# United States Patent [19]

## Bryan et al.

#### [54] PERSONAL COMPUTER ATTACHMENT FOR HOST SYSTEM DISPLAY STATION

- [75] Inventors: Barry L. Bryan, Woodstock; Martin Druckerman, Poughkeepsie; Allen W. McDowell, Lake Katrine; Ira H. Schneider, Kingston; Gary L. Newkirk, New Paltz, all of N.Y.
- [73] Assignee: International Business Machines Corporation, Armonk, N.Y.
- [21] Appl. No.: 585,813
- [22] Filed: Mar. 2, 1984

#### **Related U.S. Application Data**

- [63] Continuation-in-part of Ser. No. 473,058, Mar. 7, 1983, abandoned.
- [51] Int. Cl.<sup>4</sup> ..... G06Z 15/00; G06Z 3/00
- [58] Field of Search ... 364/200 MS File, 900 MS File,
- 364/521; 340/721, 745; 358/181

### [56] References Cited

### **U.S. PATENT DOCUMENTS**

4,115,868	9/1978	Suzuki et al 364/900
4,271,479	6/1981	Cheselka et al 364/900
4,313,176	1/1982	Cecil
4,324,954	4/1982	Taylor 179/90 B
4,433,392	2/1984	Beaven
4,445,115	4/1984	Rudgard 340/745
4,458,311	7/1984	Clements et al

#### OTHER PUBLICATIONS

"Intelligent Processor Feature Attachment to a Fixed Function Display Terminal" by Pellacto et al., IBM Technical Disclosure Bulletin, vol. 24, No. 7A, Dec., 1981, pp. 3533–3534.

Primary Examiner-James D. Thomas

Assistant Examiner-Thomas C. Lee

[57]

Attorney, Agent, or Firm-Frederick D. Poag; C.

# Lamont Whitham

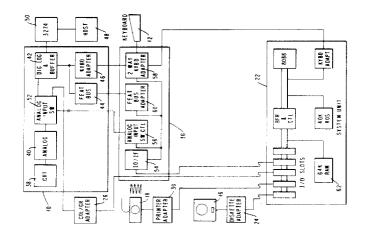
#### ABSTRACT

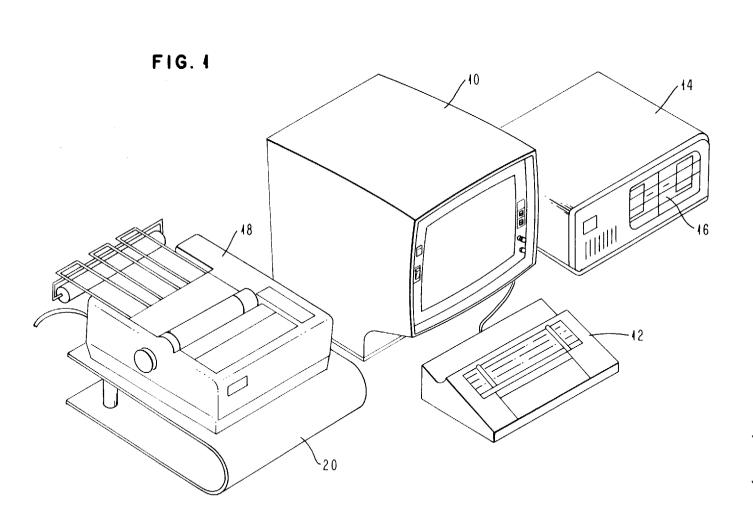
A personal computer attachment is provided for a display station of the type that communicates with a host computer (48). The display station has a display unit (10) and a keyboard (12) to which a personal computer system unit (14) is attached. The personal computer [11] Patent Number: 4,641,262

## [45] Date of Patent: Feb. 3, 1987

system unit typically supports floppy diskette drives (16) and a printer (18). The display unit includes a CRT (38), a regeneration buffer (42), a keyboard adapter (46) and a feature bus (44). The personal computer system unit includes a system bus, a microprocessor, memory, a keyboard adapter and I/O interface connected to the system bus. A display adapter (26 or 28) is connected to the I/O interface. An analog input switch (52) is disposed between the buffer (42) and the analog circuits driving the CRT (38). This switch has a second input from the display adapter (26 or 28) and is controlled from inputs from the keyboard (12) to selectively supply image data from the buffer (42) or the display adapter (26 or 28). An attachment adapter (36) mates with the I/O interface of the personal computer system unit. The adapter (36) includes an input/output interface (54), a switch control (56), a two-way keyboard adapter (58) and a feature bus adapter (60). The keyboard (12) is connected to the two-way keyboard adapter (58). Each of the switch control (56), the twoway keyboard adapter (58) and the feature bus adapter (60) communicate with the system bus of the personal computer system unit via the input/output interface. The switch control (56) is also connected to the control input of the analog input switch (52). The two-way keyboard adapter (58) is also connected to the keyboard adapter in the display station and the keyboard adapter in the personal computer system unit. The feature bus adapter (60) is also connected to the feature bus (44) of the display station. This arrangement allows keystroke signals from the keyboard (12) to be transmitted by the two-way keyboard adapter (58) via the input/output interface (54) and system bus to the memory in the personal computer system unit for interpretation by the microprocessor and then retransmitted back to the twoway keyboard adapter and then to a designated one of the keyboard adapters in either the display station or the personal computer system unit. The switch control is responsive to a unique keystroke signal to control the analog input switch (52). In this way, the operator can control the mode of operation between either a host mode or a personal computer mode. Further, data transfer can be made between the host computer and the personal computer.

#### 16 Claims, 22 Drawing Figures

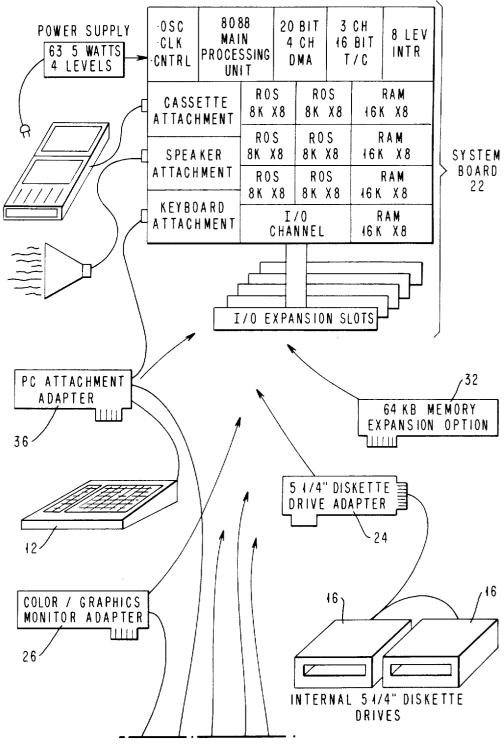


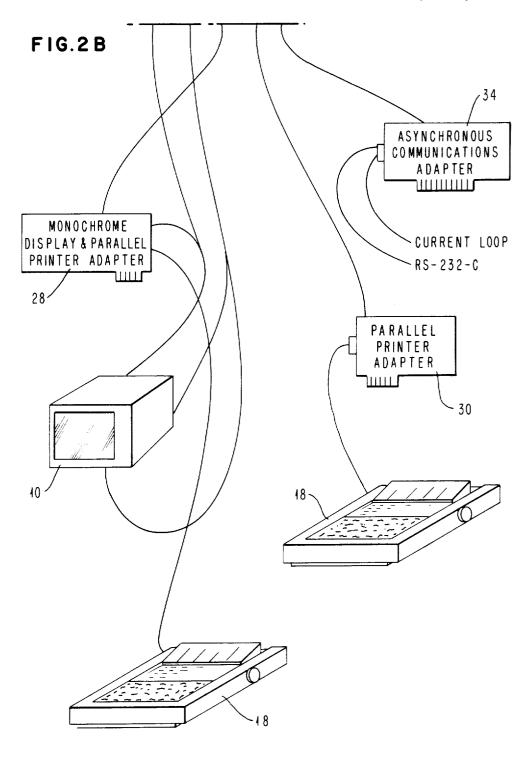


U.S. Patent Feb. 3, 1987

Sheet 1 of 16 4,641,262

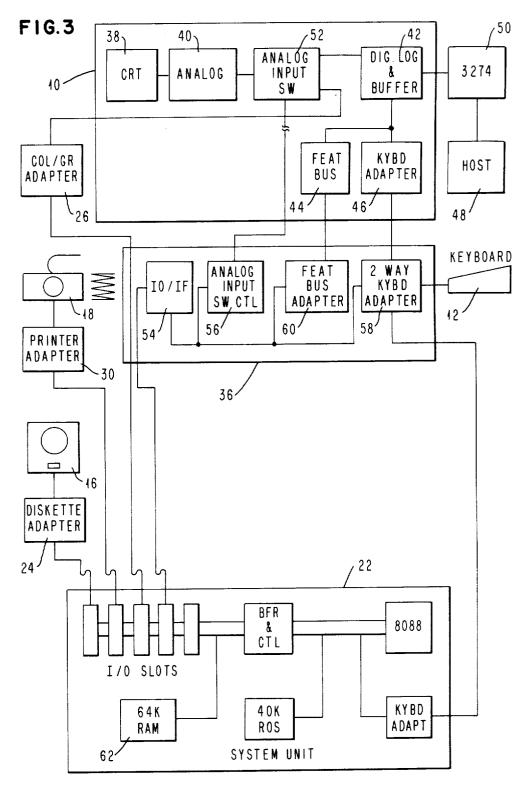
FIG.2A

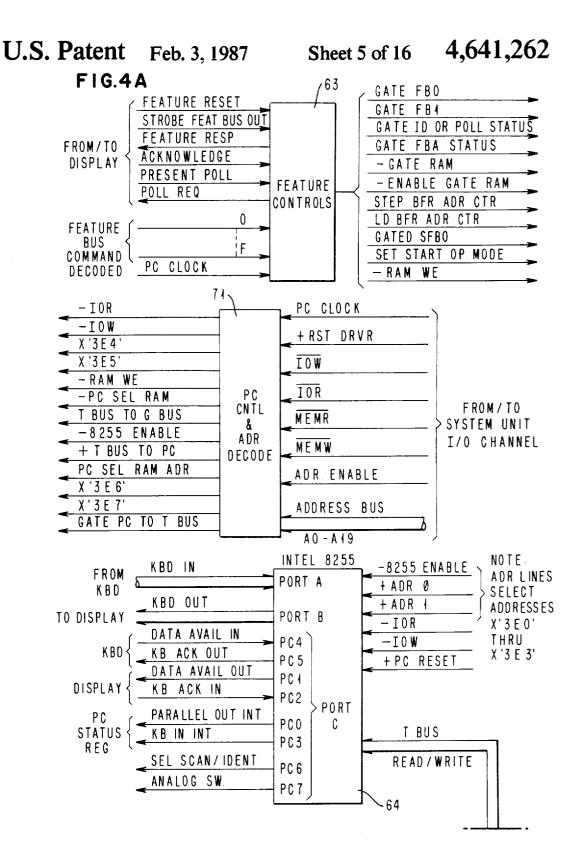


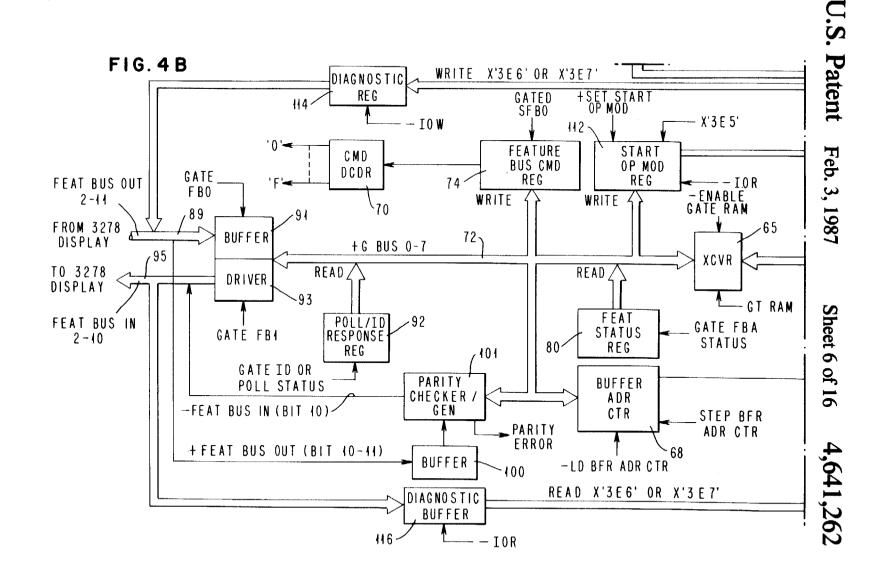


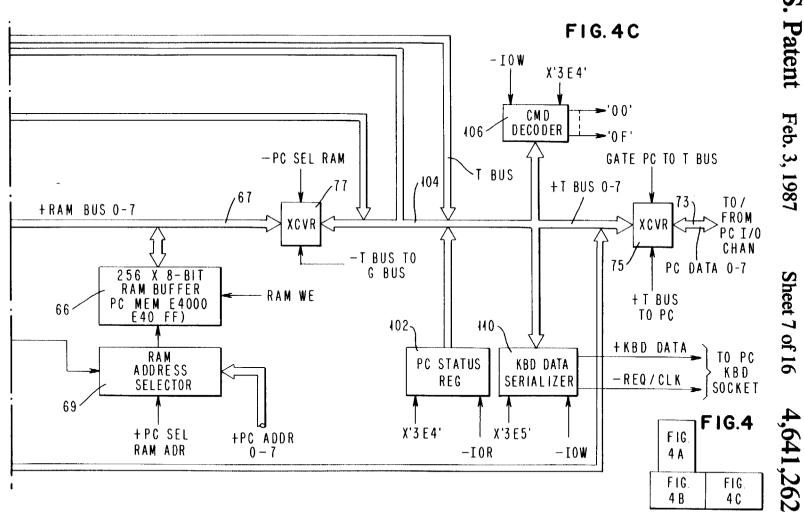
.

.









U.S. Patent Feb. 3, 1987

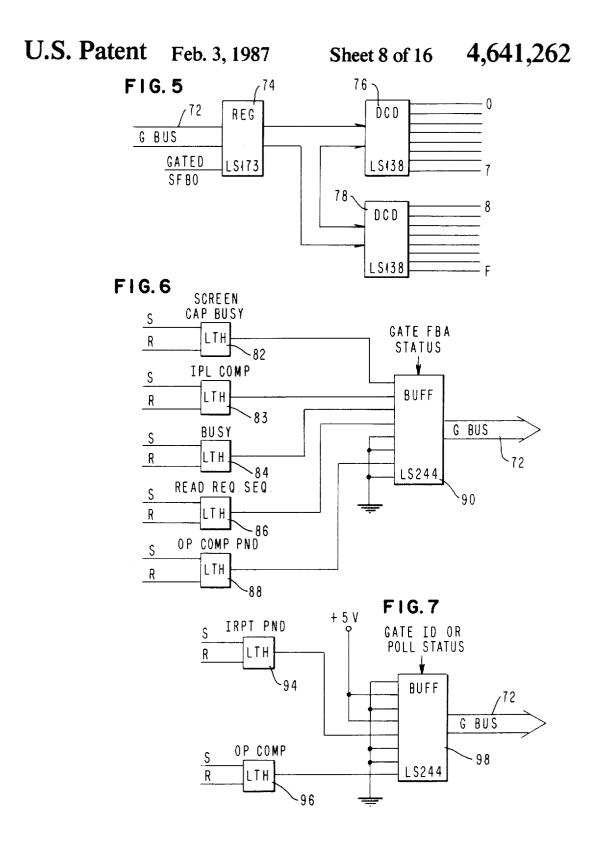
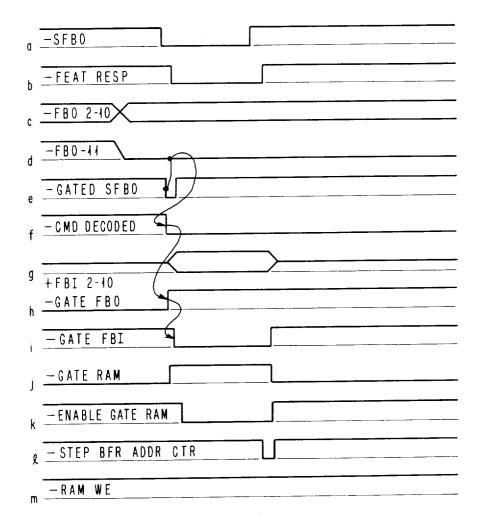


FIG. 8



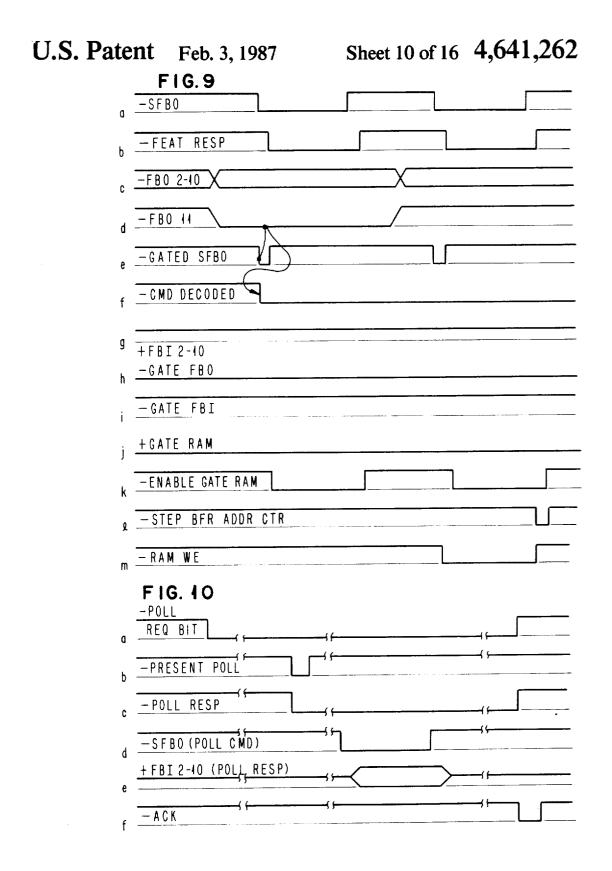


FIG. 44

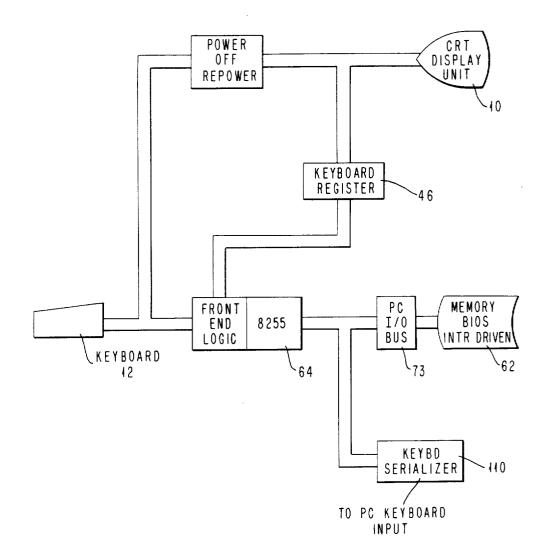
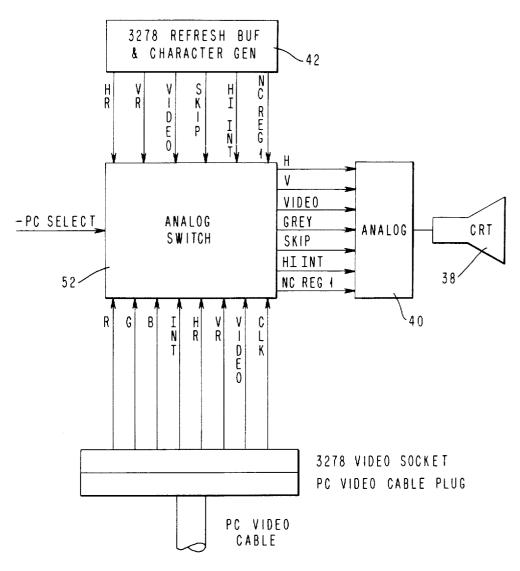
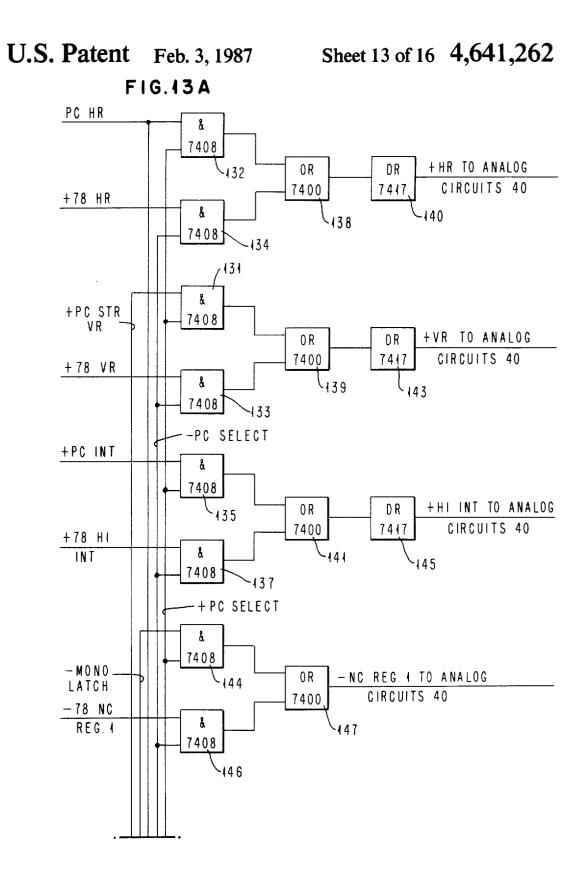
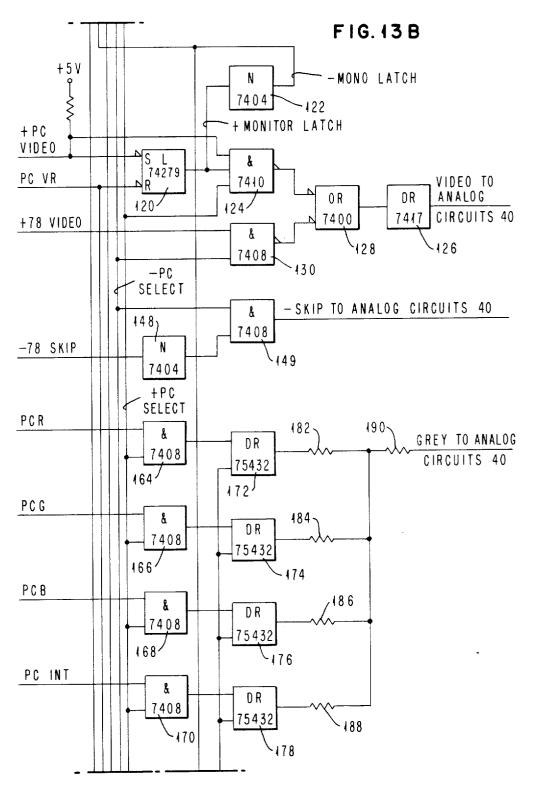
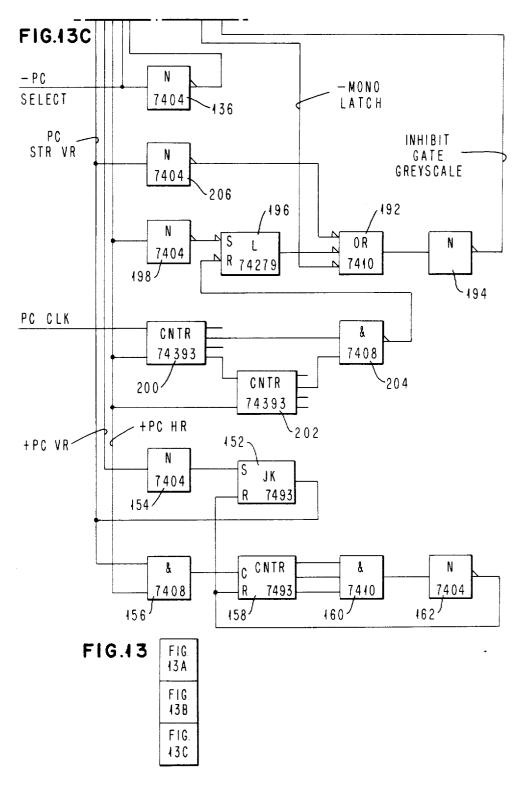


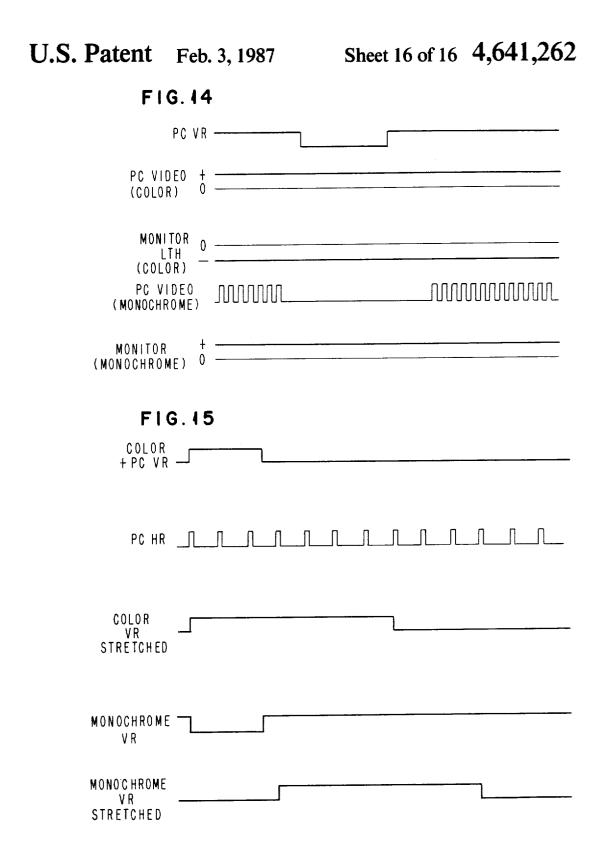
FIG. 12











### PERSONAL COMPUTER ATTACHMENT FOR HOST SYSTEM DISPLAY STATION

#### CROSS REFERENCE TO RELATED APPLICATION

This application is a continuation-in-part of application Ser. No. 06/473,058, filed Mar. 7, 1983, now abandoned.

#### FIELD OF THE INVENTION

The present invention generally relates to data processing systems of the type having a central host computer and a plurality of display stations each typically having a keyboard and a display such as a CRT, and <sup>15</sup> more particularly to a personal computer attachment for the display stations to allow the display stations to operate with the host or with the personal computer and to allow the transfer of files between the host and the personal computer and the printing of the data dis-<sup>20</sup> played on the screen of the CRT.

#### **BACKGROUND OF THE INVENTION**

Data processing systems are known where a plurality of display systems communicate with a central host <sup>25</sup> computer. These display systems typically include a base unit which acts as a display and supports a keyboard for operation as a keyboard/display terminal that communicates with the host via a controller. The terminal may interface with a plurality of features connected 30 thereto such as a card reader or a light pen. The base unit and the features receive transmission from the controller via a feature bus in the base unit and also transmit to the controller via this feature bus. The base unit also provides priority control in permitting features to re- 35 spond to a controller poll. Commands are also directly addressable to the base unit and features by the controller. Typical of such systems are the IBM 3270 information display system, the IBM 3274 control unit and the IBM 3278 display stations.

In such systems, the display stations are used to access data and run programs on the host. Storage of data or printing a hard copy typically are accomplished using bulk storage media or printers associated with the host computer. These may not always be readily acces- 45 sible or available to a display station operator. Moreover, some data may be sensitive or confidential, and the operator may desire to exercise direct control over the bulk storage media on which the data is stored or the hard copy produced by the printer. Current display 50 major sub-features. The first is the modifications to the stations do not provide this access or flexibility. Recently, small desk top or so-called personal computers have been introduced on the market. These typically include a system unit containing a microprocessor, read and various adapters connected to the system unit bus to allow connection of a keyboard, a CRT display, one or more floppy diskette drives, possibly one or more hard disk drives, and a printer. A personal computer bility to both bulk storage media, in this case a floppy diskette, and a printer, but to simply provide the display station operator with a personal computer would be a duplication of equipment and no ability to use host files on the personal computer and vice versa.

It is therefore an object of this invention to provide a personal computer attachment for a display station that will allow the display station operator to operate the display station in either a host mode or a personal computer mode without duplication of equipment.

It is another object of the invention to provide a personal computer attachment for the display station 5 that will allow transfer of files between the host and the personal computer and vice versa.

#### SUMMARY OF THE INVENTION

The objects of the invention are accomplished by 10 providing an attachment between the feature bus of the base unit of the display station and the system unit of the personal computer. Both the keyboard and the display can be used in either a host mode or a personal computer mode in response to commands entered on the keyboard on the display station operator. For purposes of providing a specific example of the invention, the following description is directed to the attachment of the the IBM Personal Computer (PC) to the IBM 3278 display station. It will be understood by those skilled in the art, however, that the invention may be practiced with other so-called personal computers or sub-units thereof and other display units manufactured by others than the International Business Machines Corporation.

The 3278 personal computer attachment provides a method for combining the IBM 3278 display station with the IBM PC. It uses the 3278 display and keyboard and does not require the display or keyboard of the personal computer. The personal computer attachment allows existing host programs to be run unmodified, and also allows most Disk Operating System (DOS) based PC programs to run unmodified. These two operations are referred to as "host mode" and "PC mode". They can run concurrently with no interference between the programs. An additional option is offered to provide data transfer between personal computer and host programs.

The personal computer attachment use the existing personal computer system unit. There are no hardware or software modifications required. The attachment does, however, add a new adapter card and some new software (called an interrupt handler) to use the 3278 display and keyboard. No host hardware or software modifications are necessary to operate the personal computer attachment. The attachment is designed to maintain the integrity of the host session in the event of a disruption in the personal computer operation (caused by program, hardware or power failure).

The personal computer attachment consists of three 3278 display to accept the output of one of two PC video adapters; either the monochrome adapter or the color graphics adapter.

The second sub-feature consists of a new adapter for only storage (ROS), random access memory (RAM), 55 the PC (called the 3278 adapter) and a group of interconnecting cables. This sub-feature also includes 51 inch diskettes. On those diskettes is an interrupt handler program, some sample application programs for the host computer and the PC to support file transfer and would provide the display station operator with accessi- 60 screen capture, and installation and diagnostic programs.

The third sub-feature is a modification to the 3274 microcode which will support data transfer (both file transfer and screen capture).

The personal computer system unit comprises a keyboard adapter, an Intel 8088 microprocessor with 40K of ROS in which is stored character generation data and the Basic Input/Output System (BIOS), 64K to 256K of

65

RAM and a group of five or eight Input/Output (I/O) slots which can receive diskette drive adapters, printer adapters, display adapters, additional RAM and so forth. The I/O slots are connected in parallel to the system unit bus.

The 3278 has some digital logic and a regeneration buffer. The digital logic includes a keyboard adapter and the feature bus. These are used as part of the personal computer attachment. The digital logic attaches to a 3274 control unit via an interface commonly re- 10 ferred to as the "DCA" for device control adapter. The output of the digital logic normally goes through some analog circuits to generate an image on the face of the CRT display. An additional circuit function (called the analog input switch) is introduced between the digital 15 logic and the analog logic. This allows the analog logic to be driven from either the normal digital logic of the 3278 or from the output of a video adapter plugged into the personal computer system unit. The state of this switch is controlled by an external source, which in the 20 specific example being described, is the 3278 adapter plugged into one of the I/O slots in the personal computer system unit.

The 3278 adapter consists of four major sub-functions. The first of these is the PC I/O interface function 25 which allows the 3278 adapter to meet the interface of the I/O slots of the personal computer.

The second sub-function of the 3278 adapter is the analog input switch control. These circuits (which are controlled by the PC program) will cause the analog 30 formed. Using the upload function, the operator ininput switch in the 3278 to switch states. Thus, a PC program has the capability to determine what is displayed on the face of the CRT; either the 3278 host image or the personal computer display adapter image. Because the analog switch is introduced in the position 35 just described, the 3278 regeneration buffer and its supporting logic operate just as they did before installation of the personal computer attachment. The same is true for the regeneration buffer and its supporting logic in the display adapter of the personal computer. This al- 40 lows the host program and the personal computer program to continue to generate images and send them to their respective regeneration buffers, thus allowing concurrent operation of both the host program and the 45 personal computer program.

The third sub-function in the 3278 adapter is the twoway keyboard adapter. It operates in conjunction with the PC program. An unmodified 3278 keyboard is plugged into the two-way keyboard adapter using the existing 3278 keyboard cable and connector. The two- 50 way keyboard adapter accepts keystroke scan codes from the 3278 keyboard and feeds them to the personal computer system unit RAM. The PC program is expected to interrogate these scan codes in RAM, alter them (if necessary) using a table look-up, send them 55 back to the two-way keyboard adapter and direct whether the adapter output is sent to the existing keyboard adapters in the 3278 or PC. The program that supports this function is called the interrupt handler. The interrupt handler also monitors the 3278 keyboard 60 operation for a unique key sequence which is the way the operator requests the interrupt handler to switch modes. When in the host mode, the CRT image and keystrokes are associated with the host program; when in the PC mode, the CRT image and keystrokes are 65 associated with the PC program. This is a toggle operation; that is, when a key is hit once, it switches from one mode to the other, and when hit again, it switches back.

An inherent capability of this design is that a PC program can generate keystrokes to the 3278 or personal computer system unit which are identical to those a human operator could have generated.

The fourth sub-function is the feature bus adapter which provides circuits to connect the 3278 adapter to the feature bus of the 3278 display. The feature bus is normally an internal bus of the 3278 display and exists to support such features as light pen, magnetic stripe card reader, etc. This feature bus is extended outside the 3278 base unit to the mating portion of the PC adapter to provide a relatively high speed, two-way path for sending data from the host to the PC and vice versa.

The function referred to as "Data Transfer" is divided into two sub-functions; file transfer and screen capture. In order to make use of this feature, a modification in the microcode of the 3274 controller is provided. This modification in the microcode will recognize additional 3270 structured fields and direct the data sent from the host to the feature bus rather than to the 3278 regeneration buffer. A program in the personal computer can accept that data and act on it immediately or store it on diskette for later use with other PC programs. In addition, that program will take data from the diskette and send it via the feature bus and the 3274 controller to the host.

The ability for a PC program to generate 3278 keystrokes enables the function called "upload" to be pervokes a keystroke generation program in the PC. This program produces the keystrokes which are sent to the host and appear to be macro-language instructions. The instructions are a bootstrap program which bring in the rest of the file transfer program. This file transfer program is a private exec which is permanently stored and available for use anytime the operator invokes it. It only requires this upload once.

The other sub-function is screen capture. The only place within the display sub-system that the screen image resides is in the regeneration buffer of the 3278 display. Additional microcode is provided in the 3274 controller to unload that screen image from the regeneration buffer of the 3278 display into the 3274 controller and redirect it out through the feature bus to the PC system unit. A program in the PC system unit will write this image to diskette or to the printer adapter giving the capability for local copy.

An important integrity aspect of the design is that if the personal computer application fails, then the operator can, using the keystroke sequence, switch to host mode and continue the host operation. Additionally, if power is removed from the personal computer, then keystroke data is redirected to the display station rather than the personal computer without invoking the switchover key sequencing. This is made possible because the front end of the two-way keyboard adapter is powered directly from the 3278 display.

## BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages of the invention will be better understood from the following detailed description with reference to the accompanying drawings, in which:

FIG. 1 is a perspective view of a 3278 display station flanked by a personal computer system unit and a printer;

FIGS. 2A and 2B together from a pictorial representation of the personal computer and the various adapter options available including the 3278 adapter;

FIG. 3 is a block diagram of the 3278 display and the personal computer showing how they are interfaced 5 with the 3278 adapter;

FIG. 4 illustrates the relationship of FIGS. 4A, 4B and 4C which, taken together, are in turn a detailed block diagram of the 3278 attachment;

FIG. 5 is a block diagram of the feature command 10 decoder of the 3278 adapter;

FIG. 6 is a block diagram of the feature status register of the 3278 adapter;

FIG. 7 is a block diagram of the feature poll register of the 3278 adapter;

FIG. 8 is a timing diagram illustrating the operation of reading from the RAM buffer on the 3278 adapter by the 3278 display;

FIG. 9 is a timing diagram illustrating the operation of writing into the RAM buffer on the 3278 adapter by 20 adapter 36 is provided to be plugged into one of the I/O the 3278 display;

FIG. 10 is a timing diagram illustrating the operation of the poll request and response sequence between the 3278 display and the personal computer attachment feature:

FIG. 11 is a block diagram of the keyboard data paths;

FIG. 12 is a block diagram showing the cable connections between the analog switch, the analog display circuits, the digital logic and refresh buffer, and the 30 video cable input from the personal computer system unit:

FIG. 13 illustrates the relationship of FIGS. 13A, 13B and 13C which, taken together, are in turn logic diagrams of the monitoring logic and video switch, the 35 driven by analog circuits 40 in a conventional manner. horizontal sync switch, the vertical sync switch and related switching logic, the vertical retrace signal stretcher, the color to grey scale translator, and the grey scale gate signal generator which comprise the analog switch in the preferred embodiment; 40

FIG. 14 is a timing diagram illustrating the operation of the monitoring logic; and

FIG. 15 is a timing diagram illustrating the operation of the vertical retrace signal stretcher.

#### DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to the drawings, and more particularly to FIG. 1, a 3278 display station is shown as comprising a CRT display 10 and a keyboard 12. On the 50 right of the CRT display 10 is an IBM Personal Computer system unit 14 which may have one or two floppy diskette drives 16 installed in the system unit cabinet. On the left of the CRT display 10 is a printer 18, here shown on an optional stand 20. The printer 18 is con- 55 nected to and controlled by the personal computer system unit 14.

FIGS. 2A and 2B show the IBM Personal Computer system unit mother board 22 mapped to generally show the location of the Intel 8088 microprocessor, the read 60 only storage (ROS), the random access memory (RAM), the keyboard attachment, and the I/O expansion slots as well as other parts of the system unit. A number of adapters in the form of printed circuit boards are available which are designed to be plugged into one 65 of the I/O slots. For example, a diskette drive adapter 24 capable of supporting up to four diskette drives would be plugged into one of the I/O slots. This in turn

would be connected to the two internal diskette drives 16. Two monitor adapters are available for the IBM Personal Computer; a color graphics monitor adapter 26 and a monochrome display and parallel printer adapter 28. Typically, one or the other of the adapters 26 or 28 would be plugged into one of the I/O slots and a connection made from the adapter to the base unit of the 3278 display. If the monochrome display and parallel printer adapter 28 is used, then a cable connection is also made from the adapter to the printer 18. On the other hand, if the color graphics monitor adapter 26 is used, it is necessary to plug the parallel printer adapter 30 in to one of the I/O slots and make the cable connection from this adapter to the printer 18. Another I/O 15 slot might have plugged into it a RAM expansion card 32 and/or an asynchronous communications adapter 34, the latter having a cable connection to a modem (not shown) for connection to a telephone line, for example.

Accordingly to the invention, a 3278 attachment slots of the IBM Personal Computer system unit. This adapter 36 has cable connections to the base unit of the 3278 display station and cable connections to both the 3278 keyboard and the keyboard attachment on the 25 mother board of the system unit. The relationship of the 3278 attachment adapter to the 3278 display station and the personal computer system unit is best shown in FIG. 3 to which reference is now made.

In FIG. 3, the same reference numerals as used in the preceding figures refers to the same components or circuits. Therefore, it will be understood that the several adapters shown in block diagram form are plugged into the I/O slots in the sytem unit. The 3278 display unit 10 is represented as having a CRT 38 which is The analog circuits 40 are controlled by digital logic and a regeneration buffer 42 which is also connected to the feature bus 44 and the keyboard adapter 46. Normally, the 3278 keyboard 12 would be connected to the keyboard adapter 46. In addition, the 3278 display station communicates with a host computer 48 via a 3274 controller 50.

The circuitry described thus far for the 3278 display station 10 is conventional. For purposes of the subject 45 invention, this circuitry is modified by the addition of an analog input switch 52. Normally, the output of the digital logic and regeneration buffer 42 goes through the analog circuits 40 to generate an image on the face of the CRT 38. However, the analog input switch 52 is introduced between the digital logic and regeneration buffer 42 and the analog circuits 40. The analog input switch has a second input which is connected to the color graphics adapter 26 or optionally the monochrome monitor and parallel printer adapter (not shown in FIG. 3). The analog input switch 52 thus allows the analog circuits 40 to be driven from the normal digital logic and regeneration buffer 42 of the 3278 display unit or from the output of a video adapter plugged into the personal computer system unit. The state of the analog input switch is controlled by the 3278 adapter 36.

The 3278 adapter 36 has four major sub-functions; the I/O interface function 54, the analog input switch control 56, the two-way keyboard adapter 58, and the feature bus adapter 60. The I/O interface function 54 allows the 3278 adapter 36 to meet the interface of the I/O slots of the personal computer system unit. The analog input switch control 56 is controlled by a personal computer program to cause the analog input

switch 52 to switch states. Thus, the personal computer program has the capability to determine what is displayed on the face of the CRT 38; either the 3278 host image or the personal computer display adapter image. Because of the position of the analog input switch 52, 5 the regeneration buffer 42 and its supporting logic operate as they did before installation of the personal computer attachment. The same is true for the regeneration buffer and its supporting logic in either the color graphics adapter 26 or the monochrome monitor and parallel 10 printer adapter 28. This allows the host program and the personal computer program to continue to generate images and send them to their respective regeneration buffers, thereby allowing concurrent operation of both the host program and the personal computer program. 15

The third sub-function in the 3278 adapter 36 is the two-way keyboard adapter 58 which also operates in conjunction with a personal computer program. In an unmodified 3278 display station, the keyboard 12 is plugged into the keyboard adapter 46; however, when 20 the personal computer attachment is installed, the keyboard 12 is plugged into the two-way keyboard adapter 58 using the existing 3278 keyboard cable and connector. The two-way keyboard adapter 58 accepts keystroke scan codes from the keyboard 12 and feeds them 25 to the personal computer memory 62 via the I/O interface 54. The personal computer program is expected to interrogate these scan codes in memory 62, alter them (if necessary) using table lookup in memory, send them back to the two-way keyboard adapter and direct 30 whether the adapter output is sent to the existing keyboard adapters in the 3278 display or the personal computer. The program that performs this function is called the interrupt handler which, in addition, monitors the keyboard 12 operation for a unique key sequence which 35 is the way the operator requests the interrupt handler to switch modes. When in the host mode, the CRT image and keystrokes are associated with the host program; when in the personal computer mode, the CRT image and keystrokes are associated with the personal com- 40 puter program.

The feature bus adapter 60 provides circuits to connect the 3278 adapter 36 to the feature bus 44. The feature bus 44 normally supports such features as light pen, magnetic strip readers, programmable symbol sets, 45 and the like. This feature bus mates with the feature bus adaper 60 of the 3278 adapter 36 to provide a relatively high speed two-way path for sending data from the host to the personal computer and vice versa.

The 3278 attachment adapter 36 is shown in greater 50 detail in FIG. 4 and includes a 256×8-bit RAM buffer 66 that can be written to or read from by either the personal computer or by a feature command, but not simultaneously. The buffer 66 resides in the personal computer memory address space starting at base ad- 55 dress X'C4000'. The only time the personal computer is allowed access to the buffer 66 is the time when the feature has received a Start Op command until the personal computer responds with an Op Complete Pending. The only other time that the personal computer 60 may have access to the buffer 66 is if the Initial Program Load (IPL) latch has been reset. At all other times the buffer 66 is dedicated to the feature bus adapter. The feature bus reads the buffer 66 at the address specified by the buffer address counter 68 with either a Read 65 Data or Read Multiple command. Data is written to the buffer 66 at the address specified by the RAM address selector 69 with a Write command from the feature bus.

The buffer address counter **68** is one byte wide. It is loaded with the Set Address Counter Low command from the 3274 controller. It can also be reset to zero with the Reset command from the 3274 controller. The buffer address counter **68** contains the address which is to be read from or written to by the 3274 controller.

The buffer 66 is connected to a RAM bus 67 which is isolated by two tri-state tranceivers 65 and 77. When provided with an enable signal and an appropriate direction control signal, the transceiver 65 allows data to flow from the G bus 72 to the RAM bus 67 or vice versa. Similarly, the transceiver 77 will allow data to flow from the T bus 104 to the RAM bus 67 or vice versa. An interface to the PC data bus 73 is provided by yet another tri-state tranceiver 75 which, when provided with an enable signal and an appropriate direction control signal, allows data to flow from the PC data bus 73 to the T bus 104 or vice versa. The interface to the 3278 display comprises two parts, a tri-state buffer 91 and a tri-state driver 93. These serve the same function as the transceivers 65, 75 and 77 but they are necessary to provide the connections to the feature bus out 89 and the feature bus in 95. The buffer 91 and the driver 93 allow data flow in only one direction so direction control signals are not required. When buffer 91 is provided with an enable signal, data on the feature bus out 89 is allowed to flow onto the G bus 72. When driver 93 is enabled, data on the G bus 72 is allowed to flow onto the feature bus in 95.

The feature command decoder 70 is connected to the G bus 72 by the feature command register 74. The decoder 70 is shown in more detail in FIG. 5 and comprises two decoder circuits 76 and 78 connected to command register 74. These decoder circuits provide feature command outputs which are listed below:

COMMAND	DESCRIPTION
X'1'	Nop
X'2'	Poll. Sends poll status to feature bus.
X'3'	Reset. Sets buffer address counter to all zeros.
X'4'	Read Data. Reads one byte of data.
	Increments buffer address counter.
X'5'	Write BAR High. Loads the buffer address counter high.
X*6*	Write BAR Low. Loads the buffer address counter low.
X'7'	Read ID. Feature ID and poll status are sent to the feature bus.
X*8*	Start Op. Loads Start Op modifier register.
X*9'	Nop
X'A'	Nop
Х'В'	Read Multiple. Reads up to four bytes of data from the buffer.
X'C'	Write Data. Stores up to 256 bytes of data in the buffer.
X'D'	Nop
X'E'	Nop
XF	Read Status. Sends contents of feature
	status register to feature bus.

Returning now to FIG. 4, the feature status register 80 is one byte wide and provides an output to the G bus 72. This register is read only by the 3274 controller and is read using a Read Status command. This Read Status command is decoded by the command decoder 70 which provides an output to the feature controls block 63 (FIG. 4A). In response to the decoded Read Status command, the feature controls block 63 generates a Gate FBA (Feature Bus Adapter) Status signal to the feature status register 80 causing it to ouput its content on the G bus 72. FIG. 6 shows the feature status register in more detail. It comprises five latches 82, 83, 84, 86, and 88 and a buffer 90. The following is a description of the bits in the feature status register beginning with the 5 most significant bit:

BIT	DESCRIPTION
7	Screen Capture Busy. Set by PC write command (bit $7=1$ ) when screen capture message is receive. Reset by a PC write command (bit $7=0$ ) or PC reset or Reset.
6	IPL Complete. Set by PC write command (bit 6=1). Reset by PC write command (bit 6=0) or PC reset.
5	Not Busy. Set by feature bus adapter when 'Op Complete Pending' is set in feature status register or PC reset. Reset when a Start Op Modifier has been received.
4	Read Request Sequence. Set by PC write command (bit $4=1$ ) when it is ready for the 3274 to read the buffer. Reset by PC write command (bit $4=0$ ) or PC reset or Reset.
3	Reserved.
2	Reserved.
1	Op Complete Pending. Set by PC write command (bit 1=1). Reset by Acknowledge to feature poll or PC reset or Reset.
0	Reserved.

Returning again to FIG. 4, the feature poll/ID response register 92 is connected to the G bus 72. This <sup>30</sup> register is also one byte wide and is read only by the 3274 controller. As shown in FIG. 7, register 92 comprises two latches 94 and 96 and a buffer 98. The bit descriptions are give below beginning with the most significant bit: <sup>35</sup>

BIT	DESCRIPTION	
7	0	
6	1	
5	$\emptyset$ Feature ID = X'5'	
4	1 /	
3	Feature Interrupt. Set if IPL complete has been activated or Read Request Sequence has been activated. Reset by Acknowledge to poll or PC reset or Reset	
2	Reserved.	
1	Reserved.	
0	Op Complete. Set by PC write command (bit $0$ =Reset by Acknowledge to poll or PC reset or Reset.	

Returning to FIG. 4, when a Read Feature ID command is decoded by the adapter, the poll/ID response register 92 is gated onto the feature bus in 95.

The control signals from the 3278 feature bus are logically combined with the outputs of the feature bus 55 command decoder 70 to generate gating and enable signals for reading from or writing to the RAM buffer 66 and various ones of the registers on the attachment adapter 38 by the 3278 display station. The logic that generates these gating and enable signals is generally 60 represented by block 63 which has as inputs the control signals Feature Reset, Strobe Feature Bus Out (FBO), Acknowledge, Present Poll, and the outputs 'Ø' to 'F from the command decoder 70. The control signals Feature Reset, Strobe Feature Bus Out, Acknowledge, 65 and Present Poll originate from the 3278 display and are communicated via the feature bus out 89. Also communicated via the feature bus in 95 to the 3278 display are

the Feature Response and the Poll Request signals. The feature read signals produced by the feature control block 63 include Gate ID or Poll Status, Gate FBA (feature bus adapter) Status, and Gate RAM in combination with the Gate FBI (feature bus in) signal. The feature write signals produced by the feature control block 63 include LD BFR ADR CTR and -RAM WE (write enable) in combination with the Gate FBO signal. The specific details of the logic circuitry in feature

<sup>10</sup> control block 63 are shown in Appendix B of the *IBM* 3270 Personal Computer Attachment User's Guide published in March 1983, and the procedure by which a 3278 display communicates with an attached feature is described in U.S. Pat. No. 4,271,479 to Cheselka et al.

<sup>15</sup> However, a few examples should illustrate the operations performed.

Referring now to FIGS. 4 and 8, consider first the operation of reading from the buffer 66 to the 3278 display via the feature bus in 95. Parallel command bits 20 on the feature bus out 89 are inputted to the tri-state buffer 91 as indicated in FIG. 8 as c and d. FIG. 8 at c indicates that bits 2 to 10 may be "0" or "1" depending on the code of the command, while FIG. 8 at d indicates that bit 11 is a "0" indicating that the data on the feature 25 bus out 89 is a command. Bits 10 and 11 are supplied to buffer 100 which supplies an input to the parity checker/generator 101. The parity checker/generator 101 checks the parity of the incoming command or other data and, if there is a parity error, provides an error output signal which is processed in a conventional manner. Since as shown in FIG. 8 at h the Gate FBO signal to the tri-state buffer 91 is low, bits 2 to 9 on the feature bus out 89 are transferred to the G bus 72. The 3278 35 display then causes the input control signal Strobe Feature Bus Out (SFBO) to go low as shown in FIG. 8 at a. A negative gated SFBO signal shown in FIG. 8 at e gates the command on the G bus 72 into the feature bus command register 74. This register 74 supplies the command to decoder 70 which provides a feature bus com-**4**0 mand signal on one of lines 'Ø' to 'F' to the feature controls block 63. At this point it has been logically determined that a read operation is to be performed from the buffer 66. The gate FBO signal now goes 45 positive as shown in FIG. 8 at h turning off the tri-state buffer 91 and blocking further input from the feature bus out 89 to the G bus 72. The gate FBI signal then goes negative as shown in FIG. 8 at i enabling the tristate driver 93 to allow data on G bus 72 to pass to the 50 feature bus in 95. Since there is assumed to be no parity error and the command is assumed to be valid in this example, the feature response signal is sent to the 3278 display as shown in FIG. 8 at b. At the same time, the gated SFBO signal goes positive isolating register 74 from the G bus 72. Two signals are now sent to the tri-state transceiver 65. These are Gate RAM and Enable Gate RAM signals shown in FIG. 8 at j and k. The Gate RAM signal is a directional a control signal to allow data to go from the G bus 72 to the RAM bus 67 (Gate RAM negative) or from the RAM bus 67 to the G bus 72 (Gate RAM positive). The Enable Gate RAM signal enables the transceiver 65 to pass the data between the two buses. Since the RAM WE signal is positive as shown in FIG. 8 at m, the data from the address specified by the address selector 69 is read from buffer 66 onto RAM bus 67 and via the transceiver 65 to G bus 72 from where it is transferred via driver 93 to the feature bus in 95. At the end of the SFBO signal, a

negative going step buffer address counter gate is generated as shown in FIG. 8 at l causing the address counter 68 to advance to the next address.

It it had been desired to read from the buffer 66 beginning at a specific address, it will of course be understood 5 that the read operation would have been first preceded by a write operation which loaded the starting address into the buffer address counter 68. Similarly, a write operation to the buffer 66 from the 3278 display would also be preceded by a write operation to load the start- 10 ing address into the buffer address counter 68.

The write operation is illustrated by a write operation to the buffer 66. Reference is now made to FIGS. 4 and 9. As before, parllel command bits on the feature bus out 89 are inputted to the tri-state buffer 91 as indicated 15 in FIG. 9 at c and d. Since as shown in FIG. 9 at h the Gate FBO signal is negative, bits 2 to 9 on the feature bus out 89 are transferred by the tri-state buffer 91 to the G bus 72. The 3278 display then causes the input control SFBO to go low as shown in FIG. 9 at a and this pro- 20 duces the negative gated SFBO signal shown in FIG. 9 at e that gates the command on the G bus 72 into the feature bus command register 74. The decoder 70 in response to the command data in register 74 provides a feature bus command signal to the feature control block 25 63. At this time in the example, it has been logically determined that a write to buffer 66 is to be performed. This time, however, the gate FBO signal remains negative and the gate FBI signal remains positive as shown in FIG. 9 at h and i, respectively, so that the tri-state 30 buffer 91 remains on and the tri-state driver 93 remains off. The feature response signal is sent to the 3278 display as shown in FIG. 9 at b and the gated SFBO signal goes positive as shown in FIG. 9 at e as in the preceding example. The Gate RAM signal shown in FIG. 9 at k 35 remains negative. This is the direction control signal to transceiver 65, and in response to this signal, the transceiver 65 allows data to flow from the G bus 72 to the RAM bus 67 when the transceiver 65 is enabled by a negative going Enable Gate RAM Signal. The first time 40 this occurs, the command on the G bus 72 is transferred to the RAM bus 67 but it is not written to buffer 66 because the RAM WE signal is positive as shown in FIG. 9 at m. On the next cycle, the data to be written to the buffer 66 is input to the tri-state buffer 92 as shown 45 at FIG. 9 at c. Note that at this point bit 11 on the feature bus out 89 goes high as shown in FIG. 9 at d to indicate that the bits 2 to 10 represent data and not a command. This time the RAM WE signal goes negative allowing the data on the data bus 67 to be written to the 50 buffer 66. At the end of the RAM WE signal, the step buffer address counter pulse shown in FIG. 9 at 1 is generated causing the buffer address counter 68 to supply the next address to the RAM address selector 69 in anticipation of the next write operation.

Consider now the poll request and response sequence with reference to FIGS. 4 and 10. As is common to all features which may be attached to the 3278 display, a poll request line is activated by the personal computer attachment when it has status to transmit. This is shown 60 in FIG. 10 at a. In order to prevent conflicts among several features which may be attached to the 3278 display, each feature is assigned a number (5 for the personal computer attachment in this specific example) and when a priority counter in the 3278 base unit 65 reaches the number assigned to a particular feature, a present poll signal is sent to the feature. This is shown in FIG. 10 at b. Furthermore, once the poll request line

has been activated by a particular feature, all other features are prevented from responding to a poll on the feature bus out 89. The poll request bit shown in FIG. 10 at a and the present poll signal shown in FIG. 10 at b cause the poll response signal shown in FIG. 10 at c to be generated and sent from the feature controls block 63 to the 3278 display. In response to the "hand shaking", the 3278 display then sends the poll command on the feature bus out 89 and the SFBO signal to the feature controls block 63, the latter being shown in FIG. 10 at d. As in the buffer read operation previously described, the gate FBO signal goes positive and the gate FBI signal goes negative turning off tri-state buffer 91 and turning on tri-state driver 93 thereby connecting the G bus 72 to the feature bus in 95. The poll command is decoded by the command decoder 70 and, in response to the decoded command, the feature controls block 63 generates the Gate ID or Poll Status signal to the poll-/ID response register 92 causing it to output its contents on the G bus 72. This data is transferred to the feature bus in 95 via the tri-state buffer 93 as shown in FIG. 10 at e. Finally, the 3278 display provides an acknowledge signal shown in FIG. 10 at f to the feature control block 63 to complete the "hand shaking".

Considering next the PC side of adapter 36, the PC status register 102 is connected to the T bus 104 and is one byte wide. This register is read only to the personal computer. It resides at I/O address X'3E4'. The following is a description of the bits of the PC status register beginning with the most significant bit:

BIT	DESCRIPTION
7	Reserved. Poll Request Active. The bit is
v	interrogated by the personal computer
	data transfer software to either allow
	the software to change the feature
	status register bits 0) or to prohibit
	the change of feature status register
	bits (1). It is set when any feature
	status register bit is initially turned
	on and the present poll signal from the
	feature bus is present and is turned off
	by the feature bus poll acknowledge.
5	Display Type. This bit is a 1 for a 3278
	and a 0 for a 3279 or similar color
	monitor.
4	Serial Port Busy. This bit is a 1 when
	the keyboard serializer is busy. It is
	reset when the port becomes available or
	with PC reset.
3	Serial Port Interrupt. This bit is a 1
	when the keyboard serial port has
	completed sending a character. It is
	reset by an adapter command or with PC
	reset.
2	Start Op Interrupt. This bit is a 1 when
	the Start Op Modifier register has been
	loaded from the feature bus. The bit is
	reset when the Start Op Modifier
	register is read by the PC or with PC
	reset.
1	Parallel Output Port Interrupt. This bit
	is a 1 when the parallel output port
	(8255 Port B) is available to accept
	another character. This bit is reset
	when data is sent to the port or by PC
	reset.
0	Keyboard In Interrupt. This is a 1 when
	the adapter has received a character
	from the keyboard (8255 Port A). This
	bit is reset when the data is read from
	the adapter or with PC reset.

The PC command decoder 106 is connected to the T bus 104 and is similar in construction to the feature command decoder 70 and so a detailed illustration of the decoder 106 is omitted from this description. The PC command decoder 106 is one byte wide and is write 5 only. This decoder resides at I/O address X'3E4'. The following is a list of the commands decoded by the decoder 106:

COMMAND	DESCRIPTION
X.00.	Turn off serial interrupt. Resets the
	keyboard serial 1/O port.
X.01.	Turn on Op Complete Pending. Turns the
	Op Complete Pending bit on and causes a
	poll request to the 3278.
X'02'	Turn off Read Request Sequence.
X'03'	Turn on Read Request Sequence.
X'04'	Turn off Screen Capture Busy.
X.02.	Turn on Screen Capture Busy.
X*06*	Turn off Test Mode.
X*07*	Turn on Test Mode.
X.08.	Turn off Keyboard Clicker Gate.
X'09'	Turn on Keyboard Clicker Gate.
X'0A'	Turn off 3278 Display gate.
X'0B'	Turn on 3278 Display gate.
X'0C'	Turn off IPL Complete.
X'0D'	Turn on IPL Complete.
X'0E'	Turn off Diagnostic Mode.
X*0F*	Turn on Diagnostic Mode.

The adapter 36 uses an Intel 8255 programmable parallel interface controller 64 to interface to the keyboard 12, the keyboard adapter 46, and the analog <sup>30</sup> switch control 56. For detailed information on the 8255 programmable parallel interface controller, the reader is referred to the *Intel Data* book. The controller 64 has four ports identified as Ports A, B, C and D. These ports, their I/O addresses and a description of their use <sup>35</sup> are listed below:

PORT	ADDRESS	DESCRIPTION	
A	X'3E0'	This port is used to read the keyboard.	
В	X'3E1'	This port is used to send keyboard data to the 3278 display.	
с	X'3E2'	This port has multiple uses. It is used for the control signal for ports A and B, the analog switch control, and control to select which input is read into port A (i.e. 3278 keyboard data or 3278 keyboard ident code).	4
D	X'3E3'	This is the control register for the 8255 controller.	

The Intel 8255 controller 64 is the main device used <sup>50</sup> to interface the keyboard. The Keyboard In at Port A of the 8255 controller is connected to the keyboard 12. The interface is asynchronous in that when the keyboard has a character to send, it activates the line "Data Avail In". The 8255 Port A is set for strobed input, and <sup>55</sup> when the "Data Avail In" line from the keyboard becomes active, the eight scan bits from the keyboard are loaded into Port A of the 8255 controller. The 8255 responds by activating the "KB Ack Out" line. When the software reads Port A of the 8255, the "Data Avail 60 Out" line from the 8255 will deactivate. The 8255 also generates an interrupt ("KB In Int") to the personal computer when a character has been read into the 8255 controller.

The keyboard data serialization register 110 is write 65 only and resides at I/O address X'3E5'. The data writen to this register will be serialized and sent to the PC keyboard input socket. After a keystroke has been read

in from the keyboard 12, the PC software determines where to send the data. If the attachment is in the PC mode, the keystroke will be mapped using a software lookup table to a corresponding PC keyboard scancode. In the specific example being discussed, scan code conversion is required because the 3279 keyboard 12 is encoded in a different scan code than the PC keyboard scan code. Obviously, in the host mode, no scan code conversion is required. Next, the new scan-code will be <sup>10</sup> output to the keyboard serializer 110. Writing this register causes PC Status Bit 4 (Serial Port Busy) to become active. When the serialization is complete, Bit 4 will be reset and PC Status Bit 3 (Serial Port Interrupt) will be set. The data sent to this port will be serialized in the <sup>15</sup> following order: First, a parity bit will be sent out. Second, the data will be sent out with the least significant bit sent first. This port is not bidirectional and therefore does not respond to the PC keystroke soft-

ware reset command. 20 FIG. 11 is a functional block diagram which summarizes the keyboard data paths just described. Note that the 8255 controller 64 operates as the two-way keyboard adapter 58 shown in FIG. 3. An important aspect of the invention is that if the personal computer applica-25 tion fails, then the operator can, using the keystroke sequence, switch to the host mode and continue the host operation. This is made possible by powering the front end of the two-way keyboard adapter directly from the 3278 display station. The front end of the two-way keyboard adapter is that section of the logic which accepts keystroke information from the keyboard and directs it either to the 3278 or to the personal computer. Additionally, if power is removed from the personal computer, then keystroke data is automatically redirected from the front end of the two-way adapter to the display station rather than to the personal computer without requiring the operator to invoke the switchover key sequencing so that host operation is main-40 tained.

The appendicies at the end of the specification are reproductions of the macro assembler print outs of the following software which facilitates the interrupt handling of the IBM Personal Computer attachment to an 45 IBM 3278 display station: Appendix A is the listing for the 3278 attachment BIOS (Basic Input/Output System) extension. Appendix B is the listing for the keyboard interrupt handler. Appendix C is the listing for the feature bus interrupt handler. Appendix D is the listing for the attachment BIOS end label. Appendix E is the listing for the 3278 attachment initialization. Appendix F is the listing for the EBCDIC keyboard translation tables and keyboard selection tables. The function of the 3278 attachment BIOS extension is to handle all hardware interrupts from the 3278 attachment and all software interrupts for the 3278 attachment functions. The function of the 3278 attachment keyboard interrupt handler is to process keystrokes from the 3278 keyboard and keyboard related function requests from an application program. The function of the 3278 attachment feature bus interrupt handler is to process all Start Operation commands from the 3274 controller and all 3278 attachment related function requests from an application program. The function of the 3278 attachment BIOS end label is to mark the physical end of the 3278 attachment BIOS extension. The function of the 3278 attachment initialization is to initialize the 3278 attachment and the 3278 attachment BIOS extension interrupt

handlers. These appendicies are intended to show an example of the software to support a specific preferred embodiment of the invention. Those skilled in the art will recognize that the software listed in the appendicies is tailored to the IBM Personal Computer used as an 5 attachment to the IBM 3278 display station. The attachment of a personal computer to a display station of different manufacture and having different characteristics would require different software support. In some cases, the software support needed will be simpler than 10 that of the specific embodiment disclosed. For example, the keyboard of the 3278 display terminal uses a scan code which is different than the scan code for the IBM Personal Computer keyboard which requires a translation between the two codes depending on what mode 15 the attachment is in. This translation is accomplished by table lookup in the memory of the PC system unit as described above. Obviously, when both the display station and the personal computer use the same keyboard code, typically ASCII, this translation function is 20 not required. Other simplifications and modifications will be apparant to those skilled in the art depending on the characteristics of the display station and personal computer which are to be attached.

Personal Computer are logically combined to generated gating, enable and address signals for reading from or writing to the RAM buffer 66 and various ones of the registers on the attachment adapter by the Personal Computer Feature. The logic that generates these sig- 30 nals is generally represented by block 71. Again, the specific details of the logic circuitry in the PC Controls and Address Decode Block 71 are shown in Appendix B of the IBM 3270 Personal Computer Attachment User's Guide. The inputs to the logic block 71 from the Per- 35 sonal Computer are the PC Clock, +Reset Driver, Input/Output Read, Input/Output Write, Memory Read, Memory Write, Address Enable, and the Address Bus A0-A19.

Consider first the operation of writing to the buffer 66 40 by the Personal Computer. The logic block 71 generates control signals to the tri-state transceiver 75 in order to connect the PC data bus 73 to the T bus 104. One of these signals, the +T Bus to PC signal, is a direction control signal, and the other, the Gate PC to T Bus 45 signal, enables the transceiver 75. Similar direction and enabling signals are generated for the tri-state transceiver 77. Specifically, the -T Bus to G Bus direction control signal allows data to flow from the T bus 104 to the RAM bus 67, and the -PC Select RAM signal 50 enables the transceiver 77. With the two transceivers enabled, a data path is established from the PC bus 73 to the T bus 104 and to the RAM bus 67. The -RAM WE(write enable) signal applied to the RAM buffer 66 then causes the data on the RAM bus 67 to be written into 55 the RAM buffer at the location specified by the RAM address selector 69. A RAM buffer read operation is similar except that the direction control signals and the -RAM WE signal are inverted.

In order to write to the PC command decoder 106, 60 the transceiver 75 is enabled as in a write to buffer operation, but the transceiver 77 is not enabled. The logic block 71 generates a -IOW (I/O write) signal and the address X'3E4' to the command decoder 106 to cause data on the T bus 104 to be written into the de- 65 coder. The PC Status Register 102 shares the address X'3E4' with the PC Command Decoder 106. This is possible since register 102 is a read only register while

the decoder 106 is write only. Therefore, to read the PC Status Register 102, the direction control signal to the transceiver 75 is inverted and the logic block 71 generates a -IOR (I/O read) signal and the address X'3E4' to the register 102.

The Keyboard Data Serializer 110 and the Start Op Modifier Register 112 also share an address, X'3E5', the former being write only and the latter being read only. The operations of writing to and reading from these registers are similar to those just described for the command decoder 106 and the status register 102. In addition to the address X'3E5', the logic block 71 generates the -IOW signal for the serializer 110 or the -IORsignal for the register 112.

The diagnostic register 114 contains the ten feature bus out bits as well as strobe feature but out, Poll Acknowledge, Present Poll, and Reset. This register can be written by the diagnostic program via the PC I/O bus. Two additional I/O addresses are decoded in the adapter for this purpose. With the feature bus cable disconnected, the diagnostic program is able to manipulate this bus (feature bus out) to simulate feature commands and data.

The diagnostic buffer 116 is attached to the feature Returning now to FIG. 4, the control signals from the 25 bus in (bits 2 to 20), Feature Response, Poll Request, and the analog switch control. The buffer 116 can be activated by the diagnostic program via the PC I/O bus. Two I/O addresses are decoded for this function. With the feature bus cable disconnected, the diagnostic program is able to detect feature responses and data.

Data transfer between the personal computer and the host program consists of file transfer and screen capture. File transfer is enabled by the 3274 controller 50 microcode which recognizes certain structured fields in the host data stream and directes the data to the feature bus 44 instead of the regeneration buffer 42 of the 3278 display station as shown in FIG. 3. The personal computer can accept the data and act on it immediately or store it on diskette for latter use with other PC programs. In addition, data can be taken from the diskette and sent via the feature bus 44 and the 3274 controller 50 to the host 48. This same ability to transmit data from the personal computer to the host permits a PC program to perform a function called "upload". With this function, an operator signs on to the host system in the normal manner. Then the operator can invoke a keystroke generation program in the personal computer. This program produces the keystrokes which are sent to the host 48 and appear to be macro-language instructions. These instructions are a bootstrap program which bring in the rest of the file transfer program. This file transfer program is a private exec which the operator now owns. It is permanently stored at the host 48 and available for use any time the operator invokes it. It only requires this upload once. Other uses can be envisioned for the automatic keyboard operations just described. For example, there could be automatic sign on or automatic generation of any keystroke sequence that the operator wishes to define.

Screen capture is the equivalent of the 3270 localcopy function and of the IBM Personal Computer printscreen function. Screen capture permits an operator to obtain a printed hard copy of the screen image or to copy the screen image to the system unit diskette. The only place within the display sub-system that the screen image resides is in the regeneration buffers. If the analog switch 52 currently connects the regeneration buffer 42 to the analog circuits 40, then the host data image is

15

displayed. The 3274 controller 50 has microcode to unload that screen image from the regeneration buffer 42 into the 3274 controller 50 and redirect it out through the feature bus 44 to the PC system unit 14. The system unit 14 then directs this screen image data to 5 either the printer 18 or a diskette drive 16 via the corresponding adapter. If on the other hand the analog switch 52 connects the output of a PC monitor adapter, either the color graphics adapter 26 or the monochrome monitor and parallel printer adapter 28, to the analog 10 circuits 40, then the image displayed is the PC program image. This image may be printed in the usual manner since it is already resident in the PC system unit. This capability for local copy also makes it possible for programs to be written to operate on the screen image in 15 lows: whatever way the programmer desires.

While the feature bus adapter physically resides in the personal computer, it appears to the 3274 controller as feature address '0101'B on the 3278 display. 3274 microcode and PC microcode communicate with each other 20 via the feature bus adapter. The Poll Response register 92 is used to signal the 3274 controller 50 that the PC status has changed. The Poll Response register 92 contains the following:

bits 0-3: Feature Address ('0101')

bit 4: Request Read Status

bits 5,6: reserved

bit 7: Operation Complete

The Poll Response register 92 can be read using a Read Feature ID command. The FBA Status register 80 is 30 The possible type fields flowing to the PC from the used to indicate the current state of the PC and the Feature Bus Adapter. The FBA Status register 80 contains the following:

bit 0: Screen Capture Busy

bit 1: FBA Interface Enabled

bit 2: FBA NOT Busy

bit 3: Request Read Operation

bits 4, 5: reserved

bit 6: Operation Complete Pending

bit 7: reserved

The Start Op Modifier register is used to tell the PC why it is being signaled by the 3274. The Start Op Modifiers include the following:

X'00'-reserved

X'01'-3278 POR

X'02'-Read Operation

X'03'-Write Operation

X'04'-Execute

X'05'-Abort Transfer (Abort-T)

X'06'-Abort to Last Verify (Abort-LV)

X'07'-Error

X'08'-Verify

X'09'-X'FF' are reserved

The address counter register 68 is used to tell the FBA hardware what buffer displacement to use when trans- 55 ferring data from/to the buffer on the next coax read/write command.

