DMSSEG SET ON SEGMENTING BIT
SAM205   SETFPL    IOSEG,0,0      SET FOR GETFLD - LGTH OF SEG
         GETFLD    IOSEG
         TSTMSKI   DMSSTAT,DMSERR TEST IF ANY ERRORS
         BRAN      MO,GETERR      TO GETFLD ERROR ROUTINE
         TSTMSKI   DMSSTAT,DMSNDS NON-DISPLAYABLE FIELD
         JUMP      MO,SAM240      IF YES DO NOTRACK READ
         JUMP      SAM250         IF NOT DO READ WITH TRACK
SAM210   EXOR      DMSFLAG,DMSFLAG  ZERO DMS CONTROL BYTE
         INORI     DMSFLAG,DMSIOT SCS PROCESSOR BOUND FLOW
         TSTMSKI   S4FLG2,SGON    TEST IF SEGMENTING
         JUMP      MZ,SAM220      IF NOT JUMP AROUND SETTING BIT
         INORI     DMSFLAG,DMSSEG SET ON SEGMENTING BIT
SAM220   PUTFLD    IOSEG          PFP AND FLI NOT CHANGED SINCE GETFLD
         MVFXD     S4INSWK,S4INSRT SET UP FIELD TO SEE
         AND       S4INSWK,DMSFLG  IF IN INSERT MODE OR NOT
         CCFXD     S4INSF,S4INSWK  SEE IF INSERT MODE CHANGED
         JUMP      EQ,SAM225       JUMP IF NO CHANGE
         MVFXD     S4INSF,S4INSWK  SAVE NEW INSERT STATUS
         EXORI     S4SIGC,S4SIGB   SWITCH ON/OFF BIT
         SIGNAL    TDS,S4SIGC,WAIT SEND OUT INDICATOR SIGNAL
SAM225   TSTMSKI   DMSSTAT,DMSMPL SUCCESSFUL COMPLETION
         JUMP      MO,SAM200      IF YES GO GET NEXT FIELD
         TSTMSKI   DMSSTAT,DMSERR EXCEPTION STATUS
         BRAN      MO,PUTERR      TO PUTFLD ERROR ROUTINE
         TSTMSKI   DMSSTAT,DMSTRM LWRITE TO DISPLAY REQUIRED
         JUMP      MO,SAM230      JUMP IF YES
         TSTMSKI   DMSSTAT,DMSRNT READ NOTRACK REQUIRED
         JUMP      MO,SAM240      JUMP IF YES
         TSTMSKI   DMSSTAT,DMSRTK READ WITH TRACK REQUIRED
         JUMP      MO,SAM250      JUMP IF YES
         TSTMSKI   DMSSTAT,DMSEOM EID RECEIVED - SEND TO HOST
         JUMP      MO,SAM310      ROUTINE TO SEND DATA TO HOST
         BRAN      PUTSTER        PROCESS OTHER STATUS
SAM230   BRANL     WRDSP          GO DISPLAY O/P
         TSTMSKI   DMSSTAT,DMSRII PUTFLD REISSUE REQUIRED
         JUMP      ZO,SAM200      IF NOT GO DO NEXT FIELD
         JUMP      SAM270         GO DO REISSUE OF PUTFLD
SAM240   LREAD     TKB,IOSEG,NOTRACK
         BRANL     RDKBD1         TO COMMON READ PROCESSING
         JUMP      SAM260         CONTINUE PROCESSING
SAM250   BRANL     RDKBD          GO READ KEYBOARD
SAM260   TSTMSKI   DMSSTAT,DMSRII TEST IF REISSUE REQUIRED
         JUMP      ZO,SAM210      LEAVE IF NOT
SAM270   EXOR      DMSFLAG,DMSFLAG ZERO OUT DMS CONTROL BYTES
         INORI     DMSFLAG,DMSREP TURN ON REISSUE FLAG
         JUMP      SAM220         GO REISSUE PUTFLD
         LEJECT
         LSPACE
```

```
*********************************************************************
*                                                                   *
*         GET ALL FIELDS FROM IMAGE BUFFER AND SEND TO HOST         *
*                                                                   *
*********************************************************************
         LSPACE
SAM300   EXOR     DMSFLAG,DMSFLAG  ZERO DMS CONTROL BYTES
         EXOR     DMSFSEQ,DMSFSEQ  MAKE SURE FIELD SEQ NUMBER ZERO
         INORI    DMSFLAG,DMSGFB READ FULL BUFFER REQUIRED FLAG