The messages between the 3274 and the PC are transferred via the RAM buffer 66. The format of the buffer 60 is as follows:

Each buffer passed to or from the attachment consists of a buffer header and data. The personal computer sends data to the 3274 that consist of the following:

1. File Transfer Data for the Host

Screen Capture Options Definition 2

3. Wrap Test Message

4. Query Keyboard State

The data sent to the PC attachment from the 3274 will consist of the following:

1. File Transfer Data from the Host

2. Screen Capture Data from the Display

3. Wrap Test Message

4. Keyboard State Information

The buffer header provides the necessary information for both the 3274 and the PC to know the amount of data in the buffer, the type of data, and other message indications. The format of the buffer header is as fol-

length = number of data bytes in the buffer (0-252)

type=message type (structured field, screen capture) flags=control flags

The length field is  $\overline{2}$  bytes. It contains the number of data bytes in the buffer. It does not include the buffer header. Byte 0 is the high order byte and byte 1 is the low order byte. The type field is 1 byte. Bit 7 signifies message direction (0=inbound, 1=outbound). The possible type fields flowing from the PC to the 3274 25 (inbound) are as follows:

X'00'-Wrap Test Request

X'02'-Inbound Structured Fields

X'04'-Define Screen Capture Options

X'06'-Query Keyboard State

3274 (outbound) are as follows:

X'01'-Wrap Test Reply

X'03'-Outboard Structured Fields

X'05'-Screen Capture Buffer Transfer

X'07'-Keyboard State

35

40

The flag field is also 1 byte. The bits are defined as follows:

Bit 0 = X'1' beginning of message

Bit 1 = X'1' end of message

Bits 2-7 = reserved

The objective of the file transfer function is to transfer files between a host application and the PC. The 3274 controller implements this objective by expanding its structured field support. The terminal operator initi-

45 ates a file transfer between a host application program and the PC by entering a message while in host mode. The host application program controls the transfer by issuing Write Structured Field (WSF) commands with FBA Structured Fields to the 3274 controller. The 3274 50 transfers the FBA Structured Fields to the PC for processing. The PC generates replies to the host application in the form of FBA Structured Fields and signals the 3274 that it has data to be read. The 3274 reads the

data and sends it to the host application. After one host message is sent to the PC, the host application will wait for a reply from the PC. The 3274 microcode changes required for file transfer are host system independent. The functions that require changes are as follows:

1. outbound data stream

2. inbound data stream

3. query reply

The Write Structured Field Data Stream is extended to provide support for File Transfer between the host and the PC. The FBA Structured Field is identified by the type of fields X'DO'. In host to PC (outbound) transfer, the structured fields except FBA Structured Fields are processed by the 3274 Data Stream Processor. The data contained in the FBA Structured Field is

30

65

not processed by the 3274. Instead, it is interpreted by the PC. When the 3274 detects a SF type of X'DO', data is passed to the PC through the FBA. A response is generated by the PC that is sent to the host. The FBA Structured Field is defined as follows:

BYTE	CONTENT	MEANING	
0-1	L	variable length	
2-3	x'DOXX'	FBA type	10
4-n	data	data	

The 3274 takes only the FBA structured fields, puts them into the FBA buffer 66, with a four byte buffer header, and issues a write start-op command for the PC 15 to get the data in the buffer. The buffer 66 holds 252 bytes of data with the 4 byte buffer header. The PC reads the data out of buffer 66 and puts the data into its own storage area. The start-op command indicates to the FBA the end of the DCA command sequence. A 20 modifier byte following the command indicates the reason for the start-op command, and the start-op command and modifier provides the 3274 control of the FBA buffer. The 3274 Outbound Data Stream Processor issues start-op commands to the FBA with the fol- 25 lowing modifiers:

- 1. X'03'-Write
- 2. X'04'-Execute
- 3. X'05'-Abort-T
- 4. X'06'-Abort-LV
- 5. X'08'-Verify

The WRITE Start-op is sent to the PC after the 3274 has written data into the FBA. The WRITE Start-op signals the PC that access to the buffer now belongs to the PC in order to remove the data placed there by the 35 3274. The PC has access to the buffer 66 until it indicates 'Op-Complete'. The EXECUTE Start-op is sent when the data in the WSF indicates End of Message and there are no transmission errors or data stream errors in the message. Execute is a signal to the PC that process- 40 ing may begin on the information received. The ABORT-T Start-op indicates to the PC that there was an error in the WSF data stream. The VERIFY Startop is sent to the PC when the 3274 detects an end-ofblock condition. The ABORT-LV Start-op indicates to 45 the PC that there was a transmission error.

For PC to host (inbound) data transfer, the PC sends an asynchronous request to the 3274 when it wants to send a FBA Structured Field to the host. The 3274 gets the data from the PC and puts it into its own line buffers 50 and sends the data to the host. The 3274 does not examine any of the data that is sent from the PC to the host. After the PC sends the 3274 an inbound file transfer message, the inbound transmission must be acknowledged before a new inbound operation can be per- 55 in resolution, modes and frequency of operation which formed. The 3274 will continue to issue Read Start-Op command queues to get more data into the FBA buffer 66 to be transferred to the host until the End of Message flag is on in the buffer header. The start-op command indicates to the FBA the end of the DCA command 60 sequence. A modifier byte following the command indicates the reason for the Start-Op command. The 3274 Inbound Data Stream Processor issues start-op. command to the FBA with the following modifiers:

1. X'02'-Read

2. X'04'-Execute

4. X'06'-Abort-LV

## 5. X'07'-Error

The READ Start-op is sent to the PC after the PC has sent asynchronous status to the 3274 requesting data to be sent to the host. On receipt of the Read Start-op, the PC has control of the buffer 66 and will put data into the buffer until it indicates Op-complete. The EXECUTE Start-op is sent to the PC when no errors are detected in the buffer header protocol. The ABORT-T Start-op is sent to terminate an inbound message operation. The ABORT-LV Start-op is sent to terminate an inbound message operation. The ERROR Start-op is sent to the PC if the 3274 detects an incorrect buffer header.

The objective of the screen capture function is to transfer the data visible on the 3278 display screen to the PC for printing on the PC printer or storing on a PC diskette. The terminal operator initiates a screen capture by pressing the print key while in host mode with the display's Print ID set for local copy to the PC. The 3274 microcode functions affected by screen capture include the following:

- 1. Print IDENT key
- 2. Print key
- 3. Host Initiated Copy
- 4. Device Cancel key
- 5. Host Loadable Printer Authorization Matrix (PAM) key sequence

The Print IDENT key is used by the operator to determine what the current printer assignment is or to assign a printer or a printer class based on the PAM. The Print key is used by the operator to initiate a local copy function. The host initiated copy function is invoked in SNA environment by the host sending a 3270 write type command to a display. The operator may use Device Cancel to cancel a current outstanding print request if the input is inhibited beacuse of a Device Busy condition. The host loadable PAM key sequence is used to update the cluster's PAM based on information transfer from the host.

The choice of the monochrome display and parallel printer adapter 28 and the color graphics adapter 26 depends on whether a graphics display function is desired. Either of these adapters may be used with the 3278 display which is itself a monochrome display. In the case where the color graphics monitor adapter 26 is used with a 3278 display, color signals are converted into a grey scale for display. If the color graphics monitor adapter 26 is used with the 3279 display or similar color monitor, the color conversion to grey scale is not required or used. The output characteristics of the monochrome display and parallel printer adapter 28 and the color graphics adapter 26 for the IBM PC are different. Both are designed around a Motorola 6845 CRT Controller module, but there are significant differences are summarized below:

	Monochrome	Color Graphics
$80 \times 25$ text mode	yes	yes
40  imes 25 text mode	no	yes
APA graphics mode	no	yes
character box size	$9 \times 14$	8 × 8
character size	7 × 9	$5 \times 7$
video frequency	18 MHz	7 MHz/14 MHz

Programming for the 6845 CRT Controller for the monochrome display adapter and the color graphics adapter is described on pages 2-45 and 2-60, respec-

<sup>3.</sup> X'05'-Abort-T

tively, of the IBM Technical Reference for the IBM PC, Revised Edition (July 1982). In the present invention, the following initialization parameters are loaded into the 6845 CRT Controller from the diskette containing the interrupt handler program for the monochrome 5 display adapter:

Register No.	Register File	Program Unit	Hex Code	
<b>R</b> 0	Horizontal Total	Characters	63H	
<b>R</b> 1	Horizontal Display	Characters	50H	
R2	HSync Position	Characters	52H	
R3	HSync Width	Characters	0FH	
R4	Vertical Total	Char Rows	19H	
R5	VTotal Adjust	Scan Line	00H	
R6	Vertical Display	Char Row	19H	
R7	VSync Position	Char Row	18H	
<b>R</b> 8	Interlace Mode	-	02H	
R9	MaxScan Line Addr	Scan Line	0DH	
<b>R</b> 10	Cursor Start	Scan Line	OBH	
R11	Cursor End	Scan Line	0CH	
R12	Start Addr (H)	-	00H	
R13	Start Addr (L)		00H	
R14	Cursor (H)	_	00H	
R15	Cursor (L)		00H	

The following initialization parameters are loaded into 25 the 6845 CRT Controller for the color graphics adapter with the register file and program unit being the same as above:

Register No.	40  imes 25 Alpha	80  imes 25 Alpha	Graphic Modes	<u> </u>
R1	38H	71H	38H	
R2	28H	50H	28H	
R3	2BH	57H	2BH	
R4	05 <b>H</b>	0AH	05H	35
R5	IAH	1 <b>AH</b>	6BH	
<b>R</b> 6	00H	00H	00H	
<b>R</b> 7	19H	19 <b>H</b>	64H	
<b>R</b> 8	19H	19H	64H	
R9	02H	02 <b>H</b>	02H	
<b>R</b> 10	07H	07H	01H	40
R11	06H	06H	06H	-
R12	07H	07H	07H	
R13	00H	00H	00H	
R 14	00H	00H	00H	
R15	00H	00H	00H	
R16	00H	00H	00H	44

In each case, only the first eight parameters are used for horizontal and vertical definitions. The second group of eight are the standard PC values. Additional details of both the monochrome display and parallel printer 50 adapter 28 and the color graphics adapter 26 may be had with reference to the aforementioned *Technical Reference* Manual for the IBM PC at pages 2-41 to 2-67. The analog input switch 52 is specifically designed to provide the interface between either of the 55 two adapters and the 3278 display.

Reference is now made to FIG. 12 which shows in detail the cable connections between the analog switch 52, the analog circuits 40, the digital logic and refresh buffer 42, and either the monochrome display adapter 60 28 or the color graphics adapter 26. Normally, the digital logic and refresh buffer 42 would provide to the analog circuits 40 horizontal sync signals (HR), vertical sync signals (VR), video data, intensity level, NC Reg 1 and vertical skip signals. When in the host mode, the 65 switch 52 is transparent and passes these signals directly to the analog circuits 40. The monochrome display adapter 28 also produces horizontal sync, vertical sync,

video and intensity (INT) signals which, in the PC mode, are modified by the analog switch 52 and passed to the analog circuits 40. The color graphics adapter 26 in addition produces red (R), green (G) and blue (B) color signals which, for the 3278 display, are converted to grey scale signals by the analog switch 52 and passed to the analog circuits 40.

The first function that is performed by the analog switch 52 is to monitor the presence of data on the PC 10 data line in the video cable to determine whether the monochrome display adapter 28 or the color graphics adapter 26 is installed in the PC system unit 22. The monitoring logic is shown in FIG. 13B and comprises a monitor latch 120 which is set by the video signal from 15 the PC adapter and reset by the vertical sync signal. As shown in FIG. 14, the PC video input (monochrome) signal from a monochrome display adapter is driven to a negative state prior to, during, and following the PC VR vertical sync signal. As a result, latch 120 is inhib-20 ited from being reset and provides a positive or high level output. When driven from a color graphics adapter, the PC video line floats and is held at +5 volts by a pullup resistor on the switch card. As a result, latch 120 is reset on the negative going edge of the PC VR vertical sync pulse. Since the latch 120 is an edge triggered device, it will not again be set and provides a negative or low level output. Thus, the latch 120 is always held in the appropriate state depending on which adapter is driving the cable.

The output of latch 120 is inverted by an inverter 122 to provide a monochrome latch control output. This control is described in more detail hereinafter but briefly summarizing, the control in combination with the PC select signal provides the following functions:

I. Monochrome and PC Select

- A. Disables the Color to Greyscale Translator
- B. Enables the Video Data and High Intensity data lines to the analog circuits
- C. Activates the -NC Reg 1 line at the analog circuits. This enables the 3278 Model 5 analog card to adapt its horizontal and vertical scan times to match the requirements of the PC monochrome adapter.
- II. Color Graphics and PC Select
  - A. Enables the Color to Greyscale Translator
  - B. Disables the Video Data and High Intensity Data lines to the analog circuits
- C. De-activates the  $-N\bar{C}$  Reg line at the analog circuits to revert to the 3278 Model 2 ramp times which match the horizontal and vertical scan times of the color graphics adapter

The output of latch 120 is also used for video switching. Specifically, the output of latch 120 is connected to enable AND gate 124 which is selected by the PC Select signal to pass the video signal from the monochrome adapter. The output of AND gate 124 is supplied to driver 126 via OR gate 128. The other input to OR gate 128 is supplied by AND gate 130 which is selected by the inverse or NOT PC mode signal to pass the 3278 video signal. The output of driver 126 is connected to the analog circuits.

The analog switch also provides switching for the horizontal sync and vertical sync signals from both the 3278 digital logic and the PC attachment. As shown in FIG. 13A, the horizontal sync switching logic comprises a pair of AND gates 132 and 134. The horizontal sync signals from the PC are supplied to AND gate 132, while the horizontal sync signals from the 3278 logic are

4,641,262

supplied to AND gate 134. The PC Select signal from the output of inverter 136 shown in FIG. 13C is used to select AND gate 132, and the inverse or NOT PC Select signal is used to select AND gate 134. The outputs of AND gates 132 and 134 are supplied via OR gate 138 5 to driver 140. The output of dirver 140 is supplied to the analog circuits 40. Similarly, the vertical sync switching logic comprises a pair of AND gates 131 and 133, an OR gate 139 and a driver 143. The AND gate 131 is used to select PC vertical sync pulses, while the AND 10 gate 133 is used to select the 3278 vertical sync pulses. However, as will be described in more detail hereinafter, the PC vertical sync pulses supplied to the input of AND gate 131 are first stretched to match the input requirements of the 3278 display. In like manner, the 15 intensity switching logic comprises a pair of AND gates 135 and 137, an OR gate 141 and a driver 145. The AND gate 135 is used to select the PC intensity signal, while the AND gate 137 is used to select the 3278 high intensity signal. The switching logic further includes 20 AND gates 144 and 146, OR gate 147, inverter 148 and AND gate 149, the latter two being shown in FIG. 13B. AND gate 144 is selected by the PC Select signal, while AND gates 146 and 149 are selected by the inverse or NOT PC Select signal. The other input to AND gate <sup>25</sup> 144 is the monochrome latch control output from inverter 122 in FIG. 13B. The other input to AND gate 146 is the NC Reg 1 signal from the digital logic and refresh buffer 42. The output of OR gate 147 is supplied to the analog circuits 40. In FIG. 13B, the Skip signal <sup>30</sup> from the digital logic and refresh buffer 42 is supplied to the other input of AND gate 149 via inverter 148. The output of AND gate 149 is supplied to the analog circuits 40. Thus, the video switch logic presents the 3278 or PC sync and video signals to analog circuits 40 under 35 the control of the PC Select signal received from the PC on the feature bus cable. A PC Select signal will cause data from the PC buffer to be displayed or data from the 3278 buffer to be displayed on the 3278 screen. The status of the PC Select signal and hence the video switch logic is controlled by the operator by entering a specific key sequence such as, for example, the "Alternate" key with the "Insert" key. Successive operations of this key sequence will toggle the screen between 45 3278 buffer data and PC buffer data.

The following table summarizes the data or signals received by the analog switch 52 and presented to the analog circuits 40:

		50
	Signal Names Presented To 3278 Analog Circuits	
Signal Names Received From 3278 Base Logic		
+ 78 Horizontal Sync + 78 Vertical Retrace + 78 Hi Intensity + 78 Video Data - 78 Vertical Skip - 78 NC Reg 1	+ Horizontal Sync + Vertical Retrace + Hi Intensity + Video Data - Vertical Skip - NC Reg 1	5:
Signal Names Received From Color Graphics Adptr		60
+ PC Horizontal Sync + PC Vertical Sync + PC Red + PC Green	+ Horizontal Sync + Vertical Sync	
+PC Blue +PC Hi Intensity Signal Names Received From Monochrome Adapter	- PC Grey	6
+PC Horizontal Sync	+ Horizontal Syne	

7		
4	-	

-0	continued
	Signal Names Presented To 3278 Analog Circuits
- PC Vertical Sync + PC Video + PC Hi Intensity	+ Vertical Sync + Video + Hi Intensity
	- NC Reg 1

In order to match the PC display adapters to the 3278 display, it is necessary to stretch the vertical retrace signal provided by the monochrome display adpater 28 or the color graphics adapter 26. The logic for doing this is shown in FIG. 13C. Both the PC adapters 26 and 28 provide a pulse of approximately 200 microseconds duration, the pulse from adpater 26 being positive going while the pulse from adapter 28 is negative going as shown in FIG. 15. What is required by the 3278 display is a pulse of approximately 450 to 500 microseconds duration. The pulse stretching logic shown in FIG. 13C receives the vertical and horizontal sync signals from the adapters 26 or 28. The vertical sync signal is supplied to the set input of an edge triggered latch 152 via an inverter 154. The latch 152 is therefore set on the leading edge of the vertical sync signal from adapter 26 and on the trailing edge of the vertical sync signal from adapter 28. When set, the latch 152 enables AND gate 156 to pass the horizontal sync pulses. Counter 158 counts the horizontal sync pulses to a predetermined count of 8 to allow for a stretched vertical retrace signal. When the predetermined count is reached, AND gate 160 produces a reset pulse via inverter 162 to reset latch 152 and counter 158. The output of latch 152 is therefore a stretched vertical retrace pulse of approximately 480 microseconds when adapter 26 is used and approximately 450 microseconds when adapter 28 is used as shown in FIG. 15.

When the color graphics adapter 26 is used with the 3278 display, the RGB color signals from the adapter are converted to greyscale signals which can be accomplished with the logic shown in FIG. 13C. The red, green, blue and high intensity signals received on the video cable from the PC are passed by respective AND gates 164, 166, 168 and 170 to corresponding drivers 172, 174, 176 and 178. The AND gates are enabled by a PC Select signal from inverter 136. The drivers are enabled by a greyscale gate signal produced by retrace blanking logic described hereinafter. The outputs of the drivers are fed to corresponding binary weighted resistors 182, 184, 186 and 188 which, together with sum-50 ming resistor 190, form a digital-to-analog converting resistive ladder network. Thus, for the sixteen possible binary combinations input, sixteen different analog levels are produced at the -PC Grey output which is 5 supplied to the intensity control of the analog circuits 40.

The greyscale gate signal is generated by the logic shown in FIG. 13C. First of all, the greyscale gate signal is inhibited by the monochrome latch control o signal from inverter 122 shown in FIG. 13B. This signal is provided as an input to NAND gate 192 which, in combination with inverter 194, functions as an AND gate. This, a low input from inverter 122 indicating that a monochrome adapter is installed will produce a low

5 output from inverter 194 disabling the drivers 172, 174, 176 and 178 shown in FIG. 13B. The other function performed by the retrace blanking logic is to blank the boarder area produced by the color graphics adapter 26 during horizontal retrace periods. This is accomplished by latch 196 which is set by the output of inverter 198. The input to inverter 198 is the PC horizontal retrace signal which also enables four stage counters 200 and 202. Counter 200 is clocked by the PC clock, and 5 counter 202 is clocked by the output of counter 200. AND gate 204 functions as a decoder to detect a count of thirty-four and produce a reset pulse to latch 196. The counters 200 and 202 are reset by the trailing edge of the horizontal retrace pulse. When latch 196 is set, it 10 produces a low level output to NAND gate 192 resulting in a low level output from inverter 194. A low level output from inverter 194 is also produced by the stretched vertical retrace signal from latch 152 after inversion by inverter 206.

25

Those skilled in the art will appreciate that what has

been disclosed as a preferred embodiment and the best mode contemplated for the practice of the invention defined in the appended claims is not to be construed to limit the invention to the specifically disclosed hardware or software. As pointed out in the specification, simplifications can be achieved if the display station and the personal computer normally use keyboards that use the same scan codes. Other simplifications can be achieved if the display station and the personal computer are fully compatible. Still other simplifications can be made if not all the data transfer functions are implemented. Moreover, those skilled in the art will recognize that different hardware and software than that which is specifically disclosed
15 could be used to perform the same or similar functions.

#### AFPENDIX A

The IBM Personal Computer MACRD Assembler 3278/3279 Attachment BIOS Extension

34

22 23 24

2567890133333333444444 33333333444444

65 66

i	; #************************** Start of Specifications ***************************
	Module Hame: 327DATT
	Descriptive Name: 2078/3079 Attachment BIDS extension
	Copyright: IBM Corp 1982
	Status: Version 1.00
	Notes:
	Dependencies: DDS 1.1 or later
	Restrictions: None
	Module Type: Program
	Purpose: Copyright notice displayed when BIOS is initialized
	Linkage: None
	Input Parameters: None
1	Entry Point: NDSP_INT
;	Purpose: To process all hardware interupts from the 3278/3279 attachment option
	Linkage: Hardware interrupt 2
1	Input Parameters: None
	Exit Normal:
	Purpose: To return to the interrupted program
1	Linkage: IRET
;	Output: None
;	Exit Error: None
, ; ;	
1 1	Procedure Invoked: PROC_KEYEDARD_DATA
;	Purpose: To process a keystroke from the 3278/3279 keyboard
;	Linkage: CALL
;	Input Parameters: None

67 68971277777777890 Procedure Invoked: PROC\_PC\_PORT\_AVAIL Purpose: To send the next waiting keystroke to the Personal Computer Linkage: CALL Input Parameters: None Output: None Procedure Invoked: PROC\_3278\_PORT\_AVAIL Purpose: To send the next waiting keystroke to the 3278/3279 81 82 83 Linkage: CALL 84 85 Input Parameters: None 86 87 Output: None 86 89 Procedure Invoked: PROC\_START\_OP 90 91 92 Purpose: To process a Start Operation command received from the 3274 control unit 93 94 Linkage: CALL 95 96 97 Input Parameters: None 98 99 Output: None 100 Control Blocks: 101 NDSPDCLS - 3278/3279 adapter interfaces NDSPUSER - 3278/3279 attachment BIOS software interfaces 102 103 104 105 Messages Issued: None ; 106 ; Change Activity: None 108 ; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Specifications \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 109 110 SUBTTL 3278/3279 Attachment Option Interfaces 111 112 THOUSE HOSFDOLS.ASH 113 3 114 С 115 00000000000 116 ;× 117 ;× 1/0 ADDRESS ASSIGNMENTS 118 ;\* 119 120 3 BASE I/O ADDRESS FOR NDSP ADAPTER 121 = 03E0 BASE ADDR EQU DEEDH NDS\_KB\_IN NDS\_KB\_OUT KB\_CTL 3278/3279 KEYBOARD INFUT PORT 3278/3279 DISPLAY OUTFUT PORT 122 = 03E0 EQU BASE\_ADDR+0 EQU BASE\_ADDR+1 EQU BASE\_ADDR+2 123 = 03E1 124 KEYBOARD CONTROL PORT = 03E2 EQU EASE\_ADDR+3 EQU BASE\_ADDR+4 : 8255 COMMAND PORT 125 = 03F3 С CMD\_8255 ; NDSP ADAPTER STATUS FORT = 03E4 STATUS\_PORT 126 с с ; NDSP ADAPTER COMMAND PORT ; START OPERATION MODIFIER PORT EQU BASE\_ADDR+4 EQU BASE\_ADDR+5 127 = 03E4 COMMAND\_PORT 128 = 03E5 START\_OP\_MOD PC\_KB\_OUT 0000 F PERSONAL COMPUTER KEYBDARD OUTPUT PO 129 = 03E5 EQU BASE\_ADDR+5 130 ; 8259 INTERRUPT CONTROLLER PORT : 8259 INTERRUPT MASK REGISTER 131 CTL\_8259 EQU = 0020 20H 132 = 0021 С IMR\_8259 EQU 21H 133 с с 134 135 00000000 136 ;× VALUES TO BE SENT TO KEYBOARD CONTROL PORT (KB\_CTL) 137 ;\* 138 1 # \* 教士全法不需要要要承承在某些事实是是法法承承还在常长来这个成本的要求我们有关的法法规则没是没有不能不是不是有不能有不能要要要要有法则。 139 140 SELECT\_DATA EQU OOH ; SELECT KEYBOARD DATA ; SELECT KEYBOARD IDENTIFIER 141 = 0000 142 = 0040 с SELECT\_1D EQU 40H 143 c 144 145 С \$ 最优美成学家最多成果在实际发展发展不同的不是不能不是不是有些有些有效的不能是不能是不能不能不能不能不是不是有这个不是有些不能有这个。 С c 146 ;× VALUES TO BE SENT TO 8255 COMMAND PORT (CHD 8255) 147 3× 148 С ;\* 3 我家家我们参考家我不是我这些承认我就就找不能需要我就把你要要做我就要我就要我就要我们不能不能不能要不能要要要要要做我有 149 C 150 00000 ; NO OUTPUT CHARACTER AVAILABLE ; DISABLE INTERRUPT FROM 3278/3279 151 = 0003 NOT\_OBF EQU 0 3 H RESET\_INTE\_B EQU 04H SET\_INTE\_B EQU 05H 152 = 0004 ; ENABLE INTERRUPT FROM 3278/3279 ; DISABLE INTERRUPT FROM 3278/3279 ; KEYBOARD 153 = 0005 = 0008 RESET\_INTE\_A EQU OBH 154 155 C ; ENABLE INTERRUPT FROM 3278/3279 156 = 0009 c SET\_INTE\_A EQU 0.9H ; KEYBOARD ; SWITCH 3278/3279 TO PERSONAL COMPUTER 157 C = 000E PC MODE EQU OEH 158 С 159 HODE SHITCH 3278/3279 TO HOST PROCESSOR C HOST HODE EQU OFH = 000F 160

.

		29		30	
161		6		MODE	
		C		; MODE	
162		C			
163		c			
164		C :*********	******	我就跟这些我这是我就能能没有不能能要要要做这个的事,我们都没有不能有多多多少?"	**1
165		C ;*			
166			ES TO BE SENT	TO INTERRUPT CONTROLLER PORT (CTL_8259)	
167		C ;*			
168				*************************************	***
169		C			
170	= 0062	C EOI EQU	62H	; END OF INTERRUPT FOR LEVEL 2	
171		C			
172		C			
173		· C			
174			*************	法武法官官 化化合金 化合金化合金 化合金化合金 化合金化化合金 化化合金化化合金 化合金化合金	***
175		-			
176					
			ES TO BE SENT	TO NDSP COMMAND PORT (COMMAND_PORT)	
177		C ;*			
178		C ;*************	********	浓草浆油带浆和浓油浓浓浓浓浓浓浓浓浓浓浓浓浓浓浓浓浓水和浓水水不可不能发发发发发发发	***
179		C			
160	= 0000	C SERIAL INT_D	FF EQI	U DOH ; TURN OFF SERIAL FORT INTERRUPT FLA	G
181	= 0001	C OP_COMPLETE		U DIH ; OPERATION COMPLETE	
182	= 0002	C READ COMPLETI	- FO	U 02H ; 3274 CAN READ DATA FROM HOSP ADAPT	FD
	- 0002	C REAU_CUMPLEM			
183		C		; BUFFER	
164	= 0003	C READ_REQUEST	EQ	U 03H ; REQUEST READ FROM 3274	
185	= 0004	C SCREEN_CAPTUR	RE_COMPLETE EG	U 04H & SCREEN CAPTURE COMPLETE	
166	= 0005	C SCREEN CAPTUR	EQ:	U 05H ; SCREEN CAPTURE IN FROCESS	
187	= 0006	C DIAGNOSTIC M	DE GEE FOI	U 06H ; TURN OFF DIAGNOSTIC MODE U 07H ; TURN ON DIAGNOSTIC MODE	
165	= 0007	C DIAGUOSTIC H	10F CH F C	I ATH : THEN ON DIAGNOSTIC MODE	
169	= 0008	C CIRCOUSIIL_IN		N AND A THEM ALE TATE /TATE HOULD BE THERE	,
190		C CLICK_DFF C CLICK_ON	EW	U USH ; TURN OFF 5278/5279 RETBUARD CLICKER	C
	= 0009	C CLICK_ON	EGI	U 08H ; TURN OFF 3278/3279 KEYBOARD CLICKER U 09H ; TURN 0H 3278/3279 KEYBOARD CLICKER U 08H ; DISABLE 3278/3279 DISPLAY INTERFACE U 08H ; ENABLE 3276/3279 DISPLAY INTERFACE	
191	A000 =	C DISABLE_NDS C ENABLE_NDS	EQ	U GAH ; DIGABLE 3278/3279 DISPLAY INTERFACE	
192	= 000B	C ENABLE NDS	EQ	U OBH ; ENABLE 3276/3279 DISPLAY INTERFACE	
193	= 0000	C RESET TPL COM	PLETE FOI	U OCH ; NDSP ADAPTER CODE NOT INITIALIZED U ODH ; NDSP ADAPTER CODE INITIALIZED	
194	= 0000	C TRI COMBLETE		U ONN : NOSP ADAPTED CODE THITTALTED	
195	- = 000E	C RESET ADAPTER		U OEH ; RESET 3278/3279 ADAPTER	
196				U VEN ; RESEL S2/6/32/9 AVAPIER	
		C			
197		C 24444444444444	**********	<b>家家家家家家家家家家家家家家家家家家家家家家家的人名匈米米卡鲁尔卡卡卡卡卡卡卡卡</b>	***
198		C ;*			
199		C ;* HASK	VALUES FOR NO	SP ADAPTER STATUS (READ FROM STATUS_PORT)	*
200		C ;*			*
201				<b>张雅家爱爱教圣教圣教教教教教教教教教教教教教教教教教教教教教教教教教教教教教教教教</b>	***
202					
203	= 0001	c			
		C ST_KB_IN	EQU 011		
204	= D002	C ST_KB_OUT	EQU 021	H 3278/3279 KEYBOARD OUTPUT PORT	
205		C		; AVAILABLE	
206	= 0004	C ST_START_OP	EQU 041	H ; START OPERATION FROM 3274	
207	= 0008	C ST_PC_KB_OUT	EQU DB		TROP
208		c 51_1C_KD_001	240 001		•
209		C		; AVAILABLE	
	= 0010	C ST_PC_KB_BUSY	EQU 10		VOR 1
210		С		; BUSY	
211	= 0020	C ST_3278	EQU 201	i ; DISPLAY IS A 3278	
212	= 0040	C ST_POLL_ACTIV	E EQU 401	<pre>i ; FOLL REQUEST IS ACTIVE</pre>	
213		c			
214		č			
215					
			************	<b>K 教授教授 我 我 我 广告 的 外子 的 子子 的 外子 的 外子 的 外子 的 外子 的 外子 的 外</b>	
216		C ;*			*
217		C 1* START	OPERATION MOD	DIFIERS (READ FROM START_OP_MOD)	¥
218		C ;*			*
219		C ;***********	*****	*******	****
220		c ,			
221	= 0001	C SOM_POR	EQU 01H	1 ; 3276/3279 POWER ON RESET	
222	= 0002				
		C SOM_READ	EQU 02H		
223	= 0003	C SOM_WRITE	EQU 03H		
224	= 0004	C SOH_EXECUTE	EQU 04H		
225	= 0005	C SOM_ABORT_E	EQU 05H	ABORT TRANSFER COMMAND	
226	= 0006	C SOM_ABORT_V	EQU 06H	ABORT TO LAST VERIFY COMMAND	
227	= 0007	C SOM_ERROR	EQU 07H		
228	= 0008	C SOM_VERIFY	EQU 08H		
229			240 000	· · · · · · · · · · · · · · · · · · ·	
		c			
230		c			
231		C			
232			******	· 那就这就就这些我就不可能没有什么?" 化卡卡卡卡卡拉 的复数的现在分词 化化合金化化合金化合金	***
233		C ;*			*
234			ARD TRANSLATTO	IN TABLE VALUES - SPECIAL FUNCTION KEYS	*
235		C ;*			
236			*************	(你跟我就给我我我有了。 卢武学汉史文大大大利产者名爱尔斯尔文名意大利名大名中有多有多名	
237		C			
238	=-0001	C SHIFT_LOCK	EQU -1	; SHIFT LOCK KEY	
239	=-0002	C L_SHIFT	EQU -2	; LEFT HAND SHIFT KEY	
240	=-0003	C R_SHIFT	EQU -3	; RIGHT HAND SHIFT KEY	
241	=-0004	C NUM_LOCK	EQU -4	NUMERIC LOCK KEY	
242	=-0005	C NUM_SHIFT	£Qป -5	; NUMERIC SHIFT KEY	
243	=-0006	C ALPHA_SHIFT	EQU -6	; ALPHAMERIC SHIFT KEY	
244	=-0007	C ALT_SHIFT	EQU -7	; ALT KEY	
245	=-0008	C RESET	EQU -8	; RESET KEY	
246	=-0009	C CLICK	EQU -9	; CLICK CH/OFF KEY	
247	=-000A	C DEC_INP	EQU -10	; ST/RT DECIMAL INPUT KEY	
248	=-000B	· •			
			EQU -11	; TOGGLE HODE KEY	
249	=-0000	C CTL_KEY	EGU -12	; ENTER FERSONAL COMPUTER CTRL CASE	
250	=-000D	C ALT_KEY	EQU -13	; ENTER PERSCHAL COMPUTER ALT CASE	
251	=-000E	C PAUSE KEY	EQU -14	; PAUSE UNTIL ANOTHER KEY IS HIT	
252	= 0007	C NO_SHIFT_KEYS		; NULBER OF SHIFT KEYS DEFINED	
253			L40 /	, NUMBER OF CHIFT REFS DEFINED	
		c			
254		C			

.

•

	4,641,262
31	32
	— — — — — — — — — — — — — — — — — — —
	C :* C :* KEYBOARD TRANSLATION TABLE VALUES - REGUIRED STATES *
	C - 3###################################
0000	C L EQU OOH ; LDWER CASE C U EQU GOH ; UPPER CASE
0040	C C EQU 40H 3 CTRL CASE
020	C A EQU 20H ; ALT CASE C BA EQU 02H ; INDICATE BREAK ALT AFTER KEYSTROKE
001	C BA EQU OCH ; INDICATE BREAK ALT AFTER REISTROKE C B EQU OTH ; MUST ALSO INDICATE EPEAK FOR THIS
-	C ; KEYSTROKE
	C C
	C
00	C FBA_BUFFER SEGMENT AT 0E400H C
	C 」这些学家就是这些爱爱的是你们的是是这些没有这些没有这些没有这些没有这些没有不是不是我们的这些不是不是不是我们的这些不是不是不是有些的
	C ;* C ;* FEATURE BUS ADAPTER BUFFER
	C ] ###################################
000 ????	C C HDR_LENGTH DH ? ; LENGTH OF DATA
2 77	C HDR_TYPE DB ? I MESSAGE TYPE
3 ?? 080	C HDR_FLAGS DB ? ; MESSAGE FLAGS C HDR_F START FOU ANH : START OF MESSAGE
8040	C HDR F START EQU 80H ; START OF MESSAGE C HDR F END EQU 40H ; END OF MESSAGE C HDR DATA DB 252 DUP(?) ; TEXT OF MESSAGE
104 FC 1	C HDR_DATA DB 252 DUP(?) ; TEXT OF MESSAGE
??	C C
	C
0100	C C SIZE_FBA_BUFFER EQU \$-FBA_BUFFER ; SIZE OF FBA BUFFER
100	C FBA_BUFFER ENDS C
	C
	C 2# C 3#*###################################
	C ;* TYPE CODES IN FEATURE BUS MESSAGE HEADER
	C ;* C ;**
	C ;************************************
	C ; MESSAGES SENT TO THE 3274
000	C C HT_WRAP_TO_3274 EQU 00H ; KRAP TEST MESSAGE TO 3274
002	C HT_SF_TO_HOST EQU 02H ; STRUCTURED FIELD TO HOST
0004 0006	C HT_SET_SC_OPTIONS EQU 04H ; SET SCREEN CAPTURE OPTIONS C HT_QUERY_KBD_STATE EQU 06H ; QUERY KEYBOARD STATE
008	C HT_ERROR_LOG EQU OBH ; ERROR LOG HESSAGE TO 3274
	C
	C
0001	C HT_WRAP_FROM_3274 EQU 01H ; KRAP TEST MESSAGE FROM 3274 C HT_SF_FROM_HOST EQU 03H ; STRUCTURED FIELD FROM_HOST
0003 0005	C HT SC DATA EQU 05H ; SCREEN CAPTURE DATA
0007	C HT_KBD_STATE EQU 07H ; KEYBDARD STATE
	C C ; DEVICE TYPE CODES
	C
0001	C HT_ID EQU 01H ; 3278/3270 PERSONAL COMPUTER ATTACHMEN C
	SUBTTL 3278/3279 Attachment BIOS Extension Interfaces
	C INCLUDE NDSPUSER 251
	C INCLUDE RUSPUSER / 25/1
	() · · · · · · · · · · · · · · · · · · ·
	C ;# C ;# HDSP USER INTERFACES
	C 3*
	C ;************************************
050	C C NDSP_INT_CODE EQU 50H ; NDSP SOFTWARE INTERRUPT CODE
	C C
	C ; " **********************************
	C ;#
	C ;* FUNCTION CODES - PLACED IN REGISER AN BEFORE INVOKING BIOS
	C ;* FUNCTION CODES - PLACED IN REGISER AH BEFORE INVOKING BIOS C ;* C ;******
0000	C ;* FUNCTION CODES - PLACED IN REGISER AH BEFORE INVOKING BIOS C ;* C ;*******************************
	C ;* FUNCTION CODES - PLACED IN REGISER AH BEFORE INVOKING BIOS C ;* C ;*******************************
0001 0002	C ;* FUNCTION CODES - PLACED IN REGISER AH BEFORE INVOKING BIOS C ;* C ;*******************************
- 0000 - 0001 - 0002 - 0003 - 0004	C ;* FUNCTION CODES - PLACED IN REGISER AH BEFORE INVOKING BIOS C ;* C ;*******************************
0001	C ;* FUNCTION CODES - PLACED IN REGISER AH BEFORE INVOKING BIOS C ;* C ;*******************************
0001 0002 0003 0004 0005 0006	C ;* FUNCTION CODES - PLACED IN REGISER AH BEFORE INVOKING BIOS C ;* C ;*******************************
0001 0002 0003 0004 0005	C ;* FUNCTION CODES - PLACED IN REGISER AH BEFORE INVOKING BIOS C ;* C ;*******************************

•

		4,641,262
	33	34
348	c	
349 350	C	
351	c	; N *
352 353	C	
354		
355 356	≈ 0000 r	
357	= 0001	SET_MODE_ROST     EQU D ; CHANGE TO HOST PROCESSOR MODE       SET_MODE_PC     EQU 1 ; CHANGE TO PERSONAL COMPUTER MODE       SET_MODE_HOST_SCREEN     EQU 2 ; DISPLAY HOST SCREEN IMAGE       SET_MODE_PC_SCREEN     EQU 3 ; DISPLAY PC SCREEN IMAGE
358 359	= 0002 C	SET_MODE HOST_SCREEN EQU 2 ; DISPLAY HOST SCREEN IMAGE
360	- 0003 C	SEI_HODE_PC_SCREEN EQU 3 ; DISPLAT PC SLREEN IMAGE
361	c	
362 363	C	
364	C	
365 366	C	
367	C C	
368	c	
369 370	= 0000 C	
371	= 0001 C	VECTOR_SCREËN_CAPTURE EQU 1 ; SCREEN CAPTURE
372 373	= 0002 C	VECTOR_MRAP_TEST EQU 2 ; WRAP TEST VECTOR_QUER"_KDD_STATE . EQU 3 ; QUERY KEYBOARD STATE (USED INTERNALLY)
374	C	
375 376	C	
377	c c	
378	C	* RETURN CODE (IN AL REGISTER) FROM NOSP BIOS *
379 380	c c	
381	-	
382 383	= 0000 C = 0001 C	RC_DK     EQU     0 ; FUNCTION PERFORMED       RC_INVALID_FUNC     EQU     1 ; FUNCTION CODE INVALID       RC_INVALID_HODE     EQU     2 ; HODE SELECTION CODE INVALID       RC_FBA_NOT_OP     EQU     3 ; FEATURE BUS ADAPTER NOT OPERATIONAL       RC_INVALID_KEY     EQU     4 ; INVALID KEY STROKE IN MESSAGE       RC_INVALID_VECTOR     EQU     5 ; INVALID INTERRUPT VECTOR SPECIFIED       RC_POMER DN RESET     EQU     6 ; MESSAGE CANCELLED BY 3276/3279/3274
354	= 0002 C	RC_INVALID_MODE EQU 2 ; MODE SELECTION CODE INVALID
385	= 0003 C	RC_FBA_HOT_OP EQU 3 ; FEATURE BUS ADAPTER NOT OPERATIONAL
386 387	= 0004 C = 0005 C	RC INVALID_RET EQU 4 ; INVALID RET STRUKE IN HESSAGE
388		
389 390		; POWER ON RESET RC_FRDTOCOL_VIOLATION EQU 7; MESSAGE REJECTED BY 3274 DUE TO
391	c	; PROTOCOL VIOLATION
392 393		RC_MESSAGE_CANCELLED EQU 8 ; MESSAGE CANCELLED BY USER REQUEST RC_INVALID_PARM EQU 9 ; INVALID_PARAMETER
394	= 0004 C	
395	C	
396 397	= 000B C	RC_KEYFOARD_INHIBITED EQU 11 ; 3C78/3C79 KEYBOARD IS INHIBITED ; FOR SCHE DIHER REASON
398	= 000C C	RC_MITCORED_LIMITED       ;       FOR SCHE DIHER REASON         ;       FOR SCHE DIHER REASON         RC_AID_SENT       EQU 12 ; 3272/3279 KEYBOARD IS NOW INSTELLED         ;       FECLUSE AN AID WAS SENT         RC_MESSAGE_LOST       EQU 13 ; 3278/3279 KEYBOARD IS NOW INHIBITED         ;       SOME OF THE KEYSTROKES WERE LOST
399 400	= 0000 C	; BECAUSE AN AID WAS SENT PO MESSAGE LOST FOR 13 : 3078/3270 KEYBOARD IS NOW INHIBITED -
401	- 0000 C	SOME OF THE KEYSTROKES WERE LOST
402 403	ç	
404	С С	
405	c	。
406 407	C C	;* ;* CODE (IN AH REGISTER) WHEN INTERRUPT HANDLER IS
408		; * ENTERED FROM NDSP BIOS
409 410	C C	· "你我家面写我家家家家就把客桌里把我有这些家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家
411	c	
412 413	= 0001 C = 0002 C	REASON_MESSAGE_RECEIVED EQU 1 ; MESSAGE RECEIVED FROM 3274 REASON POWER ON RESET EQU 2 ; POWER ON RESET RECEIVED FROM 3278/32:
414	= 0003 C	REASON_POURE ON RESET EQU 2 ; POWER ON RESET RECEIVED FROM 3278/32: REASON_DATA_OVERRUN EQU 3 ; DATA RECEIVED FROM 3274 WHEN NO BUFFE
415 416	с с	; NAS AVAILABLE
417	c	
418 419		SUBITL Process an interrupt from the 3278/3279 attachment optic
420		PAGE
421 422		· 被我被把你提出来见你来来这些是我们有是我说在我又帮你不不不不不不不不不不不不不不不不不不不不不不不不不不不不不不不不不不不不
423		;# · · · · · · · · · · · · · · · · · · ·
424		GROUP ALL BIOS DATA SEGMENTS TOGETHER
425 426		1 在我在来来是分型客架的家族家家来有个个人在我家家来在我来看着我来在这家家就在不不不会有不不能用的发展上的家长来有多个是我来来在我来看着那个 <sup>1</sup> 当
427		
428 429		BIOSDATA GROUP BIOSDATA_COMMON,BIOSDATA_KEYBCARD,BIOSDATA_FBA
430	0000	BIOSDATA_COMMON SEGMENT COMMON 'NDSPBIOS'
431	0000	BIOSDATA_COMMON ENDS
432 433	0000	BIOSDATA_KEYBOARD SEGMENT PUBLIC 'NDSPBIOS'
434	0000	BIOSDATA_KEYBOARD EHDS
435 436	0000	BIOSDATA_FBA SEGMENT PUBLIC 'NOSPBIOS'
437	0000	BIOSDATA_FBA ENDS
438 439		Page
440		-
441	0000	NDSPBIOS SEGMENT PUBLIC 'NDSPBIOS'

4,641,262

## 34

	35	4,6	541,262	36
		EXTRN EXTRN EXTRN EXTRN PUBLIC PUBLIC PUBLIC	PROC_INTERRUPTS	LIL:NEAR Avail:near Ear
0000	49 42 4D 20 50 65 72 73 6F 6E 61 6C 20 43 6F 6D 70 75 74 65 72 20 33 32	NDSP_COPYRIGHT DB	LABEL BYTE 'IBM Personal C	omputer 3278 Attachment',ODH,CAH
0027	74         55         72         20         53         52           57         38         20         61         74         74           61         63         68         60         65         6E           74         00         0A         73         69         6F           56         65         72         73         69         6F           62         20         31         2E         30         30           20         28         43         29         43         6F           70         79         72         69         67         68           74         20         49         42         4D         20           43         6F         72         70         20         31           39         38         32         20         0.0         24	DB	'Version 1.00 (	C)Copyright IBM Corp 1982',0DH,0AH,'\$'
	37 36 32 00 0A 24		1.从水水水水水水水水水水水水水水	**************************************
		;* ;* Procedure h ;*	lame: NDSP_INT	
		I# Function: 1	To process all ha Attachment Option	rdware interrupts from the 3278/3279
			3/3279 Attachment	Option status register
		;# ;* Output: Nor ;*	YE	
			(%¥¥%¥¥¥++¥++¥¥¥¥	********
051		NDSP_INT PROC F	AR	
0051 0052 0053 0054 0055 0056 0057 0058 0059	50 53 52 56 57 1E	ASSUME STI PUSH PUSH PUSH PUSH PUSH PUSH PUSH	CS:NDSPBIOS.DS: AX BX CX DX SI DI DI DS ES	BIOSDATA ; ALLOX HIGHER PRIORITY INTERPUPTS ; SAVE REGISTERS
	118 R DE D8	HOV HDV	AX,BIOSDATA DS,AX	; ESTABLISH ADDRESSABILITY TO WORK A
005F	28 006B R	CALL	PROC_INTERRUPTS	; FROCESS ALL PENDING INTERRUPTS
0062 0063 0064 0065 0066 0067 0068 0069	5E 5A 59 5B	FOP POP POP POP POP POP POP	ES DS SI DX CX BX AX	; RESTORE REGISTERS
006A		IRET		; RETURN TO INTERRUPTED PROGRAM
006B	_	NDSP_INT ENDP		
		;* ¦* Procedure N ;*	ame: PPOC_INTEPP	
			ttachment option	nding interrupts from the 1975-1979
		;⊭ Input: 3278 ;#	•	Option status register
		;* Dutput: All :*		essed - EDI sent to 8259 for level 2
				~~~~ <del>~~</del>
	BA DJE4	FROC_INTERRUPTS DETERHINE_CAUSE MOV IN	DX.STATUS_PORT	; read the adapter status register
006E 006F	EC A8 04	TEST	AL,DX AL,ST_START_OP	; WAS THE INTERRUPT CAUSED BY RECEIPT A START OFERATION COMMAND FROM THE

.

-

## 4.641.262

				туч	
			37		38
535	0071	74 05		JZ	CAUSE_1 ; NO, NOT START OPERATION COMMAND
536	0073	E8 0000 E		CALL	PROC_START_OP ; YES, PROCESS START OPERATION COMMAND
537	0076	EB F3		JHP	DETERMINE_CAUSE ; PROCESS NEXT PENDING INTERRUPT
538	0078		CAUSE_1	:	
539	0078	A6 08	.=.	TEST	AL, ST_FC_KB_OUT ; WAS THE INTERRUPT CAUSED BY ACCEPTANC
540			;		OF DATA BY THE PERSONAL COMPUTER?
541	007A	74 05	· · · · · · · · · · · · · · · · · · ·	JZ	CAUSE_2 ; NO, NOT PC PORT AVAILABLE
542	0070	E8 0000 E		CALL	FROE PE FORT AVAIL ; YES, CAN SEND NEXT BYTE TO PE
543	007F	EB EA		JHP	DETERMINE_CAUSE ; FROCESS NEXT PENDING INTERRUPT
544	0081		CAUSE_2	:	-
545	0061	A8 02		TEST	AL, ST_NB_DUT ; WAS THE INTERRUPT CAUSED BY ACCEPTANC
546			;		OF DATA BY THE 3278/3279?
547	0083	74 05		JZ	CAUSE_3 ; NO, NOT 3276/3279 PORT AVAILABLE
342	0085	E8 0000 E		CALL	FROC_3278_PORT_AVAIL ; YES, CAN SEND NEXT BYTE TO
ېېز	2055	EB E1		JMP	DETERMINE_CAUSE ; PROCESS NEXT PENDING INTERRUPT
350			;		3278/3279 TERMINAL
551	CC3A		CAUSE_3	i :	
552	<b>ASCO</b>	A8 01	-	TEST	AL, ST_KB_IN ; WAS THE INTERRUPT CAUSED BY DATA
553			;		AVAILABLE FROM THE 3278/3279 KEYBOARD
554		74 05		JZ	CAUSE_4 ; NO, NOT KEYBOARD DATA AVAILABLE
555	DOCE	E8 0000 E		CALL	FROC_MEYEDARD_DATA ; YES, PROCESS KEYEDARD DATA
55£	0091	EB DS		JITP	DETERMINE_CAUSE ; FROCESS NEXT PENDING INTERRUPT
557					
555	0093		CAUSE_4	:	
55%	0093	FA		CLI	; DISABLE ALL INTERRUPTS
566		B0 62		HOV	AL,EOI ; SIGNAL END OF INTERRUPT FOR LEVEL 2
561	0096	E6 20		OUT	CTL_0259,AL
560					
563	093	C3		RET	; RETURN TO CALLER
5: +					
503	0099		PROC_II	ITERRUPTS	5 ENDP
56 ŝ					
567	0099		NDSPBIC	IS ENDS	
563					
5ć 9				EHD	

APPENDIX B

The IBM Personal Computer MACRO Assembler 3278/3279 Attachment Keyboard Interrupt Handler

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Specifications \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Module Name: NDSPBKBD ı Descriptive Name: 3278/3279 Attachment Keyboard Interrupt Handler ; Copyright: IBM Corp 1982 ; Status: Version 1.00 program ; Notes: Dependencies: None Restrictions: None Module Type: Program Processor: Assembler ; Entry Point: PROC\_KEYBOARD\_DATA Purpose: To process a keystroke from the 3278/3279 keyboard Linkage: CALL from NDSPBIDS Input Parameters: None Entry Point: PROC\_PC\_PORT\_AVAIL Purpose: To send the next waiting keystroke to the Personal Computer Linkage: CALL from NDSPBIOS Input Parameters: None Entry Point: PROC\_3278\_PORT\_AVAIL Purpose: To send the next waiting keystroke to the 3278/3279 Linkage: CALL from NDSPBIDS Input Parameters: None ; Entry Point: SEND\_ENCODED\_KEYSTROKES\_T0\_3278

4,641,262

101

103

104 105 106

107 108

109 110

111 112 113

114

116 117

118

123

124 125

126

136 137 138

139 140

141

143

144

145

146

```
40
```

Purpose: To send a series of encoded keystrokes to the 3278/3279 Linkage: CALL from NDSPBFBA Input Parameters: BX:DX = address of encoded keystrokes CX = number of encoded keystrokes Outputs: CF = 0 if all keystrokes are valid = 1 if any keystroke is invalid AL = return code Entry Point: MOD\_XLAT\_IN Purpose: To modify the inbound keyboard translation table Linkage: CALL from NDSPBFBA Input Farabeiers: CH = 3078/3079 relative key number CL = 3278/3279 keyboard case DH = Fersonal Computer keyboard scan code DL = Personal Computer keyboard case Outputs: CF = 0 if parameters are valid = 1 if parameters are invalid AL = return code BX on stack = old table entry ; Entry Point: HOD\_XLAT\_OUT Purpose: To modify the outbound keyboard translation table Linkage: CALL from NDSPBFBA Input Parameters: CH = encoded keystroke value DH = 3278/3279 relative key number DL = 3278/3279 keyboard case Outputs: CF = 0 if parameters are valid = 1 if parameters are invalid AL = return code BX on stack = old table entry ; Entry Point: QUERY\_KEYBOARD\_ID Purpose: To find out the type of keyboard which is attached Linkage: CALL from NDSPBFBA Input Parameters: None Outputs: BX on stack = keyboard identifier : Exit Normal: Purpose: To continue processing Linkage: Return to caller Output: None unless specified under Entry Point . ; Exit Error: None ; External References: Procedure Invoked: PROC INTERRUPTS Purpose: To process all other pending hardware interrupts Linkage: CALL Input Parameters: None Control Blocks: NDSPDCLS - 3278/3279 Attachment option interfaces NDSPUELS - 3276/3279 BIOS extension interfaces NDSPBCOM - 3278/3279 BIOS common data area ; : .

		41	4,641,262 <b>42</b>
47		71	; Nessages Issued: None
48 49			; ; Change Activity: None
50 51			
52			; *********************************
53 54			SUBTTL WORK AREAS
55 56 <b>5</b> 7			INCLUDE NOSTOCLS.ASH INCLUDE NOSPUSER.ASH .LIST
58 59	0000		·
60	5000		ROMDATA SEGMENT AT 40H
61 62			3 提 3 按书牌新学校内发生型教会教育行为我们的个人的处理不同的不是的学校的家庭的最高级的家族的教育和学校和学校的学校的生活和分子下的产品和专业和自己和
63 64			* WORK AREAS USED BY THE ROM BIOS ROUTINES
65			· 繁洁温餐水来放水放放的水面有不少小个不有多的大的不能有不能不能不能不能有这些不是这些没是来来不能不是有多有些都有的的的不是有不能有多。
66 67	0017		ORG 17H
58 59	D017 ?? = 0D40		R_KB_FLAG DB ? ; FIRST BYTE OF KEYBOARD STATUS
70	= 0020		R_CAPS_STATE EQU 40H ; PC CAPS LOCK HAS BEEN TOGGLED R_NUM_STATE EQU 20H ; PC NUM LOCK HAS BEEN TOGGLED
71 72	= 0010 = 0008		R_SCROLL_STATE EQU 10H ; PC SCROLL LOCK HAS BEEN TOGGLED
73	= 0004		R_ALT_SHIFT EQU 08H ; PC ALTERNATE SHIFT KEY IS DEFRESSED R_CTL_SHIFT EQU 04H ; PC CONTROL SHIFT KEY IS DEFRESSED
74 75	= 0002 = 0001		R_LEFT_SHIFT EQU 02H ; PC LEFT SHIFT KEY IS DEPRESSED
76			R_RIGHT_SHIFT EQU 01H ; PC RIGHT SHIFT KEY IS DEPRESSED
77 78	0018 ?? = 0040		R_KB_FLAG_1 DB ? ; SECOND BYTE OF KEYBOARD STATUS R_CAPS_SHIFT EQU 40H ; PC CAPS LOCK KEY IS DEPRESSED
79	= 0020		R_NUM_SHIFT EQU 20H ; PC NUM LOCK KEY IS DEPRESSED
30 31	= 0010		R_SCROLL_SHIFT EQU 10H ; PC SCROLL LOCK KEY IS DEPRESSED
32	0049		DRG 49H
33 34	0049 ?? 0065		CRT_MODE DB ? ; CURRENT CRT MODE DRG 65H ;
35 36	0065 ??		CRT_MODE_SET DB ? ; CURRENT SETTING OF THE 3X6 REGISTER
57	0066		ROMDATA ENDS
38 39			
0			
71 72			BIDSDATA GROUP BIOSDATA_COMMON,BIOSDATA_KEYEOARD
73			C INCLUDE NDSPBCOM.ASM
94 95	0000		C BIOSDATA_COMMON SEGMENT COMMON 'NDSPBIOS' C
96		1	C _ ;***********************************
97 98			C ;* C ;* NDSPBIOS COMMON WORK AREAS
29 10			C ;* C ;********************************
1			C
12	0000 00 = 0000		C TERH_MODE DB 0 ; CURRENT MODE OF TERHINAL C T_PC EQU 00H ; PERSONAL COMFUTER MODE
4	= 0080	1	C T_HOST EQU BOH ; HOST PROCESSOR MODE
15 16	0001		C BIOSDATA COMMON ENDS
17			C BIOSDATA_COMMON ENDS C
18			
0	0000		BIDSDATA_KEYEDAPD SEGMENT PUBLIC 'NDSPBIDS'
1			"重水埃武武客家这是不是这些主个了????你就不少不紧张子还不能不能不能不能不能不能不能不能不能不能不能不是不多不成,你?要来多这么
3			3 M
5			I* HDSFEIOS KEYBOARD WORK AREAS
6 7			; #*###################################
6	= 681A		DELAY_100_MS EQU 26650 ; LODP CONSTANT NEEDED TO DELAY
9 0			FROCESSING FOR 100 MILLISECDIADS TO FACE KEYSTROKES SENT TO THE
1			3278/3279 DISPLAY UNIT AT & RATE OF
2 3			; 10 KEYSTROKES PER SECOND
4 5	0000 00		KBD_STATE DB 0 ; CURRENT 3278/3279 KEYBOARD STATE
6	= 0080		SHIFT_LOCK EQU 80H ; SHIFT LOCK KEY IS DEPRESSED
7 8	= 0040 = 0020		
9	= 0020		RIGHT_SHIFT EQU 20H ; RIGHT SHIFT KEY IS DEPRESSED NUM_SHIFT EQU 20H ; NUMERIC SHIFT KEY IS DEPRESSED
0 1	= 0010 = 0008		ALPHA_SHIFT EQU 10H ; ALPHA SHIFT KEY IS DEFRESSED
2	= 0004		LOCK_STATE EQU 08H · ; SHIFT LDCK STATE CLICK_STATE EQU 04H · ; CLICKER IS ON ·
3 4	= 0002 = 0001		CLICK_STATE EQU 04H ; CLICKER IS ON ALT_STATE EQU 02H ; ALT KEY IS DEPRESSED FORCED_ALT_STATE EQU 01H ; SEQUENCE OF ALT KEYSTFOKES
5			
	0001 00		PC_STATE DB 0 ; PERSONAL COMFUTER KEYEOARD STATE
6 7	=		P SHIFT , FOULU : SHIFT KEY IS DEPERSIED
5			P_SHIFT EQUU ; SHIFT KEY IS DEFRESSED P_CTL EQUC ; CTRL KEY IS DEFRESSED P_AAT EQUA ; ALT KEY IS DEFRESSED

44 43 240 ; BIOS LOGICAL STATE 241 0002 00 LOGICAL\_STATE DB 0 NEXT KEY IS IN ALT MODE NEXT KEY IS IN CTRL MODE EQU BOH 1 242 243 L\_ALT L\_CTL = 0080 = 0040 EQU 40H ı BIOS IS IN FAUSE STATE PC OUTPUT PORT BECAME AVAILABLE 244 EQU 20H = 0020 PAUSE L\_PC\_PORT\_AVAIL EQU 10H = 0010 ; WHILE IN FAUSE STATE KEYBOARD LOCICALLY DISABLED 246 ş L\_K6D\_DISABLED EQU 08H 247 248 = 0008 ; NUMBER OF DECIMAL DIGITS EXPECTED DEC\_DIGIT DB 0 249 0003 00 250 SCAN CODE FRON 3278/3279 KEYBOARD 251 KBD\_SCAN\_CODE DB 0 0004 00 252 ; SCAN CODE TO PERSONAL COMPUTER 253 0005 00 PC\_SCAN\_CODE DB 0 ; BIT IN SCAN CODE INDICATING BREAK EQU 80H 254 255 = 0080 BREAK\_BIT ; PC SCAN CODE FOR CTRL KEY PC\_CTL\_KEY EQU 29 PC\_SHIFT\_KEY EQU 42 PC\_ALT\_KEY EQU 56 EQU 29 256 = 001D ; PC SCAN CODE FOR SHIFT KEY ; PC SCAN CODE FOR ALT KEY 257 = D02A 258 = 0.038 259 ; 3278/3279 SCAN CODE FOR LOCK KEY ; 3278/3279 SCAN CODE FOR LEFT SHIFT KE ; 3278/3279 SCAN CODE FOR RIGHT SHIFT EQU 4CH = 004C NDS\_LOCK\_KEY 260 NDS\_L\_SHIFT\_KEY EQU 4DH 261 = 004D 262 = 004E 1 3278/3279 SCAN CODE FOR ALT KEY = 0045 NDS\_ALT\_KEY EQU 4FH 263 264 ; PC SCAN CODES FOR DECIMAL DIGITS ON PC\_DIGITS LABEL BYTE 0006 265 THE NUMERIC KEYPAD 266 1 79,80,81,75,76,77,71,72,73,82 0006 4F 50 51 4B 4C 4D DB 267 47 48 49 52 268 269 ; SCAN CODES WAITING TO BE SENT TO THE BUFFER\_3278 DB 16 DUP(?) 0010 10 E 270 271 ?? 3 272 273 3278/3279 274 275 ; BUFFER\_3278\_END LABEL BYTE ; END OF BUFFER BUFFER\_3278\_HEAD DH BIDSDATA:BUFFER\_3278 ; ADDRESS OF FIRST SCAN CODE ; TO BE SENT ; TO BE SENT 0020 0020 0010 R 276 277 EUFFEP\_3278\_TAIL DH BIOSDATA: EUFFEP\_3276 ; ADDRESS OF NEXT FREE SLOT ; IN DUFFEP 278 0022 0010 R 279 ADDRESS OF LAST SCAN CODE TO BE SENT BUFFER\_3278\_LAST DH D 0024 0000 280 281 ; SCAN ECDES WAITING TO BE SENT TO THE EUFFER\_PC DB 16 DUP(?) 10 T 0026 282 283 ?? 1 284 285 FERSONAL COMPUTER 286 BUFFER PC\_END LABEL BYTE ; END OF BUFFER BUFFER\_PC\_HEAD DW BIOSDATA:BUFFER\_PC ; ADDRESS OF FIRST SCAN CODE 287 0036 28B 0036 0026 R TO BE SENT 289 BUFFER\_PC\_TAIL DH BIOSDATA:BUFFER\_PC ; ADDRESS OF NEXT FREE SLOT 290 0038 0026 R IN BUFFER 291 ADDRESS OF LAST SCAN CODE TO BE SENT BUFFER\_PC\_LAST DW 0 0034 D000 292 293 294 295 \* 张客韵把张慧娟在这些我有这些还要在这里却在这些中了这些个个孩子?—— 几十分小过来,了中午不过来吃不过就吃着我不能把不能把你把她把我做我和 296 297 ŧ¥ 298 ;\* KEYBOARD TRANSLATION TABLES 299 ;× 300 301 3278/3279 SCAN CODE TO KEY HUMBER TRANSLATION TABLE ; 302 303 SCAN\_CODE\_TABLE LABEL BYTE 304 0030 0, 0, 0, 0, 0, 0, 0, 0,42,44, 0, 0,68,69,70,41 ; 00-OF 305 003C 00 00 00 00 00 00 DB 306 00 00 2A 2C 00 00 44 45 46 29 307 75,13,40,71,54,27,72, 0,57, 0,73,26, 0,88, 0,89 ; 10-1F 4B 0D 28 47 36 1B DB 308 004C 48 00 39 00 49 1A 00 58 00 59 309 310 08 02 03 04 05 06 07 08 09 0A 00 00 11, 2, 3, 4, 5, 6, 7, 8, 9,10, 0, 0, 0, 0, 0, 0 ; 20-2F 005C DB 311 312 00 00 00 00 313 12,14,53,52,56,28,15, 0, 0, 0, 0, 0, 0, 1, 0, 0 ; 30-3F OC DE 35 34 38 10 DB 314 006C 0 F 00 00 00 00 315 00 00 01 00 00 40 40 46 4F 50 51 316 76,77,78,79,80,81,82,83,84,85,86,87,29,43,55,74 ; 40-4F DB 007C 317 318 52 53 54 55 56 57 1D 2B 37 4A 319 58,59,60,61,62,63,64,65, 0, 0, 0, 0, 0, 0, 0,67,66 ; 50-5F 008C 3A 3B 3C 3D 3E 3F DB 320 321 40 41 00 00 00 00 00 00 43 42 322 30,49,47,32,18,33,34,35,23,36,37,38,51,50,24,25 ; 60-6F 1E 31 2F 20 12 21 22 23 17 24 25 26 DB 0090 323 324 33 32 16 19 10 13 1F 14 16 30 325 16,19,31,20,22,48,17,46,21,45, 0, 0, 0, 0,39, 0 ; 70-7F 326 DACO DB 11 2E 15 20 00 00 327 00 00 27 00 328 329 KEY NUMBER TO 3278/3279 SCAN CODE TRANSLATION TABLE : 330 331 SCAN CODE XLAT LABEL BYTE DOBC 332

4,641,262

### 41 0/0

4	,041,202
DB	3DH,21H,22H,
DB	20H,30H,11H,
OB	78H,74H,68H.6

						4,641,262
			45	5		46
333	0080	3D 21 22		25		DB 3DH,21H,22H,23H,24H,25H,26H,27H,28H,29H ; 1-10
334 335	0006	26 27 28 20 30 11		70		DB 20H,30H,11H,31H,36H,70H,76H,64H,71H,73H ; 11-20
336 337	0000	76 64 71 78 74 68		IR		DB 78H,74H,68H.6EH,6FH.1BH,15H,35H,4CH,60H ; 21-30
338		15 35 4C	60			
339 340	00DA	72 63 65 6A 6B 7E		69		DB 72H,63H,65H.66H,67H,69H,6AH,6BH,7EH,12H ; 31-40
341 342	00E4	0F 08 4D 62 75 61		77		DB 0FH,08H,4DH,09H,79H,77H,62H,75H,61H,6DH ; 41-50
343	ODEE	6C 33 32	14 4E	34		DB 6CH, 33H, 32H, 14H, 4EH, 34H, 18H, 50H, 51H, 52H ; 51-60
344 345	ODF8	18 50 51 53 54 55		5F		DB 53H,54H,55H,56H,57H,5FH,5EH,0CH,0DH,0EH ; 61-70
346 347	0102	5E OC OD 13 16 1A		40		DB 13H,16H,1AH,4FH,10H,40H,41H,42H,43H,44H ; 71-80
348 349		41 42 43	44			
350	0100	45 46 47 4B 1D 1F	40 47	44		DB 45H,46H,47H,48H,49H,4AK,48H,1DN,1FH ; 81-89
351 352						KEY NUMBER TO PC SCAN CODE TRANSLATION TABLES
353 354						PUBLIC XLAT_TABLES,XLAT_TABLES_END
355 356	0115 0116	90				EVFN XLAT_TABLES LABEL WORD
357	0116	???? 59 [				XLAT_TABLE_ID DH ? ; KEYEDARD IDENTIFIER LOHER_CASE_XLAT DH 89 DUP(?) ; LOHER CASE
358 359	0118	57 1	????	_		COMER_CASE_XERT DW BY DUP(:) ; COMER LASE
360 361				1		
362 363	A 310	59 I	????			UPPER_CASE_XLAT DH 89 DUP(?) ; UPPER CASE
364			••••	1		
365 366	027C	59 1				CTL_CASE_XLAT DH 89 DUP(?) ; PC CTRL CASE
367 368			????	1		
369		4		1		
370 371	032E	59 [	????			ALT_CASE_XLAT DW 89 DUP(?) ; PC ALT CASE
372 373				1		
374	03E0	59 i				ALT_KEY_XLAT DH 89 DUP(?) ; ALT CASE
375 376			????	1		
377 378	0492	0100 E				KEYSTROKE_XLAT DH 256 DUP(?) ; ENCODED KEYSTROKE TRANSLATION TABLE
379	0172	0.00 .	????			RETSTRORE REAL DR 250 DOP(;) ; ERCODED RETOTRORE HURDERTEN THEE
380 381				1		
362 383	0692					XLAT_TABLES_END LABEL BYTE ; END OF TRANSLATION TABLES
384 385	0692					BIOSDATA_KEYBOARD ENDS
386 367						· SUBITL Process an interrupt from the 3278/3279 keyboard
325 359	0000					NDS7DIDS SEGMENT FUELIC 'NOSFBIOS'
390	0000					
391 392						PUBLIC PROC_KEYBOARD_DATA PUBLIC PROC_FC_PORT_AVAIL
393 394						FUBLIC FROC_3276_FORT_AVAIL FUBLIC SEND_ENCODED_KEYSTROKES_TO_3278
395						PUBLIC MOD_XLAT_IN
396 397						PUBLIC HOD_XLAT_OUT PUBLIC QUEPY_KEYEOAFD_ID
398 399		-				EXTRN FFDC_INTERRUPTS:NEAR
400						ASSUME CS:HDSPBIOS,DS:BIOSDATA
401 402						X35672 C3.763-0103,03.6105674
403 404						•
405						。
406 407						;* }* Procedure Name: PRDC_KEYBOARD_DATA *
405 409						<pre>3* 3* Function: To process a keystroke from the 3278/3279 *</pre>
410						;*
411 412						;#         Input: 3278/3279 keyboard scan code         *           ;#
413 414						1* Output: keystroke sent to Personal Computer or 3278/3279 *
415						】 "你我们我我这个年龄大学我们为小小子的这些我去们不能要我这些我就能不能出现我有多少的好多是我不会要不会要要不要要要要要不是我的的是是没有帮助。" 3、
416 417	0000					PRDC_KEYBOARD_DATA PROC NEAR
418 419					с	INCLUDE NDSPKBD1.ASM
420 421					c c	************************
422					С	
423 424					С	3* RELATIVE KEY NUMBER
425					c	;* *

.

#### 

.