         JUMP     SAM320         TO COMMON PROCESSING
PROCBD   EQUATE   (DMSTIO+DMSMAP)  ALL 3270 PROCESSOR BOUND FIELDS
SAM310   EXOR     DMSFLAG,DMSFLAG  ZERO DMS CONTROL BYTES
         EXOR     DMSFSEQ,DMSFSEQ  MAKE SURE FIELD SEQ NUMBER ZERO
         INORI    DMSFLAG,PROCBD SET DMS CONTROL BYTES
SAM320   ANDI     SWITCH,FSTOFF  MAKE SURE 1ST TIME SWITCH OFF
SAM330   SETFPL   IOSEG,0,0      SET FOR FULL SEGMENT
         GETFLD   IOSEG          GET REQUIRED FIELDS
         TSTMSKI  DMSSTAT,DMSMPL NO STATUS STORED
         JUMP     MO,SAM370      JUMP IF YES
         TSTMSKI  DMSSTAT,DMSERR EXCEPTION STATUS RETURNED
         JUMP     MO,GETERR      LEAVE IF YES
         TSTMSKI  DMSSTAT,DMSBOF TEST IF BUFFER OVERFLOW
         JUMP     MO,SAM340      JUMP IF YES TO PROCESS
         TSTMSKI  DMSFLAG,DMSFNF TEST IF FIELD NOT FOUND RETURN
         JUMP     MO,SAM370      JUMP IF YES TO SEND TO HOST
         BRAN     GETSTER        IF NOT GO PROCESS OTHER STATUS
SAM340   TSTMSKI  SWITCH,FSTON   TEST IF FIRST TIME SW ON
         JUMP     ZO,SAM360      IF NOT DO FIRST IN CHAIN PROCESSING
*     PROCESS MIDDLE OF CHAIN
         MVFXD    SMSCWC,MINC    SET WR CTL FIELD FOR MIDDLE OF CHAIN
SAM350   BRANL    WRHOST         SEND DATA TO HOST
         EXOR     DMSFLAG,DMSFLAG  ZERO DMS CONTROL BYTES
         INORI    DMSFLAG,DMSREI SET ON REISSUE GETFLD FLAG
         JUMP     SAM330         GO REISSUE GETFLD
*     PROCESS FIRST IN CHAIN
SAM360   MVFXD    SMSCWC,FINC    SET WR CTL FIELD FOR FIRST IN CHAIN
         INORI    SWITCH,FSTON   TURN ON FIRST TIME SWITCH
         JUMP     SAM350         TO COMMON PROCESSING
SAM370   TSTMSKI  SWITCH,FSTON   FIRST TIME SWITCH ON
         JUMP     ZO,SAM390      IF NOT MUST BE NO CHAIN - DATA ONLY
*     PROCESS LAST IN CHAIN
         MVFXD    SMSCWC,LINC    SET WR CTL FIELD FOR LAST IN CHAIN
SAM380   BRANL    WRHOST         SEND TO HOST
         BRAN     EXIT           PROCESSING COMPLETE - EXIT
*     PROCESS DATA ONLY - NO CHAIN
SAM390   MVFXD    SMSCWC,DONLY   SET WR CTL FIELD FOR DATA ONLY
         JUMP     SAM380         TO COMMON END PROCESSING
```

| DMSERCD Error Codes

Each DATSM instruction can, under certain conditions, set a one-byte error code in the DMSERCD field of the DATSM machine segment. The following table defines those error codes. To determine which codes are set by which DATSM instructions, refer to the individual instruction descriptions in "DATSM Instructions" on page 5-43.

ERROR CODE	MEANING
X'01'	*Cause:* Invalid GETFLD or PUTFLD request (invalid bit combination). *Action:* Correct the application program.
X'02'	*Cause:* Invalid use of 'windowing' in Kanji mode. *Action:* Correct the application program.
X'03'	*Cause:* Kanji field was defined on an uneven boundary. *Action:* Redefine the field on an even boundary.
X'04'	*Cause:* Invalid data stream command or order. *Action:* 1. Correct the data stream at the source, or 2. Translate invalid characters to valid characters with LTRT before sending the data stream to DATSM.
X'05'	*Cause:* The required system buffer was not defined, or the station has no dedicated image buffer. *Action:* Correct the configuration to include either a system or image buffer for the station.