; <del>* * * * * *</del>		
	***	**************************************
	HOV IN	DX,NDS_KB_IN ; READ THE 3278/3279 KEYBOARD SCAN CODE AL,DX
	TEST	LOGICAL_STATE,L_KBD_DISABLED ; IS THE KEYBOARD ; LOGICALLY DISABLED?
	JZ XOR	KEYBOARD_0 ; NO, CONTINUE Al,al ; YES, SIMULATE A KEYBOARD OVERFUN
KEYBOAR	MOV	KBD_SCAN_CODE,AL; SAVE 3278/3279 KEYBOARD SCAN CODE
	OR	AL,AL   DID THE KEYBOARD INDICATE OVEREUN?
	JNZ	KEYBOARD_1 ; ND, CONTINUE PROCESSING
	JMP	OVERRUN ; YES, PROCESS OVERRUN
KEYBOAR	D_1:	
	MOV	AH,AL ; SAVE THE MAKE/BREAK BIT IN AH
	AND	AH,80H
	AND	AL,7FH ; ISOLATE SCAN CODE
	LEA	BX,SCAN_CODE_TABLE ; CONVERT SCAN CODE TO
	XLAT OR	SCAN_CODE_TABLE ; RELATIVE KEY NUMBER AH,AL
	ψĸ	
3		5 POINT:
;		= RELATIVE KEY NUMBER
5		= RELATIVE KEY NUMBER WITH MAKE/BREAK BIT SCAN_CODE = SCAN CODE WITH MAKE/BREAK BIT
;	KDD	CAR_CODE - SEAR CODE ATTA TAREFOREAR DAT
	*****	* 《形武家教教家》:"我就要要在这家庭家庭的家族的家庭和这些家庭家庭家庭家庭家家家家家家家家家家家家家家家家家家家家家家家家家家家
3* ;*	TRANSLA	TE THE RELATIVE KEY NUMBER TO A 3278/3279 SCAN CODE
;+ ;+		. *
;*	THE TRA	ANSLATION TABLE USED MAY BE ONE OF THE FOLLOWING: *
;*		CASE_XLAT - WHEN THE NEXT KEYSTROKE IS IN PC ALT CASE *
;# ;#		CASE_XLAT - WHEN THE NEXT KEYSTROKE IS IN PC CTL CASE * Key_XLAT - WHEN THE 3278/3279 ALT KEY IS DEFRESSED *
;*	LOHE	R_CASE_XLAT - WHEN THE 3278/3279 KEYBOARD IS IN LOWER *
j#		OR ALFHA CASE SHIFT *
3*	UPPE	R_CASE_XLAT - WHEN THE 3278/3279 KEYBOARD IS IN UPPER *
;*		OR NUMERIC CASE SHIFT *
;+ ;+	THE TRA	ANSLATED SCAN CODE MAY BE:
3*	0	= IGNORE THIS KEYSTROKE *
:*		B3 = PERSONAL COMPUTER SCAN CODE
1*	-1 1	TO -127 = INDEX INTO SPECIAL FUNCTION TABLE *
; <del>.</del> ;		在安全最高级无意思不能要求这些的考虑不能就不能就在这些不能要要不能有这些不能帮助我就要承受不能不能不能不能不是不是
,		
	1.84	CHIALT_CASE_XLAT ; FOINT TO PROPER TRANSLATION
	TEST	LOGICAL_STATE,L_ALT ; TABLE
	JNZ LEA	XLAT_SCAN_CODE - BX;CTL_CASE_XLAT
	TEST	
		LOGICAL STATE, L_CTL
	JHZ	LGGICAL_STATE,L_CTL XLAT_SCAN_CODE
	LEA	XLAT_SCĀN_CODE DX.ALT_KEY_XLAT
	LEA TEST	XLAT_SCAN_CODE DX.ALT_KEY_XLAT FED_STATE.ALT_STATE
	LEA	XLAT_SCĀN_CODE DX.ALT_KEY_XLAT
	LEA TEST JNZ LEA TEST	XLAT_SCAN_CODE EX.ALT_KEY_XLAT FCD_STATEXALT_STATE XLAT_SCEALCODE EX.LOUER_CASE_XLAT KED_STATE.ALPHA_SHIFT
	LEA TEST JHZ LEA TEST JNZ	XLAT_SCAN_CODE LX.ALT_KEY_XLAT FD_STATE.ALT_STATE XLAT_SCAN_CODE EX.LOWER_CASE_XLAT KED_STATE.ALPHA_SHIFT XLAT_SCAN_CODE
UP_SHIF	LEA TEST JHZ LEA TEST JNZ TEQU	XLAT_SCAN_CODE EX,ALT_KEY_XLAT FED_STATE,ALT_STATE XLAT_SCAN_CODE EX,LOHER_CASE_XLAT KED_STATE,ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT
UP_SHIF	LEA TEST JHZ LEA TEST JNZ	XLAT_SCAN_CODE LX.ALT_KEY_XLAT FD_STATE.ALT_STATE XLAT_SCAN_CODE EX.LOWER_CASE_XLAT KED_STATE.ALPHA_SHIFT XLAT_SCAN_CODE
UP_SHIF	LEA TEST JHZ LEA TEST JNZ TEQU TEST	XLAT_SCAN_CODE EX.ALT_KEY_XLAT FED_STATE_ALT_STATE XLAT_SCAN_CODE EX.LOHER_CASE_XLAT KED_STATE.ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.UP_SHIFT
_	LEA TEST JHZ LEA TEST JNZ TEQU TEST JZ LEA	XLAT_SCAN_CODE EX,ALT_KET_XLAT FED_STATE,ALT_STATE XLAT_SCAN_CODE BX,LOHER_CASE_XLAT KED_STATE,ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE,OP_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT
_	LEA TEST JHZ LEA TEST JNZ TEST JZ LEA CAN_CODE :	XLAT_CCAN_CODE LX,ALT_KEY_XLAT FTD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KED_STATE.ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.OP_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT
_	LEA TEST JHZ LEA TEST JNZ TEQU TEST JZ LEA	XLAT_SCAN_CODE LX,ALT_KET_XLAT FD_STATE,ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KED_STATE,ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE,UP_SHIFT XLAT_SCAN_CODE EX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL,AL
_	LEA TEST JHZ LEA TEST JNZ FT EQU TEST JZ LEA CAN_CODE : DEC MOV XOR	XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KED_STATE.ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.OP_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH
_	LEA TEST JHZ LEA TEST JNZ TEQU TEST JZ LEA CAN_CODE : DEC MDV XOR SAL	XLAT_SCAN_CODE LX.ALT_KEY_XLAT FTO_STATE.ALT_STATE XLAT_SCAN_CODE BX.LOWER_CASE_XLAT KED_STATE.GODE LCCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE BX.UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.I
_	LEA TEST JHZ LEA TEST JNZ TEST JZ LEA LEA CAN_CODE: DEC MOV XOR SAL MOV	XLAT_SCAN_CODE LX,ALT_KEY_XLAT FLD_STATE.ALT_STATE XLAT_SCAN_CODE BX,LOHER_CASE_XLAT KED_STATE.ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.UP_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.AL CH.AL CX.1 SI,CX
_	LEA TEST JHZ LEA TEST JJZ TEST JZ LEA CAN_CODE: DEC MOV XOR SAL HOV HOV	XLAT_SCAN_CODE LX,ALT_KET_XLAT FTD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KED_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI,CX DX,[BX+SI] ; DH = SCAN CODE ; DL = REQUIRED PC SHIFT STATE
_	LEA TEET JHZ LEA TEET JJZ TEEU JZ LEA CAN_CODE: DEC MOV XOR SAL HOV MOV	XLAT_SCAN_CODE LX,ALT_KEY_XLAT FD_STATE.ALT_STATE XLAT_SCAN_CODE BX,LOHER_CASE_XLAT KED_STATE.ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.UP_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CL.AL CL.AL CX.1 SI,CX DX,IBX+SI] ; DH = SCAN CODE ; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE,DH ; SAVE PERSONAL COMPUTER SCAN CODE
_	LEA TEST JNZ LEA TEST JNZ TECU TEST JZ LEA CAN_CODE: DEC MOV XOR SAL MOV MOV CMP	XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KEC_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI,CX DX.1EX+SI] ; DH = SCAN CODE j DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE.OH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO
_	LEA TEET JHZ LEA TEET JJZ TEEU JZ LEA CAN_CODE: DEC MOV XOR SAL HOV MOV	XLAT_SCAN_CODE LX,ALT_KEY_XLAT FD_STATE.ALT_STATE XLAT_SCAN_CODE BX,LOHER_CASE_XLAT KED_STATE.ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.UP_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CL.AL CL.AL CX.1 SI,CX DX,IBX+SI] ; DH = SCAN CODE ; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE,DH ; SAVE PERSONAL COMPUTER SCAN CODE
_	LEA TEST JHZ LEA TEST JNZ TECU TEST JZ LEA CAN_CODE: DEC MOV XOR SAL HOV HOV CHP JNE	XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KEC_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE EX.UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI,CX DX.1EX+SI] ; DH = SCAN CODE ; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE.DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO, CONTINUE TERM_MODE.T_HOST ; IS THE TERMINAL IN HOST MODE? DONT_IGNORE ; YES, PROCESS THE KEYSTROKE
_	LEA TEEST JHZ LEA TEEST JZ LEA CAN_CODE: LEA CAN_CODE: DEC MOV XOR MOV MOV MOV HOV CHP JNE TEST	XLAT_CCAN_CODE LX,ALT_KEY_XLAT FTO_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KED_STATE.ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX,1 SI.CX DX,IBX+SI] ; DH = SCAN CODE ; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE,DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO, CONTINUE TERM_MODE,T_HOST ; IS THE TERMINAL IN HOST MODE? DOWN_IGNORE ; YES, PROCESS THE KEYSTROKE RESET_SHIFTS ; NO, IGNORE KEYSTROKE - RESET LOGICAL
XLAT_SC	LEA TEEST JHZ LEA TEET JZ LEA TEET JZ LEA CAN_CODE: DEC MOV XOR XOR XOR XOR XOR XOR XOR XOR XOR XOR	XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KEC_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE EX.UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI,CX DX.1EX+SI] ; DH = SCAN CODE ; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE.DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO, CONTINUE TERM_MODE.T_HOST ; IS THE TERMINAL IN HOST MODE? DONT_IGNORE ; YES, PROCESS THE KEYSTROKE
_	LEA TEEST JHZ LEA TEET JZ LEA TEET JZ LEA CAN_CODE: DEC MOV XOR XOR XOR XOR XOR XOR XOR XOR XOR XOR	XLAT_CCAN_CODE LX,ALT_KEY_XLAT FTO_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KED_STATE.ALPHA_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX,1 SI.CX DX,IBX+SI] ; DH = SCAN CODE ; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE,DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO, CONTINUE TERM_MODE,T_HOST ; IS THE TERMINAL IN HOST MODE? DOWN_IGNORE ; YES, PROCESS THE KEYSTROKE RESET_SHIFTS ; NO, IGNORE KEYSTROKE - RESET LOGICAL
XLAT_SC	LEA TEET JHZ LEA TET TEGU TEST JZ LEA LEA CAN_CODE: JC CODE: MOV XOR HOV HOV CMP JNE TEST JNZ JNP SHORE: JMP	<pre>XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KED_STATE.ALTPAL_SHIFT XLAT_SCAN_CODE LOCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE EX.UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI.CX DX,IEX+SI] ; DH = SCAN CODE j DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE.DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO, CONTINUE TERM_HODE.T_HOST ; IS THE TERMINAL IN HOST MODE? DONT_IGNORE ; YES, PROCESS THE KEYSTROKE RESET_SHIFTS ; NO, IGNORE KEYSTROKE - RESET LOGICAL ; SHIFT STATES</pre>
XLAT_SC	LEA TEST JHZ LEA TEST JZ LEA CAN_CODE: DEC MOV XDR SAL MOV MOV CHP JNE TEST JNZ JMP	<pre>XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KEC_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE LCCK_STATE.OP_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE EX.UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI.CX DX.IEX+SI] ; DH = SCAN CODE i ; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE.DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO, CONTINUE TERM_MODE.T_HOST ; IS THE TERMINAL IN HOST MODE? DONT_IGNORE ; YES, PROCESS THE KEYSTROKE RESET_SHIFTS ; NO, IGNORE KEYSTROKE TO 3274/3276</pre>
XLAT_SC	LEA TEST JNZ LEA TEST JSZ LEA CAN_CODE: DEC MOV XOR SAL MOV MOV CHP JNE TEST JNZ JHP SNORE: JMP	<pre>XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASS_XLAT KED_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.UP_SHIFT XLAT_SCAN_CODE BX.UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI.CX DX.IBX+SI] ; DH = SCAN CODE ; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE.DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.O ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO. CONTINUE TERM_MODE.T_HOST ; IS THE TERMINAL IN HOST MODE? DOWT_IGHORE ; YES.PROCESS THE KEYSTROKE RESET_SHIFTS ; NO, IGHORE KEYSTROKE - RESET LOGICAL ; SHIFT STATES ROUTE ; SEND KEYSTROKE TO 3274/3276 XLAT_2 ; NEGATIVE, PROCESS FUNCTION CODE</pre>
XLAT_SC	LEA TEST JHZ LEA TEST JZ LEA LEA LEA LEA CODE:: JZ LEA MOV XOR HOV CMP JNE TEST JNZ JNP SKORE: JMP	<pre>XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KEC_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE LCCK_STATE.OP_SHIFT XLAT_SCAN_CODE EX.UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI.CX DX.IEX+SI] ; DH = SCAN CODE i ; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE.DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO, CONTINUE TERM_MODE.T_HOST ; IS THE TERMINAL IN HOST MODE? DONT_IGNORE ; YES, PROCESS THE KEYSTROKE RESET_SHIFTS ; NO, IGNORE KEYSTROKE TO 3274/3276</pre>
XLAT_SC DONT_IC XLAT_1: XLAT_2:	LEA TEEST JHZ LEA TEST JZ LEA LEA LEA LEA LEA LEA LEA MOV MOV CHP JNE TEST JNZ TEST JNZ TEST JNZ SAL MOV CHP JNE TEST JNZ SAL HOV CHP JNE TEST JNZ LEA LEA LEA LEA TEST JZ LEA LEA LEA TEST JZ LEA LEA LEA TEST JZ LEA LEA LEA LEA TEST JZ LEA LEA LEA LEA LEA TEST JZ LEA LEA LEA LEA LEA LEA LEA LEA LEA LEA	<pre>XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASS_XLAT KED_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.UP_SHIFT XLAT_SCAN_CODE BX.UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI.CX DX.IBX+SI] ; DH = SCAN CODE ; OL = REQUIRED PC SHIFT STATE PC_SCAN_CODE,DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO. CONTINUE TERM_HODE,T_HOST ; IS THE TERHINAL IN HOST MODE? DOWIT_IGNORE ; YES, PROCESS THE KEYSTROKE RESET_SHIFTS ; NO, IGNORE KEYSTROKE - RESET LOGICAL ; SHIFT STATES ROUTE ; SEND KEYSTROKE TO 3274/3276 XLAT_2 ; NEGATIVE, ROUTE SCAN CODE AS REQUIRED</pre>
DONT_IC XLAT_2: XLAT_2: ;*****	LEA TEEST JHZ LEA TEST JZ LEA LEA LEA LEA LEA LEA LEA MOV MOV CHP JNE TEST JNZ TEST JNZ TEST JNZ SAL MOV CHP JNE TEST JNZ SAL HOV CHP JNE TEST JNZ LEA LEA LEA LEA TEST JZ LEA LEA LEA TEST JZ LEA LEA LEA TEST JZ LEA LEA LEA LEA TEST JZ LEA LEA LEA LEA LEA TEST JZ LEA LEA LEA LEA LEA LEA LEA LEA LEA LEA	<pre>XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASS_XLAT KED_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE LCCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT KED_STATE.OP_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI.CX DX.1BX+SI] ; DH = SCAN CODE ; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE,DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO. CONTINUE TERM_MODE,T_HOST ; IS THE TERMINAL IN HOST MODE? DOWIT_IGHORE ; YES, PROCESS THE KEYSTROKE RESET_SHIFTS ; NO, IGNORE KEYSTROKE - RESET LOGICAL ; SHIFT STATES ROUTE ; SEND KEYSTROKE TO 3274/3276 XLAT_2 ; NEGATIVE, PROCESS FUNCTION CODE</pre>
XLAT_SC DONT_IC XLAT_1: XLAT_2:	LEA TEST JHZ LEA TEST JZ LEA TEST JZ LEA CAN_CODE: DEC MOV XOR MOV MOV MOV MOV MOV MOV MOV MOV MOV SAL MOV JNE TEST JNZ JHP JNZ JHP	<pre>XLAT_CCAN_CODE LX,ALT_KEY_XLAT FCD_STATE.ALT_STATE XLAT_SCAN_CODE EX,LOWER_CASE_XLAT KED_STATE.OR_LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE LCCK_STATE OR LEFT_SHIFT OR PIGHT_SHIFT OR NUM_SHIFT XLAT_SCAN_CODE BX,UPPER_CASE_XLAT AL ; PICK UP CODE FROM TRANSLATION TABLE CL.AL CH.CH CX.1 SI.CX DX,IBX+SI] ; DH = SCAN CODE i; DL = REQUIRED PC SHIFT STATE PC_SCAN_CODE.DH ; SAVE PERSONAL COMPUTER SCAN CODE DH.0 ; COMPARE SCAN CODE TO ZERO XLAT_1 ; NOT ZERO, CONTINUE TERM_MODE,T_HOST ; IS THE TERMINAL IN HOST MODE? DONT_IGNORE ; YES, PROCESS THE KEYSTROKE RESET_SHIFTS ; NO, IGNORE KEYSTROKE - RESET LOGICAL ; SHIFT STATES ROUTE ; SEND KEYSTROKE TO 3274/3276 XLAT_2 ; NEGATIVE, PROCESS FUNCTION CODE ROUTE ; POSITIVE, ROUTE SCAN CODE AS REQUIRED</pre>

-

		c c	; ************************************	- 2 4 1 - 2 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0083	F6 DE	C C	LIF C	DH ; FIND PROPER ENTRY IN BRANCH TABLE
0065	F6 06 0002 R 20	c	NEG Test	DH ; FIND PROPER ENTRY IN BRANCH TABLE LCGICAL_STATE,L_PAUSE ; ARE WE IN PAUSE STATE?
006A	74 08	С	JZ	XLAT_3 ; NO, NORMAL PROCESSING
008C -	80 FE 07 76 03	c	CMP	DH,NO_SHIFT_KEYS; YES, IS THIS A SHIFT KEY?
0091	E9 021F R	C C	JBE JMP	XLAT_3 ; YES, PROCESS IT NORMALLY ROUTE ; NO, EXIT PAUSE STATE
0094		C	XLAT_3:	
0094 0096	FE CE 8A DE	C C	DEC	DH
0098	32 FF	C C	MOV XOR	BL,DH BH,BH
009A	D1 E3	č	SAL	BX,1
0090	8B F3	C	NOV	SI,BX
009E 00A3	2E: 8B 9C 00A5 R FF E3	C C	NOV JHP	EX,CS:FUNCTIONS(SI) EX ; PERFORM SPECIAL FUNCTION
		C		
00A5 00A5	00C1 R	C C	FUNCTIONS LABE	L WORD PROC_SHIFT_LOCK ; SHIFT LOCK KEY PRESSED/RELEASED
00A7	ODEF R	č	אמ	PROC_L_SHIFT ; LEFT SHIFT KEY PRESSED/RELEASED
0049	ODF3 R	C	DH	PROC_P_SHIFT ; RIGHT SHIFT KEY PRESSED/RELEASED
DAB	0114 R	C	DW	PROC_HUM_LOCK ; NUM LOCK KEY FRESSED/RELEASED
DAD	0120 R 0125 R	C C	שם אם	FROC_NUM_SHIFT ; NUM SHIFT KEY FRESSED/RELEASED FROC_ALPHA_SHIFT ; ALPHA SHIFT KEY FRESSED/RELEASED
00B1	0134 R	č	DH	FROC_ALT_SHIFT ; ALT KEY FRESSED/RELEASED
0083	0140 R	c	DH	FROC_RESET ; RESET KEY FRESSED
065	0181 R 019f R	C C	DH DH	FROC_CLICK ; CLICK KEY FRESSED FROC DECIMAL KEY ; ENTER DECIMAL DIGIT INPUT MODE
089	OIBF R	č	DH	FROC_SHITCH_MODE; TOGGLE MODE FRESSED
DOBB	0101 R	C	DW	FTCC_CTL_KEY   NEXT KEYSTROKE IS IN PC CTL MODE
OBD	OIED R	C	DW	FLOC_/ET_KEY & NEXT REVISITORE IS IN PC ALT HODE
DOBF	OIEF R	с С	DW	INDELPAUSE ; ENTER PAUSE STATE
		С	I SHIFT L	OCK KEY WAS PRESSED OR RELEASED
0001		C C	PROC_SHIFT LOCK	;
0001	F6 C4 80	с	TEST	AH,BREAK_BIT ; WAS KEY FRESSED?
0004	75 22	с с	JNZ	EREAK_SHIFT_LOCK ; ND, SHIFT LOCK WAS RELEASED KBD_STATE,LOCK_STATE ; INDICATE SHIFT LOCK STATE
DOCB	80 OE 0000 R 08 80 DE 0000 R 80	c	OR OR	KED_STATE, SHIFT_LOCK ; INDICATE SHIFT LOCK KEY DEFRESSED
		С		
000 000	F6 06 0000 R 80	c c	PROC_SHIFT_KEY: TEST	TERM_HODE,T_HOST ; ARE WE IN HOST PROCESSOR MODE?
005	75 06	C	JNZ	TEST_PAUSE   YES, THE SHIFT KEY WILL BE ROUTED
007	10 0004 0	C	MOLT	; TO THE 3278/3279 NORMALLY AL,KBD_SCAN_CODE ; NO, ROUTE THE SHIFT KEY TO THE
ODA	A0 0004 R E8 02FD R	с с	MDV CALL	SEND_TO_3278 ; 3278/3279 DISPLAY TO KEEP THE CONTROL
		С		; UNIT IN SYNCH WITH BIDS
000		с С	TEST_PAUSE:	
000	F6 06 0002 R 20	C	TEST	LOGICAL_STATE, L_PAUSE ; ARE WE IN PAUSE STATE?
E2	75 03	C	JNZ	TEST_PAUSE_1 ; YES, DISCARD THE KEY STROKE Route ; route the key stroke
0E4 0E7	E9 D21F R	C C	JMP TEST_PAUSE_1:	ROUTE ; ROUTE THE KEY STROKE
0E7	C3	С	RET	3 DISCARD THE KEY STROKE
0E8	80 26 0000 5 75	C	BREAK_SHIFT_LOC	K: KBD_STATE,NOT SHIFT_LOCK ; INDICATE SHIFT LOCK KEY IS NO
0E8	80 26 0000 R 7F	C C	AND ;	LONGER DEPRESSED
OED	EB E1	С	JMP	PROC_SHIFT_KEY ; ROUTE KEYSTROKE AS NEEDED
		C C	; LEFT HA	ND SHIFT KEY WAS PRESSED OR RELEASED
		С		
DEF	B0 40	C C	PROC_1_SHIFT:	AL, LEFT_SHIFT ; SET UP MASK FOR LEFT SHIFT KEY
	BO 40 EB 02	C	MOV JMP	SHORT PROC_SHIFT ; PROCESS SHIFT KEY
		С		
		C C	I RIGHT H	AND SHIFT KEY WAS PRESSED OR RELEASED
DF3		c	PROC_R_SHIFT:	
	B0 20	С	- MOV	AL, RIGHT_SHIFT ; SET UP MASK FOR RIGHT SHIFT KEY
		C C	I LEFT OF	RIGHT SHIFT KEY WAS FRESSED OR RELEASED
		C		
0F5 0F5	F6 C4 80	C C	PROC_SHIFT: TEST	AH, BREAK_BIT ; WAS SHIFT KEY PRESSED?
IF8	75 06	c	JNZ	BREAK_SHIFT ; NO, SHIFT KEY WAS RELEASED
FA	08 06 0000 R	С	OR	KED_STATE, AL ; INDICATE SHIFT KEY IS DEPRESSED
	EB DO	C C	JMP BREAK SHIET.	PROC_SHIFT_KEY ; ROUTE THE KEYSTROKE AS NEEDED
100	F6 D0	c c	BREAK_SHIFT: NOT	AL ; INDICATE SHIFT KEY IS NO LONGER
	20 06 0000 R	C	AND	KED_STATE,AL ; DEPRESSED
106	F6 D6 0000 R E0	с с	TEST	KED_STATE,LEFT_SHIFT+RIGHT_SHIFT+SHIFT_LOCK ; IS EITHER
		С	;	SHIFT KEY OR SHIFT LOCK DEPRESSED?
	75 05	c	JNZ	BREAK_SHIFT_1 ; YES, LEAVE SHIFT LOCK STATE AS IS
100	80 26 0000 R F7	с с	dha t	KED_STATE,NOT LOCK_STATE ; THE LAST SHIFT KEY HAS BEEN RELEASED - CANCEL SHIFT LOCK STATE
12	50 D.C	С	BREAK_SHIFT_1:	
	EB BC	ç	JWb	PROC_SHIFT_KEY ; ROUTE THE KEYSTROKE AS NEEDED
		С		
		с С С	; NUMERIC	LOCK KEY WAS PRESSED OR RELEASED

		4,641,262
	51	52
	01	
612	0114	C PROC_NUM_LOCK: C TEST AH,BREAK_BIT   WAS THE KEY PRESSED?
613 614	0114 F6 C4 80 0117 75 05	C TEST AH, BREAK_BIT   HAS THE KEY PRESSED ? C JNZ JMP_ROUTE ; NO, NUM LOCK KEY WAS RELEASED
615	0119 80 36 DODO R 08	C XOR KBD_STATE,LOCK_STATE ; TOGGLE NUMERIC LOCK STATE
616	011E	C JHP_ROUTE: C JHP PROC SHIFT KEY ; ROUTE THE KEYSTROKE AS NEEDED
617	OIIE EB BO	C JMP PROC_SHIFT_KEY ; ROUTE THE KEYSTROKE AS NEEDED
618 619		C ; NUMERIC SHIFT KEY WAS PRESSED OR RELEASED
620		c
621	0120	C PROC_NUM_SHIFT:
622		HOW ALARM SHIFT ; SET UP MASK FOR HOMERIC SHIFT KEY
623		C JHP FOCCLAR_SNIFT & PROCESS SHIFT MEY C
624 625		C 3 ALPHYMERIC SHIFT KEY WAS PRESSED OR RELEADED
626		
627		C PROC_ALPHA_SHIFT:
628	0125 B0 10	C HOV AL,ALPHA_SHIFT ; SET UP MASK FOR ALPHAMERIC SHIFT KEY
629 630		C ; ALPHA OR HUMERIC SHIFT KEY WAS PRESSED OR RELEASED
631		C
632	0127	C PROC_AN_SHIFT:
633	0127 F6 C4 80	C TEST AH, BREAK_BIT ; WAS SHIFT KEY PRESSED? C JNZ EPEAK_AN_SHIFT ; NO, SHIFT KEY WAS RELEASED
634	012A 75 06	
635 636	012C 08 06 0000 R 0130 EB 9E	C OR KCD_STATE,AL ; INDICATE SHIFT KET IS DEFREGUED C JMP FROC_SHIFT_KEY ; ROUTE THE KEYSTROKE AS NEEDED
637	0132	C RDEAK AN SHIET.
638	0132 F6 D0	C NOT AL ; INDICATE SHIFT KEY IS NO LUNGER
639	0134 20 06 0000 R	C AND KBD_STATE,AL ; DEPRESSED C JMP PROC_SHIFT_KEY ; ROUTE THE KEYSTROKE AS NEEDED
640	0138 EB 96	
641 642		C ALT KEY PRESSED OR RELEASED
643		
644	013A	C PROC_ALT_SHIFT:
645	013A F6 C4 80	C TEST AH, BREAK_BIT ; WAS THE KEY PRESSED? C JNZ BREAK_ALT_SHIFT ; NO, ALT KEY WAS RELEASED
646	013D 75 07	C JNZ BREAK_ALT_SHIFT ; NO, ALI KET WAS RELEASED C OR KED_STATE,ALT_STATE ; INDICATE ALT KEY IS DEPRESSED
647 648	013F 80 0E 0000 R 02 0144 EB 8A	C JHP PROC_SHIFT_KEY I ROUTE THE KEYSTROKE AS NEEDED
649	0146	C DREAV ALT SUTET.
650	0146 80 26 0000 R FD	C AND KED_STATE, NOT ALT_STATE ; INDICATE ALT KEY IS NO LUNGER
651		C ; DEPRESSED C JMP PROC SHIFT KEY ; ROUTE THE KEYSTROKE AS NEEDED
652	014B EB 83	C JMP PROC_SHIFT_KEY ; ROUTE THE REFSIRORE AS REEDED
653 654		C ; RESET KEY WAS PRESSED
655		c
656	014D	C PROC_RESET:
657	014D FG 06 0000 R 80	C TEST TERM_MODE, THOST ; ARE WE IN PC MODE? C JNZ PROC_RESET 1 ; ND, DO NOT PROCESS RESET KEY
658	0152 75 2A	T T T T TRANSPORT THRUT MODE
659 660	0154 C6 06 0003 R 00 0159 80 26 0002 R 3F	C HOV DEC_DIGIT,0 ; CANCEL DECIMAL DIGIT INFOT HODE C AND LOGICAL_STATE,NOT L_ALT+L_CTL ; CANCEL PC ALT
661		C 3 AND PC CTL MODES
662		c
663	015E	C RESET_ALT: C NOV AL, BREAK_BIT OR PC_ALT_KEY; INDICATE THE ALT KEY
664 44E	015E B0 B8 0160 E8 0367 R	C HOV AL, BREAK_BIT OR PC_ALT_KEY ; INDICATE THE ALL KEY C CALL SEND_TO_PC ; WAS RELEASED
665 666	0160 EB 0367 R 0163 B0 90	C MOV AL, BREAK_BIT OR PC_CTL_KEY ; INDICATE THE CTL KEY
667	0165 EB 0367 R	C CALL SEND TO PC ; WAS RELEASED
668	0168 80 26 0001 R 1F	C AND PC_STATE, NOT (P_SHIFT OR P_ALT OR P_CTL)
669	D16D BA R	C HOV DX, ROMDATA ; ESTABLISH ADDRESSABILITY TO DATA AREA C HOV ES, DX ; USED BY THE ROM BIOS ROUTINES
670 671	0170 BE C2	C NOV ES,DX ; USED BY THE ROM BIDS ROUTINES
672	= 0070	C R MASK 14 EQU R CAPS STATE OR R NUM STATE OR R_SCROLL_STATE
673	= 0070	C R MASK IB EQU R MASK IA OR R ALT SHIFT OR R_CTL_SHIFT
674	= 007F	C D MASK IC FOUL D MASK IB OR R LEFT SHIFT OR R RIGHT_SHIFT
675	0172 26: 80 26 0017 R 80	C AND R_KB_FLAG, NOT R MASK_IC ; INSURE CAPS STATE, NUM LOCK C ; STATE AND SCROLL LOCK STATES ARE
676 677		C ; STATE AND SCROLL LOLK STATES ARE C ; RESET IN THE KEYBOARD BIOS ROUTINE
678		C ; AND THAT THE ALT, CTRL, AND SHIFT
679		C ; KEYS ARE NOT DEPRESSED
680	= 0070	C R_MASK_2 EQU R_CAPS_SHIFT OR R_NUM_SHIFT OR R_SCROLL_SHIFT
681	0178 26: 80 26 0018 R 8F	C AND R_KB_FLAG_1,NOT R_MASK_2 ; INSURE THE ROM BIOS ROUTINE ; KNOHS THE CAPS LOCK, HUM LOCK, AND
682		C ; KNOWS THE CAPS LOCK, NOT LOCK, AND C ; SCROLL LOCK KEYS ARE NOT DEPRESSED
683 684		C ASSUME ES:NOTHING
685		C
686	017E	C PROC_RESET_1:
687	017E E9 021F R	C JMP ROUTE ; ROUTE THE KEYSTROKE AS NEEDED
688		C ; CLICK KEY WAS PRESSED - TOGGLE THE KEYBOARD CLICKER STATE
689 690		
691	0181	C FROC_CLICK:
692	0181 F6 06 0000 R 80	C TEST TERM_MODE,T_HOST ; ARE WE IN PERSONAL COMPUTER MODE?
693	0186 75 14	C JHZ PROC_CLICK_1 ; NO, JUST PASS ON CLICK KEY C XOR KED_STATE,CLICK_STATE ; TOGGLE CLICKER STATE
694	0188 80 36 0000 R 04	
695 494	016D B0 09	C MOV ALICLICK_ON ; SET UP TO TUPH CLICHER ON C TEST KED_STATE.CLICK_STATE ; IS CLICHER OFFS
695 697-	018F F6 D6 0000 R D4 0194 75 02	C TEST KED_STATE,CLICK_STATE ; 15 CLICKLN GFA: C JNZ SET_CLICKER ; YES, CONTINUE
698	0196 50 08	C MOV AL, CLICK_OFF ; NO, SET UP TO TURN CLICKER OFF
699	0198	C SET CLICKER:
700	0198 BA 03E4	C MOV DX,COMMAND_PORT ; TURN KEYEDARD CLICKER CH CR OFF
701	019B EE	C OUT DX,AL
702 703	0390	C C PROC_CLICK_1:
703	019C E9 021F R	C JHP ROUTE ; ROUTE THE KEYSTROKE AS NEEDED

-

# 4,641,262

53

705

706

707

708

709

710

712

713

714

715

716 717

718

719

720

721

722

723 724

725

726 727

728

729

730

731 732

733 734

735

736 737

738

739

740

741 742

743 744

745

750

751 752

753

754

755

756 757

758

759

76 D

761

762

763

764

765

766 767

76 B

769

770

772

773 774

775

776 777

778 779

780 781

782

783

784

785

786

787

788

769

790 791

792

793

794

795

796

797

019F

019F

01A4

0146

DIAB

0180

01B2

01B4

01BC

01BF

01BF

01C4

01C6

61C8

01CA

01CA

0100

0100

DICE

01D0

01D1

01D1

0106

0108

DDD

OIED

01E0

01E5

01E7

01EC

01EF

01EF

01F4

01F6

01F9

01F9

0264

0211

0216

0217

C с; START DECIMAL DIGIT INPUT c C PROC\_DECIMAL\_KEY: TERM\_MODE,T\_HOST ; ARE WE IN PC MODE? PROC\_DECIMAL\_KEY\_1 ; NO, DO NOT FROCESS THIS KEY DEC\_DIGIT,3 ; INDICATE 3 DIGITS EXPECTED PC\_STATE,P\_ALT ; IS THE PERSONAL COMPUTER IN ALT MODE? F6 06 0000 R 80 C C TEST 75 16 JNZ C6 06 0003 R 03 F6 06 0001 R 20 С ноу C C TEST PROC\_DECIMAL\_KEY\_1 ; IS THE PERSONAL CONTINUE PROC\_DECIMAL\_KEY\_1 ; NO, CONTINUE AL,BREAK\_BIT OR PC\_ALT\_KEY ; YES, INDICATE THE ALT KEY SEND\_TO\_PC ; WAS RELEASED - THIS WILL TERMINATE ; THE DECIMAL INPUT FIELD IN PROGRESS 74 DA JZ 80 B8 C HOV E8 0367 R C C CALL PC\_STATE,NOT P\_ALT 01B7 80 26 0001 R DF с AND ē c PROC\_DECIMAL\_KEY\_1: 01BC EB 61 90 ROUTE ; ROUTE THE KEYSTROKE AS NEEDED С JMP c TOGGLE BETWEEN PERSONAL COMPUTER AND HOST PROCESSOR MODES С ; С с PROC\_SWITCH\_MODE: TERM\_MODE,T\_HOST ; IS THE TERMINAL IN HOST MODE? F6 06 0000 R 80 С TEST 74 04 С JZ SWITCH\_TO\_HOST ; NO, SWITCH TO HOST HODE c c AL,SET\_HODE\_PC ; YES, INDICATE SHITCH TO PC HODE B0 01 MOV SHORT SHITCH\_MODE\_1 FB 02 с JMP С SWITCH\_TO\_HOST: С B0 00 AL, SET MODE\_HOST ; INDICATE SWITCH TO HOST MODE c c MOV с с SWITCH\_MODE\_1: B4 00 AH, FUNC\_SET\_MODE ; INDICATE SET MODE HOV NDSP\_INT\_CCDE ; SHITCH TERMINAL MODE ; PROCESS NEXT PENDING INTERRUPT CD 50 C INT C3 с RET c c CTRL KEY - INDICATE NEXT KEY STROKE IS IN PERSONAL COMPUTER ; č CTRL CASE ÷ С PROC\_CTL\_KEY: С TERM\_MODE,T\_HOST ; ARE WE IN PC MODE? PROC\_CTL\_KEY\_1 ; NO, DO NOT PROCESS PC CTL KEY LOGICAL\_STATE,L\_CTL ; INDICATE PC CTL CASE F6 06 0000 R 80 С TEST 75 05 с с JNZ 80 OE 0002 R 40 OR С PROC\_CTL\_KEY\_1: С 01DD EB 40 90 ¢ JMP ROUTE & ROUTE KEYSTEDKE AS NEEDED с С PC ALT KEY - INDICATE NEXT KEY STROKE IS IN PERSONAL COMPUTER ĵ c c ; ALT CASE PROC\_ALT\_KEY: С TERM\_MODE,T\_HOST ; ARE WE IN FC MODE? F6 06 0000 R 80 С TEST PROC\_ALT\_KEY\_1 ; NO, DO NOT FROCESS PC ALT KEY LOGICAL\_STATE,L\_ALT ; INDICATE PC ALT CASE 75 05 C JHZ 80 DE 0002 R 80 С CR С С PROC\_ALT\_KEY\_1: ; ROUTE KEY STROKE AS NEEDED 01EC EB 31 90 С JMP ROUTE C PAUSE KEY - STOP ALL NORMAL PERSONAL COMPUTER PROCESSING UNTIL THE USER PRESSES ANOTHER NON-SHIFT KEY С 3 С 3 с PROC\_PAUSE: TERH\_HODE,T\_HOST & ARE WE IN HOST MODE? F6 06 0000 R 80 С TEST ; ND, PERSCHAL COMPUTER MODE ; YES, NO SPECIAL PROCESSING PAUSE\_1 ROUTE 74 03 r. 17 EB 27 90 JHP С PAUSE\_1: С CT: 80 0E 0002 8 20 LOGICAL\_STATE, L\_DAUGE & INDICATE WE ARE IN PAUSE STATE С С PROC\_INTERRUPTS ; PROCESS ALL OTHER PENDING INTERRUPTS 01FE E8 0000 E CALL С 0201 BA ---- R č ноу DX. ROMDATA ; POINT TO ROM BIDS DATA AREA 6E C2 С HOV ES,DX ES:ROMDATA С ASSUME С 0206 26: 80 3E 0049 R 07 ; IS THE MONOCHROME CARD BEING USED? CI1P CRT MODE,7 С ; YES, NO FROCESSING NEEDED ; NO, TUPN ON DISPLAY DURING PAUSE 020C 74 08 020E BA 03D8 С PAUSE\_2 DX,03D8H JE HOV С 26: A0 0065 R ¢ MOV AL, CRT\_MODE\_SET 0211 26 0215 EE DX.AL С . DIΠ ASSUME ES:NOTHING c С С PAUSE\_2: STI 0216 FB ; ENABLE INTERRUPTS c č ; WAIT FOR THE USER TO PRESS & KEY PAUSE\_3: 0217 F6 06 0002 R 20 021C 75 F9 С TEST LOGICAL\_STATE,L\_PAUSE C JNZ PAUSE\_3 с ; PROCESS ALL OTHER PENDING INTERUPTS 021E C3 С RET С C INCLUDE NDSPKBD2.ASM С "我就我我就会法办我就在手法提到武武政士不法承承个人父子的主义的人,我不能不能不能不能没有不能要不能不能有些没有不能不能不能不能不能。" С ;\*\* С ;× ÷\* ROUTE THE CONVERTED SCAN CODE TO THE PERSONAL COMPUTER OR с

				4,0	641,262	
		55				56
			C ;*	ROUTE	THE ORIGINAL SCAN	CODE TO THE 3278/3279
			C ;* C ;****	*****	*****	"我安宁有年老这就是这些不可能不可能是有有有有有有有不可不不有不不有有有有不不
021F			C C ROUTE:			
021F	F6 D6 D00D 74 07	R 80 .	с с с;	TEST JZ	TERH_MODE,T_HOS ROUTE_PC	ST ; IS THE TERMINAL IN HOST MODE? ; NO, SEND THE CONVERTED SCAN CODE TO THE PERSONAL COMPUTER
			C ; C ; C ;		RMINAL IS IN HOST ROKES TO THE 3278	F PROCESSOR MODE - SEND ALL 3/3279
	A0 0004 R E8 02FD R		с с с	HOV Call	AL.KBD_SCAN_CO	E ; PICK UP ORIGINAL SCAN CODE ; SEND THE SCAN CODE TO THE 3278/3274
2220	C3		C C	RET		; PROCESS NEXT PENDING INTERRUPT
					RMINAL IS IN PERS	SONAL COMPUTER HODE - SEND THE CONVERT DHAL COMPUTER
	F6 06 0002 74 15	R 20	C ROUTE C C	PC: TEST JZ	LOGICAL_STATE, ROUTE_TO_PC	L_PAUSE ; ARE WE IN PAUSE STATE? ; ND, ROUTE KEYSTROKE TO THE PC
			С С;	THIS K	EY STROKE IS USED	TO EXIT PAUSE STATE
	80 26 0002 F6 06 0002		с с с с	AND TEST		COT L_PAUSE ; NO LONGER IN PAUSE STATE L_PC_PORT_AVAIL ; DID A PC OUTPUT PORT ; AVAILABLE INTERRUPT OCCUR WHILE ; HERE IN PAUSE STATE?
240	74 08 81 26 0002 E8 03A7 R	REF		JZ AHD CALL	EXIT_PAUSE_1 LOGICAL_STATE,1 XMIT_TO_PC	; NO, CONTINUE PROCESSING NOT L_PC_PORT_AVAIL ; YES, FROCESS IT ; SEND PENDING KEY STROKES TO THE PC
0248 0248	C3		C EXIT_ C C	RET		; IGHORE THIS KEY STROKE
249			C ROUTE	TO_PC:		- TO TUTE & FINISTED COPPE
	80 3E 0005 79 01	R 00	C C	CMP JNS	PC_SCAN_CODE.0 ROUTE_PC1	; IS THIS A FUNCTION CODE? ; NO, IT IS A SCAN CODE
250 251	C3		C C ROUTE	RET PC1:		; YES, DO NOT SEND IT TO THE PC
			с ; С; С;	IF THE		DECIMAL DIGIT INPUT MODE, CONVERT THE D KINAD DIGIT COAN CODE
	80 3E 0003 74 30	5 R 00		CHP JE	DEC_DIGIT,C TEST_SHIFTS	; ARE DECIMAL DIGITS EXPECTED? ; NO, CONTINUE PROCESSING
	AD 0005 R		C C C ;	NOV	AL, PC_SCAN_COD	E ; INSURE THE KEY STROKE WAS A DECIMA DIGIT
5B 5D	3C 02 72 24		с с	CMP JB	AL,2 CANCEL_DIGITS	; NOT DIGIT - CANCEL DIGIT INFUT HOD
25F 261	3C 0B 77 20		с с	CMP JA	AL,11 CANCEL_DIGITS	; NOT DIGIT - CANCEL DIGIT INPUT MOD
0263	F6 C2 E0		C	TEST	DL,U+A+C	
0266	75 1B		с с	JNZ	CANCEL_DIGITS	; NOT DIGIT - CANCEL DIGIT INPUT MOD
0268	80 CA 20		C ; C ; C ;	DR	DL.A	; VALID DIGIT - CONVERT IT TO A NUME KEY PAD DIGIT KEY WITH THE ALT KEY PRESSED
	FE C8 FE C8		C	DEC	AL Al	
26F	8D 1E 0006	5 R	с	LEA	BX,PC_DIGITS	
0273 0274	D7 A2 0005 R		C C	XLAT MOV	PC_DIGITS PC_SCAN_CODE,A	L
0277	FE DE 0003	5 R	с с	DEC	DEC_DIGIT	; A DIGIT HAS BEEN PROCESSED
027B	75 0B 80 CA 02		C C	JNZ OR	TEST_SHIFTS DL,BA	; NOT LAST DIGIT ; LAST DIGIT - INDICATE ALT KEY HAS
			C ; C			RELEASED AFTER SENDING LAST DIGIT
600	EB 06 90		с с;	JHP	TEST_SHIFTS	S KEYED DURING DECIMAL DIGIT INPUT MOD
			C ; C	CANCEI	L DECIMAL DIGIT I	
283 283	C6 06 0003	3 R 00	C C ; C	L_DIGITS: MOV	DEC_DIGIT,0	INDICATE DECIMAL DIGITS ARE NOT EXPECTED
			C ; C ; C			CODES TO THE PERSONAL COMPUTER TO MATC ITH THE SCAN CODE
0288	81 61		C TEST_	SHIFTS:	רו אי	; DETERMINE WHICH SHIFT STATES MUST
028A	8A CA 32 DE 0001	L R	C C	MOV XOR	CL.DL CL.PC_STATE	; CHANGED
	80 El EC 74 3C		c c	AND JZ	CL,0EDH Shifts_match	; STATES MATCH, CONTINUE
			C ; C ; C ;	DL :	IS POINT: = NEEDED PERSONAL = MISMATCHED STAT	COMPUTER KEYBDARD STATE

•

•

			57				58
891	0293	F6 C1 80	c		TEST	CL.U	; DOES SHIFT CASE MATCH?
892	0296	74 OC	c		JZ	CASE OK	; YES, CONTINUE
893	029B	B0 2A	c		MOV		; NO, INDICATE SHIFT KEY PRESSED
894	029A	F6 C2 80	C		TEST	DL.U	; IS UPPER CASE NEEDED?
895	029D	75 02	C		JNZ	SEND_CASE	; YES, CONTINUE
896 897	029F 02A1	OC 80	с с		OR	AL, BREAK_BIT	; NO, INDICATE SHIFT KEY RELEASED
898	02A1	E8 0367 R	. C	SEND_CA	CALL	SEND_TO_PC	SEND KEY STROKE TO PC
899	0244	20 0307 K	č	CASE OK		2000_00_00	, still ker sikake to to
900			c	-			
901	D2A4	F6 C1 20	c		TEST	CL,A	; DOES ALT CASE HATCH?
902 903	D2A7		c		JZ	ALT_OK	; YES, CONTINUE
904	02A9	F6 C2 2D	с С		HOV TEST	AL,PC_ALT_KEY DL,A	; NO. INDICATE ALT KEY PRESSED ; IS ALT CASE NEEDED?
905		75 02	c c		JNZ	SEND ALT	; YES, CONTINUE
506	02B0	0C 80	C		DR	AL, BREAK_BIT	; NO, INDICATE ALT KEY RELEASED
907	02B2	-	C	SEND_AL			
908 909	02B2	E8 0367 R	c	117 04	CALL	SEND_TO_PC	; SEND KEY STROKE TO PC
910	0285		C C	ALT_OK:			
911	0285	F6 C1 40	č		TEST	CL,C	; DOES CTL CASE MATCH?
912	02B8	74 DC	C		JZ	CTL_OK	; YES, CONTINUE
913	02BA	B0 1D	c		HOV	AL,PC_CTL_KEY	; NO, INDICATE CIRL KEY PRESSED
914	02BC	F6 C2 40	c		TEST	DL,C	I IS CIL CASE NEEDED?
915 916	02BF 02C1	75 02 0C 80	c c		202	SIND_CTL AL,EREAK BIT	) VEC, DESTINUE ) ND, INDICATE CTRL KEY RELEASED
917	0203		c	SEND_CT	DR Lt	ALIEKLAK_DII	, NO, INDICATE CIRE KET RELEASED
918	02C3	E8 0367 R	č		CALL	SEND_TO_PC	SEND KEY STROKE TO PC
919	0206		c	CTL_OK:			
920			ç				- CAUG DERCOMAL COMPUTER REYROADD STATE
921 922	02C6 02C8	84 CA 80 E1 E0	c c		NOV AND	CL,DL	; SAVE PERSONAL COMPUTER KEYBOARD STATE
923	D2CB				HOV	CL,DEOH PC STATE,CL	
924			Ĉ			10_0/2/02	
925	02CF		c	SHIFTS_	MATCH:		
926	02CF	A0 0005 R	c		HOV		; SEND KEY STROKE TO PERSONAL COMPUTER
927 928	02D2	E8 0367 R	c c		CALL	SEND_TO_PC	
929	02D5	F6 C2 03	C		TEST	DL.B OR BA	DOES THE PERSONAL COMPUTER NEED TO BE
930			č	3		2010 00 20	TOLD THAT THE KEY WAS RELEASED ALSO?
931	02D8	74 11	c		JZ	RESET_SHIFTS	; NO, CONTINUE
932			C				
933 934	02DA 02DC	0C 80 F6 C2 02	c c		OR	AL, BREAK_BIT	; YES, INDICATE KEY WAS RELEASED ; WHAT KEY MUST BE RELEASED?
935	02DF	74 07	c c		TEST JZ	DL,BA BRFAK KEYSTROKE	THE KEY STROKE JUST SENT
936		BO BB	č		MOV	AL BREAK BIT OR	PC_ALT_KEY ; THE ALT KEY WAS RELEASED
937	02E3	80 26 0001			AND	PC_STATE,NOT P_4	ALT ; UPDATE PC KEYBOARD STATE
938 939	02E8 02E8	E8 0367 R	c c	BREAK_K	EYSTROKE		
940	0120	20 0307 A	с с		CALL	SEND_TO_PC	
941			Ċ		*****	*****	**************************************
942 943			c	3*			
944			c c	;* ;*	COMPLETE	PRUCESSING A VALI I STATE FLACS SO	ID KEY STROKE - RESET THE PERSONAL * THE NEXT KEY STROKE IS NOT INTERFRETED *
945			c	3*		S IN CTL OR ALT P	
946							*
947			с	;*			
			c c		*******	*****	* 家餐餐餐餐餐餐餐餐餐餐餐餐餐餐餐餐餐餐餐餐餐餐店店 化化化化化化化化化化化
946	02EB	-	с с с	; * <del>* * * * *</del> *		*******	* 我我我就像我爷孩里家不能要我想到我我的别家的那些我就要我我不会有人的家父女女子弟亲亲
	02EB 02EB	80 26 0002	с с с с			LDGICAL_STATE,NC	
948 949 950 951	02EB		с с с с	; * <del>* * * * *</del> *	AND		DT L_CTL+L_ALT
948 949 950 951 952			C C C R 3F C C C	; * <del>* * * * *</del> *	HIFTS:		
948 949 950 951 952 953	02EB		C C C R 3F C C C C	;******	HIFTS: AND RET	LDGICAL_STATE,HC	DT L_CTL+L_ALT ; process next pending interrupt
948 949 950 951 952	02EB		C C C R 3F C C C	;******	HIFTS: AND RET	LDGICAL_STATE,NC	DT L_CTL+L_ALT ; process Next Pending Interrupt ************************************
948 949 950 951 952 953 953 955 955 956	02EB		C C C R 3F C C C C C C C C C C C C C C C C C C C	; ****** RESET_SI ; ****** ; * ; *	HIFTS: AND RET	LDGICAL_STATE,NC	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 952 953 954 955 955 956 957	02EB		C C C R 3F C C C C C C C C C C C C C C C C C C C	;****** RESET_SI ;****** ;* ;*	HIFTS: AND RET THE 3274	LDGICAL_STATE,NC N***********************************	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 952 953 955 955 956 957 958	02EB		C C C C C C C C C C C C C C C C C C C	;****** RESET_SI ;****** ;* ;*	HIFTS: AND RET THE 3274	LDGICAL_STATE,NC N***********************************	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 952 953 954 955 955 956 957	02EB		C C C R 3F C C C C C C C C C C C C C C C C C C C	; ******* RESET_Si ; ******* ; * ; * ; * ; *	11FTS: AND RET THE 327/	LDGICAL_STATE,NC N***********************************	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 951 952 953 955 956 956 956 958 959	02F0 02F1		C C C R 3F C C C C C C C C C C C C C C C C C C C	;****** RESET_SI ;****** ;* ;*	11FTS: AND RET THE 327/	LDGICAL_STATE,NC MAXADAXANANANANANANANANANANANANANANANANAN	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 952 953 955 955 955 955 959 961 962	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	; ******* RESET_Si ; ******* ; * ; * ; * ; *	HIFTS: AND RET THE 3274 ********* MDV MDV	LDGICAL_STATE,NC N***********************************	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 953 953 954 9556 9556 9557 9558 9559 960 961 961 963	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005	R 3F C C C C C C C C C C C C C C C C C C	; ******* RESET_Si ; ******* ; * ; * ; * ; *	HIFTS: AND RET THE 3274	LDGICAL_STATE,NC N***********************************	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 952 955 955 955 955 957 957 957 957 958 957 958 959 960 961 962 964	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	; ******* RESET_Si ; ******* ; * ; * ; * ; *	HIFTS: AND RET THE 3274 ********* MDV MDV	LDGICAL_STATE,NC N***********************************	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 953 953 954 9556 9556 9557 9558 9559 960 961 961 963	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	;****** RESET_SI ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 3274 ********* MDV MDV	LDGICAL_STATE,NC N***********************************	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 952 9553 9554 9555 9556 9557 9558 9557 9558 9557 9558 9560 961 962 964 964 964 9653 964 9657	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	;****** RESET_SI ;****** ;* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;	HIFTS: AND RET THE 3274 ********* HOV HOV JMP	LDGICAL_STATE,NC MAXADANANA B/3279 KEYBOARD F MAXADANANANANANANANANANANANANANANANANANAN	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 952 955 955 955 955 957 955 957 957 957 957	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	; ******* RESET_SI ; ******* ; * ; * ; * ; * ; * ; * ; * ;	HIFTS: AND RET THE 3274 ********* HOV HOV JMP	LDGICAL_STATE,NC MAXADANANA B/3279 KEYBOARD F MAXADANANANANANANANANANANANANANANANANANAN	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 953 953 9554 9556 9557 9556 9557 9559 960 961 963 9645 9645 9645 9645 9645 9669	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	;****** RESET_SI ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	41FTS: AND RET THE 327/ MOV JMP	LOGICAL_STATE,NC MARKAN AND AND AND AND AND AND AND AND AND A	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 952 955 955 955 955 957 955 957 957 957 957	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	;****** RESET_SI ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	41FTS: AND RET THE 327/ MOV JMP	LDGICAL_STATE,NC MAXADANANA B/3279 KEYBOARD F MAXADANANANANANANANANANANANANANANANANANAN	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT ************************************
948 949 950 951 952 955 955 955 955 955 961 962 965 961 962 965 964 965 965 965 965 965 965 965 965 965 965	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	;****** RESET_SI ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 327/ HDV HDV JNP	LOGICAL_STATE,NC MARKAN AND AND AND AND AND AND AND AND AND A	T L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT AAS SIGNALLED OVERRUN H ; SET UP OVERRUN SCAN CODE ; REQUIRED STATE = CURRENT STATE ; ROUTE THE OVERRUN NOLATION TC THE ; PERSONAL COMPUTER OR TO THE 3278/3279
948 949 950 951 952 955 955 955 955 957 955 957 955 957 958 957 961 962 964 964 964 964 964 965 964 965 957 957 957 950 957 957 957 957 957 957 957 957 957 957	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	;******* RESET_SI ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 327/ HOV HOV JHP	LDGICAL_STATE,NC ACCEPTING AND ACCEPTING B/3279 KEYBOARD A PC_SCAN_CODE,OFF DL,PC_STATE ROUTE Amme: SEND_TO_3278 and a keystroke t	T L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT AAS SIGNALLED OVERRUN H ; SET UP OVERRUN SCAN CODE ; REQUIRED STATE = CURRENT'STATE ; ROUTE THE OVERRUN INDICATION TC THE ; PERSONAL COMPUTER OR TO THE 3278/3279
948 949 950 951 952 955 955 955 955 955 957 955 957 957 957	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	;******* RESET_SI ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 327/ HOV HOV JHP	LDGICAL_STATE, NC 8/3279 KEYBOARD F 9C_SCAN_CODE, OFF DL, PC_STATE ROUTE	T L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT AAS SIGNALLED OVERRUN H ; SET UP OVERRUN SCAN CODE ; REQUIRED STATE = CURRENT'STATE ; ROUTE THE OVERRUN INDICATION TC THE ; PERSONAL COMPUTER OR TO THE 3278/3279
948 949 950 951 952 955 955 955 955 957 955 957 955 957 958 957 961 962 964 964 964 964 964 965 964 965 957 957 957 950 957 957 957 957 957 957 957 957 957 957	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	;******* RESET_Si ;******** ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 327/ HOV MOV JNP cedure N. cetion: Si ut: AL =	LOGICAL_STATE,NC ACCOUNTS AND ACCOUNTS AND	THE SET UP OVERRUN SCAN CODE REQUIRED STATE = CURRENT 'STATE ROUTE THE OVERRUN INDICATION TO THE PERSONAL COMPUTER OR TO THE 3278/3279 The 3270/3279 So the 3270/3279
948 949 951 951 952 9553 955 955 955 955 957 955 957 957 957 958 957 961 962 964 965 964 965 964 965 964 965 966 971 973 975 975 975 977	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F C C C C C C C C C C C C C C C C C C	;******* RESET_SI ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 3274 HOV HOV JNP sectore N. stien: Si ut: AL = put: Sca	LDGICAL_STATE,NC NAME AND ADDED ADDED B/3279 KEYBOARD F PC_SCAN_CODE.OFF DL,PC_STATE ROUTE ADD TE ADD TE ADD TO ADD T	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT AAS SIGNALLED OVERRUN FH ; SET UP OVERRUN SCAN CODE ; REQUIRED STATE = CURRENT 'STATE ; ROUTE THE OVERRUN INICATION TC THE ; PERSONAL COMPUTER OR TO THE 3278/3279 in the 3278/3279 in the 3278/3279 in the sent SUFFER_3278 and/or sent to 3278/3279
948 949 950 951 952 955 955 955 955 955 955 961 962 965 965 964 965 964 965 964 965 964 965 964 965 971 972 973 975 977 977	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	;******* RESET_SI ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 3274 HOV HOV JNP sectore N. stien: Si ut: AL = put: Sca	LDGICAL_STATE,NC NAME AND ADDED ADDED B/3279 KEYBOARD F PC_SCAN_CODE.OFF DL,PC_STATE ROUTE ADD TE ADD TE ADD TO ADD T	THE SET UP OVERRUN SCAN CODE REQUIRED STATE = CURRENT 'STATE ROUTE THE OVERRUN INDICATION TO THE PERSONAL COMPUTER OR TO THE 3278/3279 The 3270/3279 So the 3270/3279
948 949 951 951 952 9553 955 955 955 955 957 955 957 957 957 958 957 961 962 964 965 964 965 964 965 964 965 966 971 973 975 975 975 977	02F0 02F1 02F1 02F1 02F6	C3 C6 06 0005 BA 16 0001	R 3F CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	;******* RESET_Si ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 327/ HOV MOV JNP Sedure N. Stion: Si Stion: Si Sut: AL = Dut: Scan	LOGICAL_STATE,NC ACCOUNTS AND ACCOUNTS AND	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT AAS SIGNALLED OVERRUN FH ; SET UP OVERRUN SCAN CODE ; REQUIRED STATE = CURRENT 'STATE ; ROUTE THE OVERRUN INICATION TC THE ; PERSONAL COMPUTER OR TO THE 3278/3279 in the 3278/3279 in the 3278/3279 in the sent SUFFER_3278 and/or sent to 3278/3279
948 949 950 951 952 9554 9556 957 9558 957 958 960 961 963 964 963 964 963 964 963 964 963 964 9657 960 963 964 972 972 973 975 9778 9778 9778 979	02F0 02F0 02F1 02F1 02F6 02FA	C3 C6 06 0005 8A 16 0001 E9 021F R	R 3F CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	;******* RESET_Si ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 3274 HOV HOV JNP sectore N. stien: Si ut: AL = put: Sca	LOGICAL_STATE,NC ACCOUNTS AND ACCOUNTS AND	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT AAS SIGNALLED OVERRUN FH ; SET UP OVERRUN SCAN CODE ; REQUIRED STATE = CURRENT 'STATE ; ROUTE THE OVERRUN INICATION TC THE ; PERSONAL COMPUTER OR TO THE 3278/3279 in the 3278/3279 in the 3278/3279 in the sent SUFFER_3278 and/or sent to 3278/3279
948 949 950 951 952 953 9554 9556 957 9558 956 961 963 964 963 964 964 963 964 964 9657 960 964 9657 960 963 964 965 970 971 972 9778 9778 9778 9778 9778 9778 9778	02EB 02F0 02F1 02F1 02F6 02FA 02FD 02FD 02FD	C3 C6 06 0005 8A 16 0001 E9 021F R	R 3F CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	;******* RESET_Si ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 327/ ************************************	LOGICAL_STATE,NC ACC NEAR AX LOGICAL_STATE,NC ACC NEAR AX LOGICAL_STATE,NC ACC NEAR AX LOGICAL_STATE,NC ACC NEAR AX BX	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT AAS SIGNALLED OVERRUN TH ; SET UP OVERRUN SCAN CODE ; REQUIRED STATE = CURRENT 'STATE ; ROUTE THE OVERRUN INDICATION TC THE ; PERSONAL COMPUTER OR TO THE 3276/3279 to the 3270/3279 ode to be sent SUFFER_3278 and/or sent to 3278/3279
948 949 951 951 952 9554 9555 9556 957 9558 957 957 957 961 962 964 965 964 965 964 965 964 965 964 965 964 965 971 972 974 975 977 977 977 977 977 977 978 979 979 979	02EB 02F0 02F1 02F1 02F6 02FA 02FA	C3 C6 06 0005 8A 16 0001 E9 021F R	R 3F CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	;******* RESET_Si ;******* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;*	HIFTS: AND RET THE 3274 HDV HDV JHP Cedure N. Cition: So ut: AL = put: Scar PuSH	LDGICAL_STATE,NC NAME AND ADDRESS AND ADD	DT L_CTL+L_ALT ; PROCESS NEXT PENDING INTERRUPT AAS SIGNALLED OVERRUN TH ; SET UP OVERRUN SCAN CODE ; REQUIRED STATE = CURRENT'STATE ; ROUTE THE OVERRUN INDICATION TC THE ; PERSONAL COMPUTER OR TO THE 3276/3279 to the 3270/3279 ode to be sent SUFFER_3278 and/or sent to 3278/3279

					59				00	
984	0300	57			с		PUSH	DI		
985					C					
986	0301		3E (	022			HOV	DI.BUFFER_3278_	TAIL   PICK UP CURRENT BUFFER ADDRESS	
987	0305		DF		C		HOV	BX,DI	I CALCULATE ADDRESS OF NEXT BYTE IN	
988	0307	43			C C		INC	BX	3 BUFFER	
989 990	0308		16 0	020			LEA	DX.BUFFER_3278_	END	
991	030C 030E		DA		C		CHP	BX,DX		
992	-		04		C C		JHE	SEND_TO_3278_1		
993	0310 0314	θU	1E (	010	R C C		LEA	BX,BUFFER_3278		
994	0314				с С	SEND_TU	_3278_1:			
995	0314	TR	1E 0	020			CHP	BY BUEFED 3278	HEAD ; IS THE BUFFER FULL?	
996	0318		IA		r c		JE	OVERRUN_3278	; YES, WE HAVE OVERRUN THE 3278/3279	
997					č		51	OTENRON_DETO	,,	
998	031A	88	05		č		HOV	[DI],AL	; NO, PUT THE SCAN CODE IN THE BUFFER	
999	031C		3E (	024			HOV		T,DI ; SAVE ADDRESS OF THIS BYTE	
1000	0320	89	1E (	022			HOV	BUFFER 3278 TAI	L,BX ; SAVE NEXT BUFFER BYTE ADDRESS	
1001					C					
1002	0324				c	SEND_TO	_3278_2:			
1003	0324		03E3	2	С		MOV	DX,KB_CTL	I SEE IF THE 3278 KEYBOARD DUTPUT PORT	
1004	0327				c		IN	AL,DX	; IS BUSY	
1005	0328		02		C		TEST	AL,02H	PORT TO DUCK THE KEY STROKE WILL BE	
1006	032A	74	03		C		JZ	SEND_T0_3278_3	; PORT IS BUSY - THE KEY STROKE WILL BE SENT WHEN THE 3276/3279 ACKNOWLEDGE!	
1007 1008					C C	3			THE KEY STROKE IT IS WORKING ON	
1009	032C	FA	0330		C C	;	CALL	XMIT_T0_3278	PORT IS FREE - SELED THE NEXT KEY	
1010		.0		· K	c	;			STROKE TO THE 3278/3279	
1011					č					
1012	032F				c	SEND TO	_3276_3:			
1013	032F	5F			C		POP	DI	; RESTORE REGISTERS	
1014	0330	54			C		POP	DX		
1015	0331	5B			c		POP	BX		
1016	0332	58			c		FOP	AX		
1017	0333	C3			C		RET		; RETURN TO CALLER	
1018					c					
1019	0334	<b>6</b> 0			с С	OVERRUN		DT DUCEED 7070	LAST ; PICK UP ADDRESS OF LAST BYTE USED	
1020	0334		3E 0				MOV MOV	DI,BUFFER_5270_	; REPLACE LAST KEY STROKE IN BUFFER	
1021 1022	0338	LD	05 0	.0	C C	;	HUY	DITE PIR (DI),0	WITH OVERRUN SCAN CODE	
1023	033B	FB	E7		C	,	JMP	SEND TO 3278 2	SEND KEY STROKES TO 3278/3279	
1024					č		2111		• • • • • • • • • • • • • • • • • • • •	
1025	0330				· C	SEND TO	_3278 EN	DP		
1026					С	_	-			
1027					c c					
1028							****	******	\$P\$\$P\$#A\$\$P\$\$P\$\$P\$#\$P\$P\$P\$P\$P\$P\$P\$P\$P\$P\$	
1029					c	;*	• •		,	
1030 1031					c		cedure Na	ame: XMIT_TO_327	в	
1032					c c	;* •# Eum		and the next new	ding kourtooka ta tha 3076/3079	
1033					с С	;* Fund	ction: Se	end the next pen	ding keystroke to the 3278. 3279	
1034					č		ut: BUFFE	ER 3278 contains	scan codes to be sent to the display	
1035					č	18				
1036					c		out: the	oldest scan cod	e sent to the 3278/3279 display	•
1037					C					
1038					c	; *******	***	****	<b>水水水在炭水浓水水水水水水水水水水水水水水水水水水水水水水水水水水水水水水水水</b>	
1039					c					
1040	033D				C	XHIT_TO	_3278 PRC	DC NEAR		
1041					C					
1042	033D		1E (				HOV		HEAD ; SEE IF THERE ARE ANY KEY STROKES	
1043 1044	0341 0345		1E 0	1022	R C C		CHP	BX, BUFFER_3278_	TAIL ; IN THE BUFFER ; NO, JUST RETURN TO CALLER	
1045	0345	/4	16		C		JZ	XUT1_10_2510_2	, NO, JUST RETURN TO CALLER	
1045	0347	84	27		C		HOV	AH, IBX]	; PICK UP NEXT SCAN CODE TO BE SENT	
1047	0349				c		INC	BX	; POINT TO NEXT SCAN CODE IN BUFFER	
1048	034A			020			LEA	DX,BUFFER_3278_	-	
1049	034E	38	DA		c		CHP	BX,DX		
1050	0350				c		JHE	XMIT_T0_3278_1		
1051	0352	8D	IE C	010		h=	LEA	BX,BUFFER_3278		
1052	0356				_ <u>c</u>	XMIT_TO				
1053	0356	89	1E C	020			MOV	BUFFER_3278_HEA	D,BX	
1054 1055	035A	<b>D</b> 4	0757		C C		MONT	DV CHD BACC	ALLON THE 2000/2000 TO THTERRIPT	
				)	с С		MOV	DX,CHD_8255		
1056 1057	035D 035F				C C		HOV OUT	AL,SET_INTE_B DX,AL	CODE	
1058	0551				č		001	DAIRE		
1059	0360				č	XMIT TO	3278_2:			
1060	0360	BA	03E1		č		HOV L	DX, NDS KB OUT	SEND THE SCAN CODE TO THE 3276/3279	
1061	0363	88	C4		c		MOV	AL,AH		
1062	0365				c		OUT	DX,AL		
1063					C					
1064	0366				c	XHIT_TO	3278_3:			
1065	0366	<b>C</b> 3			C		RET		; RETURN TO CALLER	
1066					C			-		
1067	0367				C	XHIT_TO	_3278 END	)P		
1068					c					
1069 1070					C F				·····································	
1070					C C	3***** 3K			-creenceedommoncreentropics areas	
1072					C C		cedure N	ame: SEND_TO_PC	•	
1073					č	3* FIG			+	
1074					c		ction: S	end a keystroke	to the Personal Computer +	
1075					С	;*			÷	
1076					С	;* Inp	ut: AL =	keyboard scan c	ode to be sent ;	