X'06'	*Cause:* GETFLD or PUTFLD 'reissue' error. The reissue control flag is set, but no reissue status flag is set, and the instruction was not within a data stream when it was reissued. *Action:* Correct the application program.
X'07'	*Cause:* PUTFLD cannot operate on the image buffer because all buffer fields are protected. *Action:* Correct the application program.
X'08'	*Cause:* An error occurred during I/O operation to a buffered terminal. *Action:* The program should test SMSDST for the error.
X'09'	(reserved)

Figure F-1 (Part 1 of 2). DATSM Error Codes in DMSERCD

ERROR CODE	MEANING
X'0A'	*Cause:* This station is not configured for DATSM. *Action:* Refer to "Using DATSM" in the chapter titled, "Processing 3270 Data Streams".
X'0B'	*Cause:* The program attempted to perform a DATSM operation using the system image buffer, and the terminal was not in local keytracking mode. *Action:* Change the terminal mode to local keytracking, or reconfigure the station to have its own image buffer.
X'0C'	*Cause:* The image buffer coordinate added to the window size is greater than the image buffer size. *Action:* Rewrite your program to avoid exceeding the image buffer.
X'0D'	*Cause:* The translation table for a buffered device was incorrectly defined as local keytracking mode. *Action:* Reconfigure the OUTRTBL macro to specify DEVICE=4704M2.
X'0E'	*Cause:* The field control table (FCT) is too small for the number of data stream fields. *Action:* Reconfigure the STATION macro to specify a larger FCT size.
X'0F'	*Cause:* The data field address in the data stream is outside the image buffer. *Action:* Correct the data stream, or redefine the image buffer size (DSM= operand of the STATION macro).
X'10'	*Cause:* The EID is invalid for local keytracking mode. No keyboard emulation should be done in local keytracking mode. *Action:* Find and remove the cause of the EID.
X'11'	*Cause:* The terminal changed between local and controller keytracking mode sometime after the initializing PUTFLD was issued to the buffered device, but before the last PUTFLD or GETFLD for the same data stream was issued. *Action:* Either remove the mode change, or make it after all non-initializing PUTFLDs and GETFLDs are ended.

Figure F-1 (Part 2 of 2). DATSM Error Codes in DMSERCD

Glossary

This glossary defines terms found in this volume of the *4700 Programming Library*; it elaborates on or qualifies the basic definitions found in general glossaries for the 4700 system and devices. For other definitions, the reader should refer first to the appropriate section of this manual, then to either the *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699, or to the prerequisite SNA publications listed at the beginning of this manual.

alphameric characters. In 4700 assembler language, the characters A/a through Z/z, the digits 0 through 9, and the characters #,$,@.

application program. (1) In 4700, a program written for or by a user that processes a transaction or performs some other financially–related work. (2) In communications, a program used to connect and communicate with stations in a network, allowing users to perform application– oriented activities.

assembler. (1) A computer program that converts symbolic instructions into machine instructions. (2) In a 4700 system, a VS version of an assembler used to: convert a program written in assembler language into machine instructions that can be executed in the controller; to convert 4700 configuration macro instructions into configuration data; and to convert customization macro instructions into customization data.

asynchronous. Without regular time relationship; unexpected or unpredictable with respect to the execution of a program or its instructions.

asynchronous entry point. In an application program, the address to which control is transferred when data is pending for an idle work station. See also *host entry point, station entry point, and terminal/device entry point.*

block. (1) The smallest complete unit of data that can be transmitted between units in a communication network. The maximum size of a block depends on the characteristics of the sending or receiving device. (2) a group of contiguous characters recorded as a unit. (3) On a controller diskette, the subdivision of a track. Depending on the diskette type, each track contains a fixed number of blocks. One record can occupy one or more blocks, or smaller records can be packed, or "blocked", to fit in diskette blocks.

buffer. In the 4700 system, a storage area that is reserved for use by data transmission operations.

communication link. (1) In general, the physical means of connecting one location to another for the purpose of transmitting and receiving information. (2) In the 4700 system, the communication link consists of an external at each location and a telephone line that connects the locations.

configuration. In the 4700 system, the group of terminals and controller storage areas and application programs that constitute a subsystem associated with a controller.

configuration image. A combination of formatted configuration data with selected modules of controller data that, when loaded into the controller storage, determines the operations of the controller.