61

62

•

							62
				€ Dutpu		n code added to BL puter	JFFER_PC and/or sent to Personal
			C ;+		*****	*****	* * * * * * * * * * * * * * * * * * *
0367			C C 51	END_TO_P	C PROC	NEAR	
0367			С	— — — Р	'USH	AX i	SAVE WORK REGISTERS
0368 0369			C C		iush iush	BX DX	
	57		C		USH	DI	
36B	88 3E 003	8 R	C C	м	IOV	DI.BUFFER PC TAIL	I ; PICK UP CURRENT BUFFER ADDRESS
036F	88 DF	•	C		iov	BX,DI i	CALCULATE ADDRESS OF NEXT BYTE IN
0371	43 8D 16 003	4 D	с с		NC	BX DUESED DC END	BUFFER
0376		OK	č		.EA :MP	DX,BUFFER_PC_END BX,DX	
	75 04		С	J	INE	SEND_TO_PC_1	
037A 037E	8D 1E 002	6 R	C C 51	L END_TO_P	.EA PC 1:	BX, BUFFER_PC	
			С		_		
37E 382	3B 1E 003 74 1A	o M	с с		:HP IE	BX,BUFFER_PC_HEAD OVERRUN_PC	) ; IS THE BUFFER FULL? ; YES, WE HAVE OVERRUN THE PERSONAL
			С;	·		-	COMPUTER
	88 05		с С		ov		NO, PUT THE SCAN CODE IN THE BUFFE
386			c		VOV	BUFFER_PC_LAST,DI	I I SAVE ADDRESS OF THIS BYTE K I SAVE NEXT BUFFER BYTE ADDRESS
	89 1E 003	OK	C C		IOV	DULLER MUTIVITY	- ; GATE HEAT BUILER DITE ADDREDD
38E 38E	BA 03E4		C 51 C	END_TO_P	C_2:	DX.STATUS PORT	SEE IF THE PERSONAL COMPUTER KEYBO
391	EC		C	I	N	AL,DX I	
	A8 10 75 03		C C		EST	AL, ST_PC_KB_BUSY SEND_TO_PC_3 :	FORT IS BUSY - THE KEY STROKE WILL
, <b>,</b> 74	13 03		C 3	J		5000_10_FC_3 1	SENT WHEN THE PERSONAL COMPUTER
			с;				ACKNOHLEDGES THE KEY STROKE IT :
396	E8 03A7 R		C 1 C	~	ALL	XHIT TO_FC	WORKING ON ; PORT IS FREE - SEND THE NEXT KEY
	LU UJAT K		С;	Ľ	~~~		STROKE TO THE PERSONAL COMPUTER
99			C C 51	ND_TO_P	C_3:		
99	5F		С	P	OP		; RESTORE REGISTERS
9A 9B	5A 5B		C C		OP	DX	
					np	EX .	
	58		с		OP	BX AX	
390	58		c c	P		XA	; RETURN TO CALLER
39C 39D 39E	58 C3		с с с о\	P R /ERRUN_P	OP ET	XA	
39C 39D 39E 39E	58 C3 88 35 003	A R	C C C C C C	ๆ Я ๆ_ <i>н</i> บяяз\ ห	OP ET C:	AX DI,BUFFER_PC_LAS1	T ; PICK UP ADDRESS OF LAST EYTE US
19C 19D 19E 19E 19E	58 C3 8B 3E 003 C6 05 FF	A R	C C C C C C C C C C C C C C C C C C C	P R VERRUN_P H H	IC: IC: IOV	AX DI,BUFFER_PC_LAS <sup>3</sup> BYTE PTRIDI),OFF	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE
59C 59D 59E 59E 542	58 C3 88 35 003	A R	C C C C C C C C	P R VERRUN_P H H	OP ET C:	AX DI,BUFFER_PC_LAST BYTE PTRIDI),OFF	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF.
39C 39D 39E 39E 39E 33A2 3A5	58 C3 8B 3E 003 C6 05 FF	A R	C C C C C C C C C C S S	P R VERRUN_P H H	OP ET C: OV OV	AX DI,BUFFER_PC_LAS <sup>3</sup> BYTE PTRIDI),OFF	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE
39C 39D 39E 39E 39E 382	58 C3 8B 3E 003 C6 05 FF	A R	C C C C C C C C C C C S F C C C C C C C	9 9 4 4 7 4 9_07_04	0P ET 0V 0V 0V MP 7C ENDP	AX DI,BUFFER_PC_LAS' BYTE PTRIDI),OFF SEND_TO_PC_2	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP
39C 39D 39E 39E 39E 39E 382	58 C3 8B 3E 003 C6 05 FF	A R	C C C C C C C C C C C C C C C C C C C	۹ ۳ ۳ ۲ ۳ ۹_07_04	0P ET 0V 0V 0V MP 7C ENDP	AX DI,BUFFER_PC_LAS' BYTE PTRIDI),OFF SEND_TO_PC_2	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE
039C 039D 039E 039E 039E 03A2	58 C3 8B 3E 003 C6 05 FF	A R	C C C C C C C C C C C C C C C C C C C	۹ ۲ ۲ ۲ ۲ ۲ ۹_0۲_04	OP ET C: IOV IOV MP C ENDP	AX DI,BUFFER_PC_LAS' BYTE PTRIDI),OFF SEND_TO_PC_2	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP
39C 39D 39E 39E 39E 33A2 3A5	58 C3 8B 3E 003 C6 05 FF	A R	C C C C C C C C C C C C C C C C C C C	q R P_MURRAY M J J Q_OT_DA Proces	OP ET C: OV OV MP C ENDP dure Ha	AX DI,BUFFER_PC_LAS BYTE PTRIDI),OFF SEND_TO_FC_2 mme: XMIT_TO_PC	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP
39C 39D 39E 39E 39E 39E 382	58 C3 8B 3E 003 C6 05 FF	A R	C C C C C C C C C C C C C C C C C C C	P R VERRUN_P H J J SND_TO_P Process Func t	OP ET C: OV OV MP C ENDP dure Ha	AX DI,BUFFER_PC_LAS BYTE PTRIDI),OFF SEND_TO_FC_2 mme: XMIT_TO_PC	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF. HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP
39C 39D 39E 39E 3A2 3A5	58 C3 8B 3E 003 C6 05 FF	A R	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	P R R H H J IND_TO_P Process Funct Input	OP ET C: OV OV MP C ENDP dure Ha ion: Se Co	AX DI,BUFFER_PC_LAS BYTE PTRIDI),OFF SEND_TO_FC_2 mme: XHIT_TO_PC and the next pendi mputer	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF. HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP
59C 59D 59E 59E 5A2 5A5	58 C3 8B 3E 003 C6 05 FF	A R	C C C C C C C C C C C C C C C C C C C	P R VERRUN_P M J J END_TO_P Proces Funct Input Outpu	OP ET C: OV MP C ENDP dure Ha ion: Se Co	AX DI,BUFFER_PC_LAST BYTE PTRIDI),OFF SEHD_TO_FC_2 mme: XMIT_TO_PC and the next pendi imputer R_PC contains sca	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF. HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng keystroke to the Personal
39C 39D 39E 39E 3A2 3A5	58 C3 8B 3E 003 C6 05 FF	A R	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	P R VERRUN_P H J U VD_TO_P Process Funct Input	OP ET C: OV MP C ENDP dure Ha ion: Se Co : BUFFE t: olde	AX DI,BUFFER_PC_LAST BYTE PTRIDI),OFF SEND_TO_FC_2 me: XMIT_TO_PC and the next pendi imputer R_PC contains sca st scan code sent	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF. WITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng heystroke to the Personal un codes to be sent to the FC
89C 89D 89E 882 885 885 885 885	58 C3 8B 3E 003 C6 05 FF	A R	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	P R R VERRUN_P H J J SND_TO_P Process Func t Input Outpu	OP ET C: OV OV MP C ENDP dure Ha ion: Se Co : EUFFE t: olde	AX DI,BUFFER_PC_LAST BYTE PTRIDI),OFF SEHD_TO_FC_2 mme: XHIT_TO_PC and the next pendi imputer R_PC contains sca st scan code sent	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF. HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng heystroke to the Personal on codes to be sent to the FC : to the Personal Computer
9C 9D 9E 9E 42 45 47	58 C3 88 3E 003 C6 05 FF EB E7		CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	P R R VERRUN_P H J J SND_TO_P Process Funct Input Output	OP ET C: OV MP C ENDP dure Ha ion: Se Co : BUFFE t: olde	AX DI,BUFFER_PC_LAST BYTE PTRIDI),OFF SEND_TO_FC_2 and the next pendi imputer R_PC contains sca st scan code sent	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF. HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng heystroke to the Personal an codes to be sent to the FC : to the Personal Computer
59C 59D 59E 542 545 545 547 47	58 C3 88 3E 003 C6 05 FF EB E7 EB E7 EB 1E 003	6 R		P R R VERRUN_P H J J SND_TO_P Fonct Funct Input Output SIT_TO_P( M	OP ET C:: OV MP c ENDP dure Ha dure Ha ion: Se Co : EUFFE t: olde	AX DI,BUFFER_PC_LAS BYTE PTRIDI),OFF SEND_TO_FC_2 me: XHIT_TO_PC and the next pendi imputer R_PC contains sca st scan code sent NEAR BX,BUFFER_PC_HEAD	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng keystroke to the Personal on codes to be sent to the FC to the Personal Computer
90 99 99 8 4 5 7 7 8 8 7 7 8 8 8 7 7 8 8 8 7 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 7 8 8 8 7 8 8 7 8 8 7 8 8 7 8 8 8 7 8 8 7 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 8 7 8 8 8 7 8 8 8 8 7 8 8 8 7 8 8 8 8 7 8 8 8 8 7 8 8 8 7 8 8 8 8 7 8 8 8 8 7 8 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 8 7 8 8 8 8 7 8 8 8 8 7 8 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 7 8 8 8 8 7 8 8 8 7 8 8 8 8 8 7 8 8 8 8 7 8 8 8 8 7 8 8 8 8 8 8 8 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	58 C3 88 3E 003 C6 05 FF EB E7	6 R	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	P R R VERRUN_P H J J SND_TO_P Fonct Funct Input Output SIT_TO_P( M	OP ET C:: OV MP C ENDP dure Ha ion: Se Co : EUFFE t: olde ****** C FROC OV	AX DI, BUFFER_PC_LAST BYTE PTRIDI), OFF SEHD_TO_FC_2 me: XHIT_TO_PC and the next pendi imputer R_PC contains sca st scan code sent ************************************	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng keystroke to the Personal on codes to be sent to the FC to the Personal Computer
900 994 A A7 78F	58 C3 88 35 C6 55 FF EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 E 003 35 15 003 57 57 20 20 20 20 20 20 20 20 20 20 20 20 20	6 R		P R R M J SND_TO_P Process Funct: Input Output STT_TO_P( Mi CC J)	OP ET C:: OV OV MP c ENDP dure Ha ion: Se Co c t: Olde ****** C PROC OV MP Z	AX DI,BUFFER_PC_LAS BYTE PTRIDI),OFF SEND_TO_FC_2 me: XHIT_TO_PC and the next pendi mputer R_PC contains sca ast scan code sent ************************************	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng keystroke to the Personal on codes to be sent to the FC to the Personal Computer ; SEE IF THERE ARE ANY KEY STROKES ; THE BUFFER NO, JUST RETURN TO CALLER
999 9984 45 7 7 78F 31	58 C3 BB 3E 003 C6 05 FF EB E7 EB E7 BB 1E 003 38 1E 003 74 19 BA 27	6 R	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	P R R M J SND_TO_P Proces Funct Input Outpur STT_TO_P4 KIT_TO_P4 KIT_TO_P4 M	OP ET C: OV MP C ENDP dure Ha ion: Se C C EUFFE t: olde ******* C PROC OV MP Z OV	AX DI,BUFFER_PC_LAST BYTE PTRIDI),OFF SEND_TO_FC_2 The second	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng keystroke to the Personal an codes to be sent to the FC to the Personal Computer to the Personal Computer
99 994 A A A AAAA BBBB	58 C3 88 85 66 85 85 85 85 85 85 85 85 85 85 85 85 85	6 R 8 R		P R R H H J J ND_TO_P Process Funct Input Output STT_TO_PA STT_TO_PA CC J J Mt L L	OP ET C: OV MP C ENDP dure Ha ion: Se C EUFFE t: olde ******* C FROC OV MP Z Z OV EA	AX DI,BUFFER_PC_LAST BYTE PTRIDI),OFF SEND_TO_FC_2 The next pendi me: XHIT_TO_PC and the next pendi mputer R_PC contains sca ast scan code sent HEAR BX,BUFFER_PC_HEAD BX,BUFFER_PC_TAIL XHIT_TO_PC_3 ; AH,IBX1 ; BX,BUFFER_PC_END	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP 
537 9 553 3 5 5 7 5 5 5 5 7 5 5 5 5 7 5 5 5 5 7 5 5 5 5	58 C3 BB 3E 003 C6 05 FF EB E7 EB E7 BB 1E 003 38 1E 003 74 19 BA 27 43 BD 16 003 38 DA	6 R 8 R	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	P R R M J SND_TO_P Process Funct Input Output STT_TO_P4 KIT_TO_P4 CI J J CI Input	OP ET C:: OV MP C ENDP dure Ha dure Ha ion: Se Co : EUFFE t: olde ******* C FROC OV MP Z OV MP	AX DI, BUFFER_PC_LAS' BYTE PTRIDI), OFF SEND_TO_FC_2 SEND_TO_FC_2 and the next pendi imputer R_PC contains sca st scan code sent ************************************	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP 
50 5954 A A 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 5	58 C3 88 66 57 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 EB E7 E E E7 E E7 E E7 E E7 E E7 E E7 E E7 E7	6 R 8 R 6 R		P R R H H J IND_TO_P Process Funct Input Output Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input Input	OP ET C:: OV MP C ENDP dure Ha ion: Se C C EUFFE t: olde ******* C PROC OV MP Z OV NC EA MP NE	AX DI,BUFFER_PC_LAST BYTE PTRIDI),OFF SEND_TO_FC_2 The next pendi me: XHIT_TO_PC and the next pendi mputer R_PC contains sca ast scan code sent HEAR BX,BUFFER_PC_HEAD BX,BUFFER_PC_TAIL XHIT_TO_PC_3 ; AH,IBX1 ; BX,BUFFER_PC_END	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP 
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	58 C3 BB 3E 003 C6 05 FF EB E7 EB E7 BB 1E 003 38 1E 003 74 19 BA 27 43 BD A 75 04 ED 1E 003	6 R 8 R 6 R	······································	P R R J SND_TO_P SND_TO_P Process Funct: Input Outpu: UIT_TO_P( Ci J) UIT_TO_P( II L)	OP ET C:: OV MP C ENDP dure Ha dure Ha ion: Se Co t: BUFFE t: olde t: FROC OV MP Z OV MP Z OV EA EA C_1:	AX DI, BUFFER_PC_LAS' BYTE PTRIDI), OFF SEND_TO_FC_2 A A A A A A A A A A A A A	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF. HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng keystroke to the Personal on codes to be sent to the FC to the Personal Computer ; SEE IF THERE ARE ANY KEY STROKES ; THE BUFFER NO, JUST RETURN TO CALLER PICK UP NEXT SCAN CODE IN BUFFER
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	58 C3 BB 3E 003 C6 05 FF EB E7 EB E7 BB 1E 003 36 1E 003 74 19 C0 16 003 36 D 16 003	6 R 8 R 6 R	, , , , , , , , , , , , , , , , , , ,	P R R J SND_TO_P SND_TO_P Process Funct: Input Outpu: UIT_TO_P( Ci J) UIT_TO_P( II L)	OP ET C:: OV MP C ENDP dure Ha dure Ha ion: Se Co t: BUFFE t: olde t: FROC OV MP Z OV MP Z OV EA EA C_1:	AX DI, BUFFER_PC_LAST BYTE PTRIDI), OFF SEND_TO_FC_2 The perfection of the perfect of the	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP 
3900 3390 EE2 3392 E2 3333 A 7 34 7 7 7 8 8 1 3 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	58 C3 BB 3E 003 C6 05 FF EB E7 EB E7 B 1E 003 38 1E 003 74 19 6A 27 6B 16 003 38 DA 75 04 ED 16 003 89 16 003 80 16 000 80 16 0000 80 16 0000 80 16 0000 80 16 0000000000000000000000000000000000	6 R 8 R 6 R		P R R J J ND_TO_P Process Funct: Input Output Output UT_TO_P( N( II II II II II II II M( M( M( M( M( M( M( M( M( M( M( M( M(	OP ET C:: OV MP c ENDP dure Ha dure Ha ion: Se Co c EUFFE t: olde t: eUFFE t: olde t: PROC OV MP Z OV MP Z OV NC EA EA C_I: OV	AX DI, BUFFER_PC_LAS' BYTE PTRIDI), OFF SEND_TO_FC_2 SEND_TO_FC_2 The next pendi mputer R_PC contains sca st scan code sent SENDFER_PC_HEAD BX, BUFFER_PC_TAIL XHIT_TO_PC_3 AH, IBX ] BX, BUFFER_PC_END BX, BUFFER_PC_I BX, BUFFER_PC_I BX, BUFFER_PC_I BX, BUFFER_PC_I BX, BUFFER_PC_I BX, BUFFER_PC_I BX, BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BX, BUFFER_PC_I BX, BY AND	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng keystroke to the Personal on codes to be sent to the FC to the Personal Computer ; SEE IF THERE ARE ANY KEY STROKES ; THE BUFFER NO, JUST RETURN TO CALLER PICK UP NEXT SCAN CODE TO BE SENT POINT TO NEXT SCAN CODE IN BUFFER SEND THE SCAN CODE TO THE PERSONAL
550 5555 55 7 5555 5555 55 7 7 7 8 5 1 3 4 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	58         35         003           68         35         FF           EB         E7           EB         12           84         27           43         16         003           38         14         003           39         16         003           39         16         003           39         16         003           39         16         003           39         16         003           39         04         202           60         16         003           39         04         202           60         16         003           39         04         202           60         16         003           39         16         003           30         04         202           60         16         003           38         04         202	6 R 8 R 6 R	, , , , , , , , , , , , , , , , , , ,	P R R M J SND_TO_P H Proces Funct Input Output SUT_TO_P( SUT_TO_P( M CT IN II II II II II II II II II II II II	OP ET C:: OV MP dure Ha ion: Se Co : BUFFE t: olde ******* C PROC OV MP Z OV EA EA MP NE EA C_1: OV	AX DI, BUFFER_PC_LAS' BYTE PTRIDI), OFF SEND_TO_FC_2 SEND_TO_FC_2 and the next pendi mputer R_PC contains sca- sst scan code sent ************************************	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP 
570 EEE 5 7 7 BF 1348AC00 47	58         35         003           68         35         FF           EB         E7           EB         12           84         27           43         16         003           38         14         003           39         16         003           39         16         003           39         16         003           39         16         003           39         16         003           39         04         202           60         16         003           39         04         202           60         16         003           39         04         202           60         16         003           39         16         003           30         04         202           60         16         003           38         04         202	6 R 8 R 6 R		P R R M J SND_TO_P H Proces Funct Input Output SUT_TO_P( SUT_TO_P( M CT IN II II II II II II II II II II II II	OP ET C:: OV MP dure Ha ion: Se Co t: BUFFE t: olde ******* C FROC OV MP Z OV EA EA MP NE EA C_I: OV	AX DI, BUFFER_PC_LAS' BYTE PTRIDI), OFF SEND_TO_FC_2 SEND_TO_FC_2 The next pendi mputer R_PC contains sca st scan code sent SENDFER_PC_HEAD BX, BUFFER_PC_TAIL XHIT_TO_PC_3 AH, IBX ] BX, BUFFER_PC_END BX, BUFFER_PC_I BX, BUFFER_PC_I BX, BUFFER_PC_I BX, BUFFER_PC_I BX, BUFFER_PC_I BX, BUFFER_PC_I BX, BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BUFFER_PC_I BX, BUFFER_PC_I BX, BY AND	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF. HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng keystroke to the Personal on codes to be sent to the FC to the Personal Computer ; SEE IF THERE ARE ANY KEY STROKES ; THE BUFFER NO, JUST RETURN TO CALLER PICK UP NEXT SCAN CODE TO BE SENT POINT TO NEXT SCAN CODE IN BUFFER SEND THE SCAN CODE TO THE PERSONAL
1270 1212 12 13 7 1270 1212 13 7 1348AC000 479 A	58         88       35       003         68       35       FF         EB       E7         B1       E       003         38       12       003         38       12       003         84       27       16       003         38       16       003       16         9       12       002       16         84       27       23       24         9       12       002       25         84       04       25       24         85       24       24       25         84       0325       25       24         85       24       24       24	6 R 8 R 6 R		P R R VERRUN_P M J J END_TO_P Proces Funct: Input Outpu: INT_TO_P( II II II II II II II II II I	OP ET C:: OV MP dure Ha ion: Se Co : BUFFE t: olde ******* C FROC OV MP Z OV EA MP NE EA C_1: OV OV OV UT C_3:	AX DI, BUFFER_PC_LAS' BYTE PTRIDI), OFF SEHD_TO_FC_2 SEHD_TO_FC_2 and the next pendi mputer R_PC contains sca- st scan code sent ************************************	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF. HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng heystroke to the Personal an codes to be sent to the FC to the Personal Computer ; SEE IF THERE ARE ANY KEY STROKES ; THE BUFFER NO, JUST RETURN TO CALLER PICK UP NEXT SCAN CODE TO BE SENT POINT TO NEXT SCAN CODE IN BUFFER SEND THE SCAN CODE TO THE PERSONAL COMFUTER
99 994 A A A AAAA BBBBBBBCC CCC C	58         35         003           68         35         FF           EB         E7           EB         12           84         27           43         16         003           38         14         003           39         16         003           39         16         003           39         16         003           39         16         003           39         16         003           39         04         202           60         16         003           39         04         202           60         16         003           39         04         202           60         16         003           39         16         003           30         04         202           60         16         003           38         04         202	6 R 8 R 6 R		P R R VERRUN_P M J J END_TO_P Proces Funct: Input Outpu: INT_TO_P( II II II II II II II II II I	OP ET C:: OV MP c ENDP dure Ha dure Ha ion: Se Co c EUFFE t: olde t: olde t: FUFFE t: olde C PROC OV MP Z OV EA MP NE EA C OV OV UT	AX DI, BUFFER_PC_LAS' BYTE PTRIDI), OFF SEHD_TO_FC_2 SEHD_TO_FC_2 and the next pendi mputer R_PC contains sca- st scan code sent ************************************	T ; PICK UP ADDRESS OF LAST EYTE US H ; REPLACE LAST KEY STROKE IN BUFF. HITH OVERRUN SCAN CODE ; SEND KEY STROKES TO PERSONAL COMP ng keystroke to the Personal on codes to be sent to the FC to the Personal Computer ; SEE IF THERE ARE ANY KEY STROKES ; THE BUFFER NO, JUST RETURN TO CALLER PICK UP NEXT SCAN CODE TO BE SENT POINT TO NEXT SCAN CODE IN BUFFER SEND THE SCAN CODE TO THE PERSONAL

.

1169		c
1170		└ С С ;#Ж#+≒
1172		C ;* C ;* Procedure Name: PROC_PC_PORT_AVAIL *
1174		C 3#
1176		C 34 interrupt **
1177		C ;# C ;# Input: Hone
1179 1180		C ;* C ]* Output: The next pending keystroke is sent to the Personal
1181 1182		C }* Computer C ;*
1183 1184		C ;************************************
1185	03CB	C PROC_PC_PORT_AVAIL PROC NEAR
1187 1188 1189	03CB BA 03E4 03CE B0 00 03D0 EE	C MOV DX,COMMAND_PORT; ACKNOWLEDGE THE INTERRUPT C MOV AL,SERIAL_INT_OFF C OUT DX,AL
1190 1191	0301 F6 06 0002 R 20	C TEST LOGICAL_STATE, L_PAUSE ; ARE WE IN PAUSE STATE?
1192 1193 1194 1195	0306 74 06 0308 80 0E 0002 R 10	C JZ PC_PORT_AVAIL_1; NO, SEND PENDING KEY STROKE C OR LOGICAL_STATE,L_PC_PORT_AVAIL; YES, INDICATE THAT THE C INTERRUPT OCCURRED SO IT CAN BE C ; PROCESSED WHEN WE EXIT PAUSE STATE
1196	0300 C3	C RET ; PROCESS NEXT PENDING INTERRUPT
1197	03DE	C C PC_PORT_AVAIL_1:
1199 1200 1201	030E E8 03A7 R 03E1 C3	C CALL XMIT_TO_PC ; SEND NEXT PENDING KEY STROKE TO THE C ; C RET ; PROCESS NEXT PENDING INTERRUPT
1202 1203	03E2	C C proc_pc_pcrt_avail endp
1204 1205		C C ;+++++++++++++++++++++++++++++++++++
1206 1207		C ;* C ;* Procedure Name: PROC_3278_PORT_AVAIL
1208 1209		C ;* C ;* Function: To process a 3278/3279 keyboard port available interrupt
1210 1211		C ;* C ;* Input: None
1212 1213		C ;* C ;* Dutput: The next pending keystroke is sent to the 3278/3279
1214 1215		C ;**
1216	03E2	C C C PROC_3278_PORT_AVAIL PROC NEAR
1218	D3E2 BA 03E3	C
1220	03E5 B0 04	C MOV AL,RESET_INTE_B
1221	03E7 EE	C DUT DX,AL C C CALL XHIT TO 3278 ; SEND NEXT PENDING KEY STROKE TO THE
1223	03E8 E8 0330 R	C ; 3278/3279
1225 1226	03E8 C3	C RET ; PROCESS NEXT PENDING INTERRUPT C
1227 1228	DJEC	C PROC_3276_PORT_AVAIL ENDP C
1229 1230	03EC	PRCC_KEYBOARD_DATA ENDP
1231		C INCLUDE NDSPKBD3.ASM C SUBTTL Process a keyboard software function
1233		<b>C</b>
1234 1235		C
1236		C ;* C ;* Procedure Name: SEND_ENCODED_KEYSTROKES_TO_3278 *
1238		C 3* * C 3* Function: To send a series of encoded keystrokes to the 3278/3279 *
1240 1241		C ;* at a rate not exceeding 10 keystrokes per second * C ;*
1242 1243		C;* Input: * C;*
1244 1245		C ;* EX:DX = address of encoded keystrokes C ;* CX = number of encoded keystrokes *
1246		C 3* *
1248		C ;* *
1249 1250 1251 1252		C ;* CF = 0 if all encoded keystroke values are valid C ;* 1 if any of the encoded keystroke values are invalid for C ;* this keyboard *
1253 1254		C ;* C ;* AL = 4 if any of the encoded keystroke values are invalid for * C ;* this keyboard *
1255 1256		* C ;***********************************
1257 1258	03EC	C SEND_ENCODED_KEYSTROKES_TO_3278 PROC NEAR
1259 1260	03EC 1E 03ED 1E	C PUSH DS ; SAVE DATA POINTER C PUSH DS ; SHAP SEGMENT REGISTERS
1261	03EE 07	C POP ES

## 

		03			00
1262 1263	03EF	8E D8	c c	HOV	DS.8X DS:NOTHING.ES:EIOSCATA
1264 1265	03F1	8B F2	c c	MOV	SI,DX ; PICK UP ADDRESS OF ENCODED KEYSTROKES
1266 1267	03F3	FC	c c	сцэ	; SET UP FOR FOPWARD SCAN OF DATA
1268 1269			C ;******	*****	* 《水水法监察就在安安安全就不会想到承担在安全大学大学大学大学大学大学大学大学大学大学大学大学大学大学大学大学大学大学大学
1270 1271			C ;× C ;¥	SCAN T	THE DATA TO INSURE THAT THE KEYSTROKES ARE ALL VALID *
1272 1273			C ;******	*****	1 寒水浅湿雾岩雾水在黄水在水水在水在 11 家族名法名法家名法女教育家家大学家女女女女女女女女女女女女女女女女女女女女女女女女女女
1274 1275	03F4 03F5	51 08 C9	c c	PUSH OR	CX ; SAVE KEYSTROKE COUNTER CX,CX ; IS THE COUNTER ZERO? SCAN_KEYS_2 ; YES, INVALID INPUT
1276 1277		74 10	c c	JZ	SCAN_KEYS_2 ; YES, INVALID INPUT
1278	03F9 03F9	32 E4	C SCAN_KE	YS_1: XOR	AH,AH ; PICK UP NEXT KEYSTROKE ENCODING
1280	03FB		с с	LODSB	
1282		D1 E0 88 F8	c c	SAL MOV	AX,1 ; PICK UP KEYSTRCKE INFORMATION DI,AX
1284		26: 88 85 0492 P		MOV	AX, KEYSTROKE_XLATIDI]
1286 1287		0B C0 75 06	c c	OR JNZ	AX,AX ; IS THIS A VALID TABLE ENTRY? SCAN_KEYS_3 ; YES, CONTINUE
1288	0409		C C SCAN_KE		
1290		BD 04		MOV POP	AL,RC_INVALID_KEY; NO, INDICATE INVALID KEYSTROKE
1292		1F	с с с	POP	CX ; RETURN TO CALLER INDICATING EPRCR DS ; IN KEYSTROKE ENCODING
1294	040E		с с	RET	
1296	040F	E2 E8	C SCAN_KE	YS_3: LOOP	SCAN_KEYS_1 ; SCAN ALL SPECIFIED KEYSTROKES
1298	0401		C		
1300			C ;* C ;*		ALLY DISABLE THE 3276/3279 KEYBOARD AND RESET ALL
1302			C ;+ C ;+		STATES
1304					THE PHYSICAL KEYBOARD MAY NOW BE OUT OF SYNCH WITH * 274/3276 CONTROL UNIT SINCE THE USER COULD STILL BE *
1366 1307				ROLDIN	IG A SHIFT MEY, HOWEVER, THIS IS THE ONLY WAY THE " EYSTROME FUNCTION CAN NORK.
1308			C ;*		·····································
1310	0411	B9 0002	C C	HOV	CX.2 ; FORCE A JOD HS DELAY BETWEEN THESE
1312		26: 80 0E 0002 R 08	C	OR	LOGICAL_STATE,L_KED_DISABLED ; MARK THE KEYEOARD AS
1314 1315			c c	Un	; DISABLED
1316 1317		26: F6 06 0000 R 20 74 0B		TEST JZ	KBD_STATE,RIGHT_SHIFT ; IS THE PIGHT SHIFT KEY FRESSED? RESET KED 3 ; NO, CONTINUE
1318 1319	0422	B0 CE	c c	mov	AL,BREAK_BIT OR HDS_R_SHIFT_KEY ; YES, IHDICATE THAT THE
1320 1321		E8 052A R 26: 80 26 0000 R DF	c c	CALL	SEND_KEY_TO_3278 ; SHIFT KEY WAS RELEASED KBD_STATE,NOT RIGHT_SHIFT
1322 1323	042D		C C RESET_K		
1324 1325		26: F6 06 0000 R 80 74 DB	с — с	TEST JZ	KBD_STATE,SHIFT_LOCK ; IS THE SHIFT LOCK KEY DEPRESSED? RESET_KBD_4 ; NO, CONTINUE
1326 1327		B0 CC		nov	AL,BREAK_BIT OR NDS_LOCK_KEY ; YES, INDICATE SHIFT LOCK
1328 1329	0437 043a	E8 052A R 26: 80 26 0000 R 7F	c c	CALL AND	SEHD_KEY_TO_3278 ; KEY WAS RELEASED KBD_STATE,NOT SHIFT_LOCK
1330	0440		C C RESET_K		
1332		26: F6 06 0000 R 08	C C	TEST	KBD_STATE,LOCK_STATE ; IS THE KEYBOARD IN SHIFT LOCK OR ; HUM LOCK STATE?
1334 1335		74 IC	c c	JZ	RESET_KBD_5 ; NO, CONTINUE
1336 1337	0448	B0 4C	C C	HOV	AL,NDS_LOCK_KEY ; YES, MUST RESET SHIFT LOCK OR NUM ; LOCK STATE
1338 1339 1340			C ;		IF THIS IS A DATA ENTRY KEYEDARD, NUM LOCK STATE IS Reset by pressing and releasing the Num Lock Key
1341					IF THIS IS A NON-DATA ENTRY KEYBOARD, SHIFT LOCK STATE
1342 1343 1344			C ; C ;		IS RESET BY PRESSING AND RELEASING EITHEP SHIFT KEY SINCE WE DO NOT KNOW THE KEYEOARD TYPE HERE, WE WILL
1344 1345 1346	<b>866</b> 4	B0 4C	C ; C		DO BOTH
1347	044C	B0 4C E8 052A I! B0 CC	с с	MOV CALL	AL,NDS_LOCK_KEY; INDICATE LOCK KEY IS DEPRESSED SEND KEY TO 3278 AL DEPK BIT OF DE DE LOCK KEY; INDICATE LOCK KEY IS
1348 1349	0451	E8 052A R	с с	MOV CALL	AL, BREAK_BIT OR NDS_LOCK_KEY ; INDICATE LOCK KEY IS SEND_KEY_TO_3278 ; RELEASED
1350 1351	0456	E8 052A R	C C	HOV CALL	AL,NDS_L_SHIFT_KEY ; INDICATE SHIFT KEY IS DEPRESSED SEND_KEY_TO_3278
1352 1353			с с	HOV CALL	AL,BREAK_BIT OR NDS_L_SHIFT_KEY ; INDICATE SHIFT KEY SEND_KEY_TO_3278 ; IS RELEASED
1354			c	AND	KBD_STATE,NOT LOCK_STATE

67

68

-

1356	0464		C C RESET_I	KBD_5:		
1357			c -			THE TRANSMENT OF T
1358				****	*****	· 张 王 王 王 王 王 王 王 王 王 王 王 王 王 王 王 王 王 王
1359 1360			C ;* C ;*	SEND TH	E SPECIFIED KEY	STROKES TO THE 3278/3279 DISPLAY
1361			C ;*	STATION		*
1362			C ;#			*
1363				****	****************	我就去去才有我就要要就没不知道我就没有我们就要没有要要的你能要要没有我们不能要能
1364 1365	8444	8B F2	C C	ноу	SI,DX	; POINT TO START OF KEYSTROKE ENCODINGS
1366	0404		c	1.01	01,04	
1367	0466		C SEND_K	EYS_1:		
1368		32 E4	С	XOR	AH, AH	; PICK UP NEXT KEYSTROKE ENCODING
1369	0468	AC	C C	LODSB		
1370 1371	0469	D1 E0	c	SAL	AX,1	FICK UP KEYSTROKE DATA FROM TABLE
1372	046B	6B F8	č	HOV	DI,AX	
1373	046D	26: 88 90 0492 R	c	HOV	BX,KEYSTROKE_X	(LATIDI)
1374		0/- 5/ 0/ 0000 D 01	с с	TEST	KED STATE FOR	ED_ALT_STATE & ARE WE IN THE MIDDLE OF A
1375 1376	0472	26: F6 06 0000 R 01	c	1631	KDD_DIAILII OKC	SEQUENCE OF ALT KEYSTROKES?
1377	0478	74 07	č	JZ	TEST_CASE	; NO, CONTINUE
1378	047A	F6 C3 02	C	TEST	BL,BA	; YES, IS THIS THE END OF THE SEQUENCE?
1379	0470	75 02	c	. SIZ	TEST_CASE	; YES, CONTINUE ; NO, FORCE THIS KEYSTROKE TO ALT CASE
1380	047F	B3 20	с с	U_7	EL,A	I HOT FORCE THIS RETURNE TO RET SHOE
1381 1382	0481		C TEST_C	451:		
1383		F6 C3 80	c T	TEST	BL,U	IS UPPER CASE REQUIRED?
1384	0484	75 15	C	382	SEND_UC	; YES, MUST BE IN UPPER CASE MODE
1385		26: F6 06 0000 R 40	C C	TEST	SEND CASE OF	SHIFT ; NO, HUST BE IN LOWER CASE MODE ; SHIFT KEY IS NOT FRESSED - CONTINUE
1386 1387		74 20 80 CD	C C	uz t:ov	AL BREAK BIT O	R NDS L SHIFT KEY ; SHIFT KET IS PRESSED
1388	0485		c	CALL	SEND_KEY_TD_32	78 3 - INDICATE SHIFT KEY IS RELEASED
1389	0493	26: 80 26 0000 R BF	С	AI ID	KED STATE NOT	LEFT SHIFT
1390	0499	EB 13 .	C C	JMP	SHORT SEND_CAS	E_OK ; NOW IN LOWER CASE - CONTINUE
1391 1392	049B		C C SEND_U	C:		
1393	049B	26: F6 06 0000 R 40	C 00110_0	TEST	KBD_STATE.LEFT	SHIFT ; MUST BE IN UPPER CASE MODE
1394		75 08	C	JHZ	SEND CASE OK	: SHIFT KEY IS PRESSED - CUNIINOC
1395		B0 4D	c	MOV	AL, NDS_L_SHIFT	KEY ; SHIFT KEY IS RELEASED - INDICATE
1396		E8 052A R	с с	CALL OR	KBD_STATE, LEFT	78 ; SHIFT KEY IS PRESSED
1397 1398	UNAO	26: 80 OE 0000 R 40	c	UR		_0
1399	04AE			ASE_OK:		
1400			С			
1401		F6 C3 20	c	TEST	BLAA	; IS ALT CASE REQUIRED? ; Yes, must be in alt mode
1402 1403	04B1 04B3	75 15 26: F6 06 0000 R 02	C C	JNZ TEST	KED STATE ALT	STATE : HUST NOT BE IN ALT HOUL
1404		74 2A	č	JZ	SEND ALT OK	: ALT KEY IS NOT PRESSED - CONTINUE
1405	04BB	B0 CF	C	MOV	AL, BREAK_BIT O	R NDS_ALT_KEY ; ALT KEY IS PRESSED
1406	04BD	E8 052A R	C	CALL	SEND_KEY_TO_32 KBD_STATE,NOT	78 ; - INDICATE ALT KEY IS RELEASED
1407 1408	0400	26: 80 26 0000 R FD EB 1D	с с	AND JMP	SHORT SEND ALT	COK ; NOW IN NON-ALT CASE - CONTINUE
1409	0400	<i>co</i> 10	č			
1410	0408			LT_MODE:		
1411	04C8	OA FF	C	OR	внъвн	; IS THIS THE START OF A SEQUENCE OF ; ALT MODE KEYSTROKES?
1412		75.0/	c	1.17	SEND ANT 1	; ALI MODE REISIRORES: ; NO, CONTINUE
1413		75 06 26: 80 0E 0000 R 01	с с	JNZ OP	SEND_ALT_1 KBD STATE,FORD	CED_ALT_STATE ; YES, INDICATE ALT SEQUENCE
1414	0400		c	-		
1416	0402		C SEND_/			THE SHOP OF THE ALT MODE
1417		26: F6 06 0000 R 02	C	TEST	KBD_STATE,ALT_	STATE ; HUST BE IN ALT MODE ; ALT KEY IS PRESSED - CONTINUE
1418		75 OB BD 4F	с с	JNZ MOV	SEND_ALT_OK	Y ; ALT KEY IS RELEASED - INDICATE
1419 1420		80 47 E8 0524 R	C	CALL	SEND_KEY TO 32	276 ; ALT KEY IS PRESSED
1421		26: 80 DE DODO R 02	С	OR	KBD_STATE, ALT_	
1422			C			
1423	04E5	F( F] A2		ALT_OK: TEST	BL.BA	; IS EREAK/ALT INDICATED?
1424		F6 C3 02 74 06	c c	JZ	BL,BA SEND_B_ALT_DK	: NO, CONTINUE
1425 1426		26: 80 26 0000 R FE	c	AND	KED_STATE, NOT	FORCED_ALT_STATE ; YES, INDICATE END DF
1427			C	-	-	; ALT SEQUENCE
1428			C			
1429	04F0	50	C SEND_1	B_ALT_OK: POP	cx	; RESTORE KEYSTROKE COUNTER
1430	04F0 D4F1	59 84 C7	C C	HOP	AL,BH	; PICK UP 3278/3279 KEY NUMBER
1431 1432		DA CD	c	07	AL, AL	; IS THERE A KEYSTROKE TO SEND?
1433		74 DA	C	JZ	SEND_KEY5_2	; NO, JUST ALT STATE CHANGE
1434			c		DY FOR FOR	XLAT-1 ; CONVERT KEY NUMBER TO 3278/3279
1435		26: 6D IE ODBB R	C C	LEA XLAT	SCAN CODE YIA	T ; SCAN CODE
1436	04FC	26: D7	C C	AL51		
1437 1438	04FF	E8 052A R	č	CALL	SEND_KEY_TO_3	278 ; SEND THE SCAN CODE TO THE
1439			С			3278/3279 DISPLAY STATION
1440			C			
1441	0501	( <b>a</b>		KEYS_2:	<b>CY</b>	; DECREMENT KEYSTROKE COUNTER
1442	0501		c c	DEC JCXZ	CX SEND_KEYS_3	; ALL KEYSTROKES HAVE BEEN PROCESSED
	0502	E3 04		PUSH	CX	SAVE KEYSTROKE COUNTER
1443		51	C			
1443	0504		с с	JHP	SEND_KEYS_1	; LOOP UNTIL ALL KEYSTROKES HAVE BEEN
1443	0504	51 E9 0466 R				I LOOP UNTIL ALL KEYSTROKES HAVE BEEN SENT TO THE 3278/3279 DISPLAY