controller log. In a controller, a temporary file on the diskette in which controller log messages are recorded and in which user data can also be recorded.

control operator. In a communication system, the person who performs special administrative, control, and testing functions.

cursor. (1) (SC1) In computer graphics, movable, visible mark used to indicate a position on a display space. (2) A movable spot of light on the screen of a display device, usually indicating where the next character will will be entered.

CPU. An abbreviation of "central processor unit", a deprecated term, that appears in macro parameters to refer to the host system.

debug. (ISO) To detect to trace, and to eliminate mistakes in computer programs or in other software. Synonymous with checkout.

digit. One of the numeric characters 0 through 9.

diagnostic diskette. A diskette containing system functions for detecting and defining system problems.

diskette. A thin, flexible magnetic disk and a semi-rigid protective jacket, in which the disk is permanently enclosed. See diagnostic diskette, installation diskette, operational diskette. Synonymous with flexible disk.

display. (1) a component that provides visual communication between the user and the controller. (2) a visual presentation of data.

dump. With reference to the controller, to copy a part of storage onto a diskette.

function key. (1) (SC1) In computer graphics, a button or switch that may be operated to send a signal to the computer program controlling the display. (2) A key on a terminal, such as the attention key, that causes the transmission of a signal not associated with a printable character. Detection of the signal usually causes the system to perform some predefined function for the user.

global storage. In a controller, programmable storage that is available to all work stations. Contrast with *private storage* and *shared storage*.

hexadecimal. A number system with a base of 16.

host or host system. (1) The primary or controlling computer in a multiple computer operation. (2) a computer used to prepare programs for use on another computer or on another data processing system; for example, a computer used to compile, link edit, or test programs to be used on another system. (3) The primary or controlling computer in a data communication system.

ID. Identification.

ID card. A card, similar in size to a credit card, that contains the users identification written on a magnetic stripe.

ID keys. Specially designed keys on shared terminals that identify the user to the controller.

inquiry. A request for information from storage.

installation diskette. A diskette used in a controller mainly to initiate communication with the host computer and to prepare the controller for reception and recording of the configuration image. Contrast with *operating diskette.*

institution. Any financial establishment, such as a commercial bank, mutual savings bank, savings and loan association, credit unit, and finance company.

LCF. Local Configuration Facility.

local control operator. A control operator at a local location. Contrast with *remote control operator.*

local location. A location that has a controller. Contrast with *remote location.*

local loop. A channel connecting the subscribers equipment to the line–terminating equipment in the central office exchange.

log. See *controller log.*

logical device address (LDA). A number used to designate a particular terminal component within a work station. Configuration data in the controller correlates the logical device address with the actual physical address. See *physical address.*

logoff. The steps by which a user signs off from the system.

logon. The steps by which a user signs on to the system.

loop. See *local loop* or *remote loop.*

loop number. In the 4700 system, a number that identifies a particular loop in a controller. See *physical address.*

magnetic stripe. The stripe on certain credit cards, ID cards, and passbooks on which data can be recorded magnetically.

magnetic stripe encoder/reader. A component available for the 4704/3604 that reads precoded information from, and writes coded information on, a magnetic stripe on a passbook, credit card, or ID card.

magnetic stripe reader. A component available for the 4704/3604 that reads precoded information from a magnetic stripe on a passbook, credit card, or ID card.

modem. (1) (SC1) A functional unit that modulates and demodulates signals. One of the functions of a modem is to enable digital data to be transmitted over analog transmissions facilities. (2)* (modulator-demodulator) A device that modulates and demodulates signals transmitted over data communication facilities. (3) See also data set (2), line adapter, modulation.

modem unit. A terminal (3604–2, –3, or –4 and any 3614 model) that has a modem or can be attached to an external modem.

numeric character. Same as digit.

operational diskette. A diskette used in a controller that contains the configuration image, and other data, relating to the operation of the controller. The operating diskette must be in the controller during its operations. A second diskette containing the same configuration image and data is referred to as a *backup operating diskette.* Contrast with *installation diskette.*

parameter. A variable that is given a fixed value for a specific application. See *parameter data byte* and *parameter flag byte.*

permanent file. In the 4700 system, a file on a diskette that can be used to store data to be retained from one controller startup to another. The permanent data might include such things as day–to–day totals or checkpoint/restart data. Contrast with *temporary file.*

physical address. In the 4700 system, an address that is used to reach a particular terminal or component. A physical address consists of a loop number, a terminal address, and a component address. In the configuration image in a controller, each physical address is correlated with a number (called a logical device address) that is used to identify a component in a work station. See *logical device address.*

private storage. In the controller, programmable storage that is associated with only one work station. Contrast with *global storage* and *shared storage.*

programmable storage. The portion of internal storage in the controller in which user–written programs are executed.

prompt. To help a terminal user by displaying messages that request information necessary to continue an operation.

record. Pertains to the classification of data stored on a diskette.

remote control operator. A control operator at a remote branch. Contrast with *local control operator.*

remote location. A location that is connected to the controller by a communication link. Contrast with *local location.*

remote loop. In the 4700 system, a closed circuit of telephones lines (not local cables) that starts at a controller and attaches remote locations one to another and back to the controller. Messages from the controller travel around the loop in one direction. Contrast with *local loop.*

remote subloop. In a remote loop, the closed circuit of cables that attach the remote terminals to each other at a remote location. See *remote loop.*

segment. In a controller, one of 16 portions into which the programmable storage related to a controller application program can be divided. The length of each segment is specified by the user.

shared storage. In the controller, programmable storage that is reserved for the application program and which may be shared between work stations. Contrast with *global storage* and *private storage.*

special character. Refers to any character other than the 26 letters and the 10 digits; for example, punctuation marks.

step. A fractional part of a print line on a passbook. There are 12 steps to a line.

storage. A part of the controller or host computer in which the program or data is kept.

subloop. See *remote loop.*

system monitor. The facility in a controller that handles communications with the control operator.

temporary file. In the 4700 system, a file on a diskette that can be used to store data tnat is not to be retained from one controller startup to another. This temporary data might include such things as a daily audit trail or a tellers cash position. Contract with *permanent file*.

transaction. (1) In the 4700 system, generally, an exchange between a terminal and another unit to effect a particular action or result. (2) More specifically, a single communication action involving an inquiry from a terminal that produces a response containing desired information (such as a request from a terminal for a customers account balance) or a more complex action in which data records must be changed (such as a request to update a customers balance with a new deposit).

work station. In the 4700 system, a collection of terminals and storage that is used by an application program executed by a controller to process transactions.

wrap test. At a local location, a test performed by the controller that checks the controller and its modem that connects to the remote loop. At a 1200–bps remote location or a location with an external modem, a test that checks the remote subloop and its modem. (**Note:** *Some external modem models cannot be wrap–tested; the wrap test is then valid for only the controller or terminal.*)

Index

D

E

F

I

P

R

S

T

U

V

4700 Finance Communication System
Controller Programming Library
Volume 4
Loop and DCA Device Programming
Order No. GC31-2069-1

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*
Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

____

____

____

____

What is your occupation?______________________________

Number of latest Newsletter associated with this publication:____________________

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 or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

GC31-2069-1

Reader's Comment Form

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 78C
1001 W.T. Harris Boulevard
Charlotte, NC, USA 28257

Fold and Tape | Please Do Not Staple | Fold and Tape

IBM®

4700 Finance Communication System Controller Programming Library Volume 4
Loop and DCA Device Programming (S370/4300/8100/S34-30) Printed in U.S.A. GC31-2069-1

4700 Finance Communication System
Controller Programming Library
Volume 4
Loop and DCA Device Programming

Order No. GC31-2069-1

READER'S COMMENT FORM

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*
Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

____

____

____

____

What is your occupation? ______________________________

Number of latest Newsletter associated with this publication: ____________________

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 or you may mail directly to the address in the Edition Notice on the back of the title page.)

GC31-2069-1

Reader's Comment Form

Fold and Tape — **Please Do Not Staple** — Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 78C
1001 W.T. Harris Boulevard
Charlotte, NC, USA 28257

Fold and Tape — **Please Do Not Staple** — Fold and Tape

IBM®

4700 Finance Communication System Controller Programming Library Volume 4
Loop and DCA Device Programming (S370/4300/8100/S34-30) Printed in U.S.A. GC31-2069-1

Publication Number
GC31-2069-1

File Number
S370/4300/8100/S34-30

Printed in
USA

IBM