	~~	•	,641,262	~~
0508	69	C SEID_KEYS_3:		70
	26: 80 26 0000 R FE	с <b>ј ј</b> ј	KBD_STATE, NOT	FORCED_ALT_STATE ; INDICATE END OF
050E	26: F6 06 0000 R 02	C C C TEDI	NED STATE AIT 5	; SEQUENCE OF ALT KEYSTROKES STATE ; WAS THE ALT KEYSTROKE SENT?
514		C J2	CEND_KEYS_4	; NO, CONTINUE
518	BO CF E8 052A R 26: 80 26 0000 R FD	C HOV C CALL C AND C	AL,BREAK_BIT OF SEND_KEY_TO_323 KED_STATE,NOT A	<pre>NDS_ALT_KEY ; YES, INDICATE THAT THE B ; ALT KEY WAS RELEASED NLT_STATE</pre>
0521 0521	26: 80 26 0002 R F7	C SEND_KEYS_4: C AND C	LOGICAL_STATE.	NOT L_KBD_DISABLED ; ENABLE THE PHYSICA ; KEYBOARD
0527 0528 0529	FB	C POP C CLC C RET C	DS .	; RESTORE DATA SEGMENT REGISTER ; Indicate Keystrokes Here Valid ; Return to Caller
			***	*****
		C ;# C ;* Procedure C ;*	Name: SEND_KEY_TO	_3278
		C # Function:		ske to the 3278/3279 and wait 100 ms : keystrokes to be sent
		C ;# C ;# AL = 327		ican code to be sent
		C ;*	•	waiting to be sent
		C 3* Output: No C 3* C 3*		*发家家爱爱家爱望亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲亲
052A		C C SEND_KEY_TO_32		
052A	53	C PUSH	BX	; SAVE REGISTERS
052B 052C	51 52	C PUSH C PUSH C	CX DX	
052D 052D 0532 0537	26: 88 1E 0020 R 26: 38 1E 0022 R 75 F4	C SEND_KEY_TO_32 C MOV C CMP C JNE	BX.BUFFER 3278	HEAD ; WAIT UNTIL ALL STACKED KEYSTRO TAIL ; HAVE BEEN SENT TO THE DISPLA 8_1
0539 053A	50 Ba 03E2	C FUSH C FUSH C MOV C SEND_KEY_TO_32	AX DX,KB_CTL	; SAVE SCAN CODE TO BE SENT ; WAIT FOR THE 3278/3279 OUTPUT PORT
	<b>FG</b>			; TO BECOME AVAILABLE
053D 053D 053E 0540	EC A8 02 74 FB	C IN C - TEST	AL,DX Al,D2H	-
0530 053E 0540		C IN C - TEST C JZ C POP	AL,DX	-
053D 053E 0540 0542 0543	A8 02 74 FB	C IN C - TEST C JZ C POP C C MOV C OUT	AL,DX AL,D2H SEND_KEY_TO_327	- 78_2
053D 053E 0540 0542 0543 0543 0546	A8 02 74 FB 58 BA 03E1 EE 83 F9 01	C IN C TEST C JZ C POP C MOV C OUT C CMP	AL, DX AL, D2H SEND_KEY_TO_327 AX DX, NDS_KB_OUT DX, AL CX, 1	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE?
0530 053E 0540 0542 0543 0546 0546 0544 0544	A8 02 74 FB 58 BA 03E1 EE	C IN C - TEST C JZ C POP C POP C MOV C OUT C	AL,DX AL,D2H SEND_KEY_TO_327 AX DX,NDS_KB_OUT DX,AL CX,1 SEND_KEY_TO_327	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY
0530 053E 0540 0542 0543 0546 0546 0544 0544	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A	C IN C TEST C JZ C POP C MOV C OUT C CMP C JE C JE C MOV C LOOP C SEND_KEY_T0_32	AL.02X AL.02H SEND_KEY_TO_327 AX DX,NDS_KB_OUT DX,AL CX,1 SEND_KEY_TO_327 CX,DELAY_100_HS \$	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO NOT DELAY
0530 053E 0540 0542 0543 0546 0546 0546 0547 0544 0547 0544 0547 0551	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE	C IN C JEST C JZ C POP C MOV C OUT C OUT C CMP C JE C MOV C LOOP C SEND_KEY_TO_32 C POP	AL.02X AL.02H SEND_KEY_TO_327 AX DX,HDS_KB_OUT DX,AL CX,1 SEND_KEY_TO_327 CX,DELAY_100_HS \$ 78_3: DX	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO NOT DELAY
0530 0532 0540 0542 0542 0546 0546 0546 0546 0546 0547 0546 0547 0551 0551 0551	A8 02 74 FB 58 BA 03E1 EE A3 F9 01 74 05 B9 681A E2 FE 54 59	C IN C TEST C JZ C PCP C MOV C OUT C OUT C CHP C CHP C JE C MOV C LODP C SEND_KEY_TO_32 C POP C POP C POP	AL, DX AL, D2H SEND_KEY_TO_327 AX DX, NDS_KB_OUT DX, AL CX, 1 SEND_KEY_TO_327 CX, DELAY_100_HS \$ 78_3:	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO NOT DELAY ; WAIT FOR 100 MILLISECONDS
0530 053E 0540 0542 0543 0546 0546 0546 0547 0544 0546 0547 0544 0554 0551 0551 0551 0552 0553	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE 58 55	C IN C TEST C JZ C POP C MOV C OUT C OUT C CMP C JE C MOV C LOOP C JE C POP C POP C POP C POP C POP C RET	AL.DX AL.D2H SEND_KEY_TO_327 AX DX,NDS_KB_OUT DX,AL CX,1 SEND_KEY_TO_327 CX,DELAY_100_HS \$ 78_3: DX CX	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO NOT DELAY ; WAIT FOR 100 MILLISECONDS
0530 0532 0540 0542 0544 0544 0544 0544 0544 0551 0551 0551	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE 58 55	C IN C TEST C JZ C PGP C MOV C OUT C OUT C CMP C JE C MOV C LOOP C LOOP C SEND_KEY_TO_32 C POP C POP C POP C POP C RET C SEND_KEY_TO_32	AL,DX AL,02H SEND_KEY_TO_327 AX DX,NDS_KB_OUT DX,AL CX,1 SEND_KEY_TO_327 CX,DELAY_100_HS \$ 78_3: DX CX BX	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO HOT DELAY 5 ; WAIT FOR 100 MILLISECONDS ; RESTORE REGISTERS
0530 0532 0540 0542 0544 0544 0544 0544 0544 0551 0551 0551	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE 58 55	C IN C TEST C JZ C POP C MOV C OUT C CMP C JE C MOV C JE C MOV C JE C POP C POP C POP C POP C POP C POP C RET C SEND_KEY_TO_32 C RET C SEND_KEY_TO_32 C ASSUME	AL,DX AL,02H SEND_KEY_TO_327 AX DX,NDS_KB_OUT DX,AL CX,1 SEND_KEY_TO_327 CX,DELAY_100_HS \$ 78_3: DX CX BX	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO NOT DELAY ; WAIT FOR 100 MILLISECONDS ; RESTORE REGISTERS ; RETURN TO CALLER
0530 0532 0542 0544 0544 0544 0544 0551 0551 0551 0555 0555	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE 58 55	C IN C TEST C JZ C POP C POP C MOV C OUT C CMP C JE C MOV C LOOP C JE C POP C POP C POP C POP C POP C POP C RET C SEND_KEY_TO_32 C RET C SEND_KEY_TO_32 C SEND_KEY_TO_32	AL, DX AL, D2H SEND_KEY_TO_327 AX DX, NDS_KB_OUT DX, AL CX, 1 SEND_KEY_TO_327 CX, DELAY_100_HS \$ 78_3: DX CX BX 78_ENDP	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO HOT DELAY 5 ; WAIT FOR 100 HILLISECONDS ; RESTORE REGISTERS ; RETURN TO CALLER
0530 0532 0542 0544 0544 0544 0544 0551 0551 0551 0555 0555	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE 58 55	C IN C TEST C JZ C POP C MOV C OUT C OUT C CMP C JE C MOV C LOOP C LOOP C SEND_KEY_TO_32 C POP C POP C POP C POP C POP C SEND_KEY_TO_32 C	AL.DX AL.DX AL.D2H SEND_KEY_TO_327 AX DX.NDS_KB_OUT DX.AL CX.1 SEND_KEY_TO_327 CX.DELAY_100_HS 5 78_3: DX CX DX CX BX 78_ENUP DS:BIDSDATA.ES: EYSTROKES_TO_3276	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO HOT DELAY 5 ; WAIT FOR 100 HILLISECONDS ; RESTORE REGISTERS ; RETURN TO CALLER
0530 0542 0542 0543 0544 0544 0544 0554 0551 0551 0551 0555 0555	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE 58 55	C IN C TEST C JZ C PDP C MOV C OUT C PDP C POP C POP C POP C POP C POP C POP C SEND_KEY_TO_32 C POP C RET C SEND_KEY_TO_32 C SEND_KEY_TO_32 C SEND_ENCODED_K C SEND_ENCODED_K C SEND_ENCODED_K C SEND_ENCODED_K C SEND_ENCODED_K C SEND_ENCODED_K	AL.DX AL.DX AL.D2H SEND_KEY_TO_327 AX DX.NDS_KB_OUT DX.AL CX.1 SEND_KEY_TO_327 CX.DELAY_100_HS 5 78_3: DX CX DX CX BX 78_ENUP DS:BIDSDATA.ES: EYSTROKES_TO_3276	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO NOT DELAY 5 ; WAIT FOR 100 MILLISECONDS ; RESTORE REGISTERS ; RETURN TO CALLER :NOTHING 3 ENDP
0530 053E 0540 0542 0543 0546 0546 0546 0547 0544 0546 0547	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE 58 55	C IN C TEST C JZ C POP C MOV C OUT C OUT C CMP C JE C MOV C LOOP C JE C POP C POP C POP C POP C POP C POP C POP C SEND_KEY_TO_32 C POP C SEND_KEY_TO_32 C SEND_KEY_TO_32	AL, DX AL, D2H SEND_KEY_TO_327 AX DX, HDS_KB_OUT DX, AL CX, 1 SEND_KEY_TO_327 CX, DELAY_100_HS \$ 78_3: DX CX BX 78_ENDP DS:BIDSDATA, ES: EYSTROKES_TO_3276 Name: MOD_XLAT_TH	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO NOT DELAY 5 ; WAIT FOR 100 MILLISECONDS ; RESTORE REGISTERS ; RETURN TO CALLER :NOTHING 3 ENDP
0530 0542 0542 0542 0543 0542 0542 0544 0546 0547 0554 0555 0555 0555 0555	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE 58 55	C IN C TEST C JZ C PGP C MOV C OUT C O	AL, DX AL, D2H SEND_KEY_TO_327 AX DX, HDS_KB_OUT DX, AL CX, 1 SEND_KEY_TO_327 CX, DELAY_100_HS \$ 78_3: DX CX BX 78_ENDP DS:BIDSDATA, ES: EYSTROKES_TO_3276 Name: MOD_XLAT_TH	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO HOT DELAY 5 ; WAIT FOR 100 MILLISECONDS ; RESTORE REGISTERS ; RETURN TO CALLER NOTHING 3 ENDP
0530 0542 0542 0542 0543 0542 0542 0544 0546 0547 0554 0555 0555 0555 0555	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE 58 55	C IN C TEST C JZ C POP C MOV C OUT C OUT C CHP C JE C MOV C LOOP C JE C POP C POP C POP C POP C POP C POP C POP C SEND_KEY_TO_32 C POP C POP C SEND_KEY_TO_32 C SEND_KEY_	AL,DX AL,DX AL,D2H SEND_KEY_TO_327 AX DX,NDS_KB_OUT DX,AL CX,1 SEND_KEY_TO_327 CX,DELAY_100_MS \$ 78_3: DX CX BX 78_ENDP DS:BIDSDATA,ES: EYSTROKES_TO_3276 Name: MOD_XLAT_JN Nodify the inboum	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3; YES, DO NOT DELAY ; WAIT FOR 100 MILLISECONDS ; RESTORE REGISTERS ; RETURN TO CALLER :NOTHING S ENDP d keyboard translation table ey number
0530 0542 0542 0542 0543 0542 0542 0544 0546 0547 0554 0555 0555 0555 0555	A8 02 74 FB 58 BA 03E1 EE 83 F9 01 74 05 B9 681A E2 FE 58 55	C IN C TEST C JZ C POP C MOV C OUT C OUT C CMP C MOV C OUT C CMP C JE C MOV C OUT C CMP C POP C POP C POP C POP C POP C POP C POP C SEND_KEY_TO_32 C POP C SEND_KEY_TO_32 C SEND_F SEND_F SEND_F C SEND_F	AL.DX AL.DX AL.D2H SEND_KEY_TO_327 AX DX.HDS_KB_OUT DX.AL CX.1 SEND_KEY_TO_327 CX.DELAY_100_HS \$ 78_3: DX CX BX 78_ENDP DS:BIDSDATA.ES: EYSTROKES_TO_3276 EYSTROKES_TO_3276 Name: MOD_XLAT_TH Todify the inbour B/3279 relative k	78_2 ; RESTORE SCAN CODE ; SEND THE KEYSTROKE TO THE DISPLAY ; IS THIS THE LAST KEYSTROKE? 78_3 ; YES, DO NOT DELAY 5 ; WAIT FOR 100 MILLISECONDS ; RESTORE REGISTERS ; RETURN TO CALLER :NOTHING 3 ENDP : d keyboard translation table ey number ase yboard scan code or BIOS function code

		4,641,262
	71	72
1541		C ;*
1542		C ;# CF = 0 if parameter values are acceptable #
1543 1544		C ;# = 1 if parameter values are not acceptable # C ;#
1545 1546		C ;* AL = 3 (invalid parameter specified) if parameter values are *
1545		C 3# not acceptable C 3#
1548 1549		C J* BX on stack = old table entry *
1550		C ;** C ;*******************************
1551 1552	0555	C HOD_XLAT_IN PROC NEAR
1553		c · · · ·
1554 1555	0555 80 FD 01 0558 72 19	C CMP CH,I ; VERIFY USER PARAMETERS C JB MOD_XLAT_IN_I ; 3278/3279 KEY NUMBER MUST BE BETWEEN
1556	0554 80 FD 59	C CMP CH,89 ; 1 AND 89
1557 1558	055D 77 14 055F 80 F9 01	C JA MOD_XLAT_IN_1 C CMP CL,1 ; 3278/3279 CASE MUST BE BETWEEN 1 AND 5
1559	0562 72 DF	C JB MOO_XLAT_IN_1
1560 1561	0564 80 F9 <b>05</b> 0567 77 DA	C CHP CL,5 C JA MOD_XLAT_IN_1
1562	0569 F6 C6 80	C TEST DH, BOH ; IS THIS A SPECIAL FUNCTION CODE?
1563 1564	056C 75 13 056E 80 FE 53	C JNZ MOD_XLAT_IN_2 ; YES, IT IS OK C CHP DH,83 ; PERSONAL COMPUTER SCAN CODE MUST BE
1565	C571 76 DE	C JHA MOD_XLAT_IN_2 ; O OR A KEY NUMBER BETWEEN 1 AND 83
1566 1567	0573	C MOD_XLAT_IN_1:
1568 1569	0573 B0 09 0575 F9	C MOV AL,RC_INVALID_PARM ; INDICATE INVALID PARAMETER C STC ; SPECIFIED
1570	0576 C3	C RET
1571 1572	0577	C MOD_XLAT_IN_TABLE LABEL WORD ; KEYBOARD TRANSLATION TABLE SECTIONS
1573	0577 0118 R	C DH BIDSDATA: LOWER_CASE_XLAT
1574 1575	0579 DICA R 0578 0270 R	C DH BIDSDATA:UPPER_CASE_XLAT C DW BIDSDATA:CTL_CASE_XLAT
1576	057D 032E R	C DH BIOSDATA:ALT_CASE_XLAT
1577 1578	057F 03E0 R	C DW BIDSDATA:ALT_KEY_XLAT C
1579	0581	C MOD_XLAT_IN_2:
1580 1581	0581 8A C1 0583 DD E0	C MOV AL.CL ; FIND ADDRESS OF APPROPRIATE SECTION C SAL AL.1 ; OF THE KEYBOARD TRANSLATION TABLE
1582 1583	0585 98 0586 88 FD	C CEW . C MOV SI.AX
1584	0588 2E: 8B 9C 0575 R	C HOV BX, MOD_XLAT_IN_TABLE[SI-2]
1585 1586	0580 84 C5	C MOV AL,CH ; FIND INDEX INTO TRANSLATION TABLE
1587	DSAF DO EO	C SAL AL,1 ; SECTION
1588 1589	0591 88 F8	C HOV DI.AX
1590 1591	0593 87 51 FE	C XCHG WORD PTR [BX+DI-2],DX ; FUT NEW ENTRY INTO TABLE
1592	0596 55	C PUSH BP ; PUT OLD TABLE ENTRY INTO THE STACK
1593 1594	0597 85 EC 0599 89 56 10	C MOV BP,SP C MOV [BP+16],DX
1595	059C 5D	C POP BP
1596 1597	059D F8	C CLC ; INDICATE SUCCESSFUL COMPLETION
1598	059E C3	C TET : RETURN TO CALLER
1599 1600	059F	C MOD_XLAT_IH EKOP
1601		c
1602 1603		
1604		L C ; + # # # # # # # # # # # # # # # # # #
1605 1606		C }* Procedure Name: MOD_XLAT_OUT
1607 1608		C ;* C ;* Function: Modify the outbound keyboard translation table
1609		C ;*
1610 1611		C ;# Input: C ;#
1612		C ;* CH = encoded keystroke value C ;* DH = 3278/3279 relative key number
1613 1614		C ;* DL = 3276/3279 keyboard case
1615 1616		C ;* _ C ;* Output:
1617		C ;#
1618 1619		C ;* CF = D if parameter values are acceptable C ;* = 1 if parameter values are not acceptable
1620 1621		C ;* C ;* AL = 3 (invalid parameter specified) if parameter values are
1622		C ;* not acceptable
1623 1624		C ;* C ;*   BX on stack = old table entry
1625		C ;#
1626 1627		C ;************************************
1628	059F	C MOD_XLAT_OUT PROC NEAR
1629 1630	059F 0A F6	C OR DH.DH : O INDICATES INVALID KEY STROKE -
1631 1632	05A] 74 0E 05A3 80 FE 01	C JZ MOD_XLAT_OUT_2 ; THIS IS ACCEPTABLE C CHP DH,1 ; VERIFY USER PARAMETERS
1633	0546 72 05	C JB MOD_XLAT_DUT_1 ; 3278/3279 KEY NUMBER MUST BE BETWEEN
1634	0548 80 FE 59	C CMP DH,89 ; 1 AND 89

.

4,641,262

				4,0	41,202	
		73				74
1635	DSAB	76 04	с	AHL	HOD_XLAT_OUT_2	••
1636			č	JIM	100_XLA1_001_C	
1637	DSAD		ĉ	HOD XLAT OUT 1:		
1638		B0 09	С	Fov	AL, RC_INVALID_P	ARM ; INDICATE INVALID PARAMETER
1639	05AF	F9	C	STC		; SPECIFIED
1640	05B0	C3	С	RET		i
1641 1642	0581		C			
1643	05B1	8A C5	C C	HOD_XLAT_DUT_2:		; FIND INDEX INTO TRANSLATION TABLE
1644	0583	32 E4	C	MOV XOR	AL,CH AH,AH	, FIND INDEX INTO INCODECTION THEFE
1645	05B5		č	SAL	AX,1	
1646	05B7	8B F8	č	HOV	DI,AX	
1647			C			
1648	05B9	87 95 0492 R	С	XCHG	KEYSTROKE_XLAT[]	DII,DX ; PUT NEW ENTRY INTO TABLE
1649			С			
1650 1651	05BD	55 AB FC	C	PUSK	BP	; PUT OLD TABLE ENTRY INTO THE STACK
1652	056E 05C0	88 EC 89 56 10	с с	MOV	BP,SP	
1653	05C3	50	č	MOV POP	[6P+16],DX BP	
1654			č			
1655	0504	F8	č	CLC		; INDICATE SUCCESSFUL COMPLETION
1656	05C5	C3	С	RET		; RETURN TO CALLER
1657			Ċ			
1658	05C6		C	MOD_XLAT_OUT EN	DP	
1659 1660			с с			
1661			c			
1662			c	*****	******	*水
1663			č	;*		÷
1664			Ċ		me: QUERY_KEYBOA	ARD_ID ,
1665			С	3 H	-	
1666			С			ntifier of the current keyboard
1667			C		ranslation table	
1668 1669			с с	;* 		
1670			č	;* Input: Nome ;*		ł
1671			č	;* Output:		•
1672			č	;*		,
1673			с	;* BX = keybo	oard translation	table identifier *
1674			С	5 M		۲ بر این
1675			с	****	****	<b>* 建浆水浆水油 水液学浓浓水液水水水水水水水水水水水水水水水水水水水水水水水水水水水水水水水水</b>
1676			C			
1677 1678	05C6		C C	QUERY_KEYBOARD_	ID PROC NEAR	
1679	05C6	55	č	PUSH	BP	; SAVE REGISTER
1680	0500	55	č	Fush		
1681	05C7	88 1E 0116 R	č	MOV	BX,XLAT_TABLE IN	; PUT THE TRANSLATION TABLE IDENTIFIER
1682	05CB	6B EC	С	MOV	BP, SP	; INTO THE STACK SO IT WILL
1683	05CD	89 5E 10	С	MOV	[BP+16],BX	BE RESTORED INTO THE BX REGISTER
1684			С			
1685	05D0	5D	C	POP	BP	RESTORE REGISTER
1686 1687	05D1	63	с с	RET		; RETURN TO CALLER
1688	0501	L3	c	KEI		S REIDEN ID GREECE
1689	0502		c	QUERY_KEYBOARD_	ID ENDP	
1690			č			
1691			-			
1692	0502			NDSPBIOS ENDS		
1693						
1694				END		

#### APPENDIX C

The IBH Personal Computer MACRO Assembler 3278/3279 Attachment Feature Bus Int Handler

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Specifications \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ; Module Name: NDSPBFBA Descriptive Name: 3278/3279 Attachment feature bus interrupt handler . ; Copyright: IBM Corp 1982 ; Status: Version 1.00 , Function: To process all Start Operation commands from the 3274 and ; all 3278/3279 Attachment related function requests from ; an application program ; Notes: Dependencies: None : Restrictions: None . ; Module Type: Program ; Processor: Assembler ÷ 1

.

40 41 42

43 44 45

57

58 59

65

80 81

99 100

101 102

103 104

105

106 107

108 109 110

111

112 113

114 115

116 117

118 119

120 121

122

#### 4,641,262

•

75 ; Entry Point: PROC\_START\_OP Purpose: To process a Start Operation command from the 3274 Linkage: CALL from NDSPBIOS Input Parameters: None Entry Point: NDSP\_API Purpose: To process a 3278/3279 Attachment software interrupt Linkage: INT 50H Input Parameters: AH = 00 - Set display mode AL = 0 - Set Host Computer mode AL = 1 - Set Personal Computer mode AL = 2 - Display Host Computer screen image AL = 3 - Display Personal Computer screen image AH = 01 - Send keystrokes to 3278/3279 DS:DX = address of encoded keystrokes CX = number of encoded keystrokes AH = 02 - Set 3278/3279 Attachment interrupt vector AL = 0 - Data Transfer AL = 1 - Screen Capture AL = 2 - Wrap Test DS:DX = address of interrupt handler AH = 03 - Reset 3278/3279 Attachment interrupt vector AL = 0 - Data Transfer AL = 1 - Screen Capture AL = 2 - Wrap Test AH = 04 - Set input buffer address AL = 0 - Data Transfer AL = 1 - Screen Capture AL = 2 - Wrap Test DS:DX = address of input buffer CX = length of input buffer AH = 05 - Send message to Host Computer AL = 0 - Structured Field AL = 1 - Set Screen Capture Options AL = 2 - Wrap Test Message DS:DX = adoress of message CX = length of message AH = 06 - Cancel pending message AL = 0 - Structured Field AL = 1 - Set Screen Capture Options AL = 2 - Wrap Test Message AH = 07 - Modify inbound keystroke translation table CH = 3278/3279 relative key number CL = 3278/3279 keyboard case DH = Personal Computer keyboard scan code or EIOS function code DL = Personal Computer keyboard case AH = 08 - Modify outbound keystroke translation table CH = encoded keystroke value DH = 3278/3279 relative key number DL = 3278/3279 keyboard case AH = 09 - Query Keyboard ID Exit Normal: Purpose: To return to the interrupted program Linkage: IRET Output: None Exit Error: None External References: Procedure Invoked: PROC\_INTERRUPTS Purpose: To process all other pending hardware interrupt Linkage: CALL Input Parameters: None Procedure Invoked: SEND\_ENCODED\_KEYSTROKES\_T0\_3278 Purpose: To send a series of encoded keystrokes to the 3278/3279

123 124 ; Linkage: CALL 125 126 Input Parameters: 127 128 129 BX:DX = address of encoded keystrokes CX = number of encoded keystrokes 130 131 Dutput: 132 133 CF = 0 if keystrokes were all valid 134 I if any keystroke was invalid for the current keyboard AL = 3 if any keystroke was invalid for the current keyboard 135 136 137 Procedure Invoked: MOD\_XLAT\_IN 138 139 Furpose: To modify the inbound keyboard translation table 140 141 142 Linkage: CALL 143 Input Parameters: 144 CH = 3278/3279 relative key number CL = 3278/3279 keyboard case 145 146 DH = Personal Computer keyboard scan code or BIOS function code DL = Personal Computer keyboard case 147 148 149 150 Output: 151 CF = 0 if parameters were valid 1 if parameters were invalid AL = 9 if parameters were invalid 152 153 154 155 z 156 Procedure Invoked: MOD\_XLAT\_DUT 158 Purpose: To modify the outbound keyboard translation table 159 160 Linkage: CALL 161 Input Parameters: 162 163 CH = encoded keystroke value DH = 3278/3279 relative key number DL = 3278/3279 keyboard case 164 165 5 166 167 ÷ 168 Output: ï 169 170 CF = 0 if parameters were valid 1 if parameters were invalid AL = 9 if parameters were invalid 171 : 173 Procedure Invoked: QUERY\_KEYBOARD\_ID 175 Purpose: To obtain the keyboard translation table identifier 176 178 Linkage: CALL 179 180 Input Parameters: None 181 Output: keyboard identifier on stack in place of BX 1.6.2 183 184 Control Blocks: 185 NDSPDCLS - 3278/3279 Attachment option interfaces NDSPUSER - 3278/3279 Attachment EIOS extension interfaces NDSFDCOM - 3278/3279 Attachment EIOS common data area 186 187 188 189 190 ; Messages Issued: None 191 ; Change Activity: None 192 193 , ; ###F######FF########### End of Specifications ################################# 195 SUBTTL WORK AREAS 196 197 198 INCLUDE NOSPOCLS.ASH 199 ; INCLUDE NDSPUSER.ASM 200 LIST 201 BIOSDATA GROUP BIOSDATA\_COMMON, BIOSDATA\_FBA 202 203 INCLUDE NDSPBCOM.ASH 204 с BIOSDATA\_COMMON SEGMENT COMMON 'NDSPBICS' 0000 205 206 ;\* 207 ;\* 208 HOSPEIDS COMMON WORK APEAS : \* 209 ;× 210 211 212 DB 0 EQU 00H ; CURRENT MODE OF TERMINAL TERM\_MODE 213 0000 00 J PERSONAL COMPUTER HODE J HOST PROCESSOR MODE c c 214 = 0000 T\_PC T\_HOST EQU BOH = 0080 215 216

			'						00
217 218	0001			c c	BIOSDATA_COMMON ENDS				
219				C.					
210	0000				BIDSDATA_FBA SEGMENT	FUBL	เม่อ "แอะเ	PD.	103.
222 223					*******	****	******	***	有手边驾驶放驶放车盘装板装放车筒把水气带力去放车的边垫放车的车车的桌上
224 225					;* NDSPBIOS FEAT	IURE	BUS ADA	LPT	ER NORK AREAS *
226 227					<b>;</b> *				<b>金子在龙水浓水的就在各些水水下水——在山山、小水水水中中水的水中中的水水水水水水水水水水水水水水水水水水水水水水水水水水水</b>
228									
229 230	= 681	A			DELAY_100_HS EQU	2665		;	FOCESSING FOR 100 MILLISECONDS
231 232					PUBLIC FBA_S	STATI	E		
233 234	0000 = 008				FBA_STATE F_READ_PENDING	DB FOU		; ;	FEATURE BUS ADAFTER STATE READ REQUEST PENDING
235	= 004	0			F_READ_IN_PROGRESS	EQU	40H	F	READ SEQUENCE IN PROGRESS
236 237	= 002 = 001				F_WRITE_IN_PROGRESS F_QUERY_IN_PROGRESS			;	
238 239	= 000				F_DEFERRED_CANCEL			\$	CANCEL MESSAGE WAS DEFERRED
240	= 000				F_ENABLED			;	FEATURE BUS ADAPTER IS ENABLED
241 242	= 000	1			F_FBA_NOT_OP	EQU		3	FEATURE BUS ADAPTER NOT OPERATIONAL
243 244	0001 = 008				FBA_PENDING_OP F_READ	DB FQU		;	FEATURE BUS ADAPTER PENDING OPERATIONS READ REQUEST RECEIVED
245	= 008				F_POR	EQU	40H	;	POWER ON RESET RECEIVED
246 247	= 002 = 001				FEXECUTE			i	EXECUTE RECEIVED ABORT TRANSMISSION RECEIVED
247 248	= 000				F_ABORT_E F_ABORT_V			;	ABORT TO LAST VERIFY RECEIVED
249	= 000				F_ERROR			;	ERROR RECEIVED USER SPECIFIED CANCEL MESSAGE
250 251	= 000	د			F_CANCEL	EQU			
252 253		00 00	00 00		FBA_MESSAGE_START	DD DH			ADDRESS OF MESSAGE TO BE SENT LENGTH OF MESSAGE TO BE SENT
253 254		0000 00 00	00 00		FBA_MESSAGE_SIZE FBA_MESSAGE	00 00	0	1	ADDRESS OF NEXT LYTE TO BE SENT
255	000C	0000			FBA_MESS/GE_LENGTH	DW			NUMBER OF BYTES FEMALINING TO BE SENT TYPE OF MESSAGE TO BE SENT
256 257	ODOE	00			FBA_MESSAGE_TYPE	DB			
258 259	0000	00 00	~~ ~~		FBA_VECTOR	STRI			INTERRUPT VECTOR FORMAT ADDRESS OF INTERPUPT HANDLER
257		00 00			INTERRUPT_VECTOR BUFFER_ADDRESS	סס	0		ADDRESS OF INPUT BUFFER
261	0008				BUFFER_LENGTH	DW			LENGTH OF INFUT BUFFER NUMBER OF BYTES IN INPUT BUFFER
262 263	000A 000C				BUFFER_COUNT VERIFY_COUNT	DW	0		NUMBER OF BYTES VERIFIED IN BUFFER
264	000E	00			MESSAGE_TYPE				MESSAGE TYPE
265 266	000F	00			MESSAGE_FLAG5 ;HF_DATA_TRUNC			;	MESSAGE FLAGS DATA TRUNCATED
267					HF_VERIFY_TRUNC			3	VERIFIED DATA TRUNCATED
268 269	0010	00			;MF_IN_USE STATUS_FLAGS		20H 0	;	BUFFER IN USE STATUS FLAGS
270					SF_PROCESSING		80H	;	USER INTERRUPT HANDLER IS PROCESSING A FEATURE BUS INTERRUPT
271 272					SF_OVERRUN		4 OH	;	DATA RECEIVED WHOLE EUFFER IN USE
273 274	0011				FBA_VECTOR	END	S		
275						_			MESSAGE FLAGS WITHEN FBA_VECTOR
276 277	= 008 = 004				HF_DATA_TRUNC HF_VERIFY_TRUNC		I 80H I 40H	;	DATA TRUNCATED Verified data truncated
275	= 004				MF_IN_USE		201	;	BUFFER IN USE
279 280	= 008	0			SF_PROCESSING	EQU	аон	;	
281	= 004				SF OVERPUN		4 DH	;	A FEATURE BUS INTERRUPT DATA RECEIVED WHILE BUFFER IN USE
282 283					-				
284 285		00 00 00 00			DATA_XFER_VECTOR	FB	A_VECTOR	R •	<,,,,,HT_SF_TO_HOST>
286	0017	0000							
287 288	0019 001B								
269	0010	02							
290 291	001E 001F	00 00							
292									
293 294	0020 0024	00 00 00 00			SCREEN_CAPTURE_VECTOR	FE)	A_VECTOR	? <	+++++HT_SET_CC_OPTIONS>
295	0028	0000							
296 297	002A 002C	0000 0000							
298	002E	04							
299 300	002F 0030	00 00							
301					INAN YEAR INTERNA			<b>.</b> .	UT 10010 TO 7274N
302 303	0031 0035	00 00 00 00			WRAP_TEST_VECTOR	۲B	A_VECTOR	~ <	,,,,,HT_WRAP_T0_3274>
304	0039	0000							
305 306	003B 003D								
307	003F	00							
308 309	0040 0041	00 00							
310		- •							

.

			4,641,	262	
		81			82
311 312		00 00 00 00	QUERY_STATE_VECTOR	FBA_VECTOR	<,,,,,HT_QUERY_KBD_STATE>
313	0046 004a				
314		0000			
315	004E	0000			
316 317	0050	-			
318	0051 0052	00 00			
319					
320					
321	0053	0000	FBA_SEND_VECTOR	••••••	INTERRUPT VECTOR FOR MESSAGE BEING
323	0055	0000	FBA_RECEIVE_VECTOR		SENT TO THE 3274 INTERRUPT VECTOR FOR MESSAGE BEING
324	-		·DA_REGETTE_TECTOR		RECEIVED FROM THE 3274
325 326	0057				
327	0057	00 00 00 00	SC_SECTION_LENGTH		ADDRESS OF LENGTH FIELD IN CURRENT
328				;	SCREEL CAPTURE BUFFER
329	005B	0 D	QUERY_KBD_STATE_HSG	DB 0 ;	DUMMY HESSAGE FOR CUERY KEYBDARD STAT
330 331	005C 005D		KBD_STATE		CURRENT KEYDOARD STATE
332	0050	60	QUERY_RC	08 0 ;	RETURN CODE FROM QUERY KEYBOARD STATE
333	005E		BIOSDATA_FBA ENDS		
334			_		
335			SUBTTL Pro	cess a start o	peration command from the 3274 control
336 337					
338				*****	P 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
339			; <del>•</del>		
340 341			<pre>3* Procedure Name:</pre>	PROC_START_OP	
342			;* :* Function: To pr	ncess a Start (	Operation command from the 3274
343			;*		
344 345				eration modifi-	er from 3278/3279 Attachment Option
346			;* ;* Output: None		
347			3*		
348			******	********	就在冰水就都拿着中意不要就有这个意义就是就没有就有这些我们就要不能不是不能能能
349 350	0000		NDSPBIDS SEGMENT PU	NTC INTERTOS	1
351			NUSPBIDS SCORENT FO	DETE NUSPOIDS	
352			PUBLIC PRO		
353 354			PUBLIC NDS		F 1 D
355				LINTERRUPTS:N	TROKES_TO_3278:NEAR
356				XLAT_IN:NEAR	
357				XLAT_OUT:NEAR	
358 359			EXTRN QUE	Y_KEYBOARD_ID	:NEAR
360			ASSUME CS:	OSPBIOS,DS:BI	DSDATA
361					
362 363	0000		NDSP_SOP PROC NEAR		
364		c	INCLUDE NDS	SOP1.ASM	
365		c	:		
366 367		0		*****	<b>张大武水水东水水水水水水水水水水水水水水水水水水水水水水水水水水</b>
368				PT OPERATION	MODIFIER AND FROCESS IT
369		c			
370		C		********	<b>张家家水水家家家水家水水水水水水水水水水水水水水水水水水水水水水水</b>
371 372	0000	C C C		ABEL MOSO	•
373	0000	0025 R C		RECEIVED	
374		0081 R C	DH REAL	_RECEIVED	
375 376		00AD R C		TE_RECEIVED	
377		D20C R C		T_E_RECEIVED	
378		0235 R C	DW ABO	T_V_RECEIVED	
379 380		0266 R C		R_RECEIVED	
381	= 0010			IFY_RECEIVED ENGTH EQU \$-S	TART OF JMP TABLE
382		c	:		
383	0010	C			BEAD THE STILL ODEDITION MODIFIED
384 385	0010	BA 03E5 C			READ THE START OPERATION MODIFIER
366					
387		FE CB C	DEC AL		
388 389		D0 E0 C 98 C			ROUTE CONTROL BASED ON THE START OPERATION MODIFIER
390		3C 10 C		TART_OP_JHP_T	
391		77 07 C	JA INV.	LID_SON	
392	0010	2 EO		.v	
393 394		BB F0 C 2E: FF A4 0000 R C		XX RT_OP_JMP_TABL	EISII
395		C			
396	0024				
397 398	0024	C3 C		;	INVALID START OPERATION MODIFIER - IGNORE IT - THE 3274 WILL TIME
390		C		;	DUT AND TAKE THE 3278/3279 DFFLINE
400		c	:		
401		C			
402		C			

		83			04
403			С ;	********	中接过这条状态中有不能不不不能完全并且在大波就这些浓度的浓度的浓度的变化的变化不是有不是有不是有不是有不是有
404			Ci		·
405 406			C ;:		Name: POR_RECEIVED
407			C ;		To process a Start Operation command from the 3274
408			С;	<b>H</b> 1	which indicates Power On Reset
409 410			C;		
411			C ;: C ;:	•	• · · ·
412			C 1	• Output:	
413 414			C ;		ve interrupt handlers are called with a reason code
415			C i		ng Power On Reset
416			C ;		-
417 418			С; С	*****	我就就就就最爱你就没当这事外系派不安没没有没有不要不要不要不要没有没有不要要没有不要不要不要不要不要不要不要不要不要
419	0025			DR_RECEIVED:	
420			C	ASSUME	ES:FBA_BUFFER ; ESTABLISH ADDRESSABILITY TO THE
421 422	0025	BA R 8E C2	C C	MOV	DX,FBA_BUFFER ; FEATURE BUS ADAPTER BUFFER ES,DX
423			C		
424 425	002A 002D	B8 OOFC	ç	HOV	AX, SIZE_FBA_BUFFER-4 ; PUT LENGTH OF DATA AREA INTO
426	0020 002F	86 ED 26: A3 DODO R	с с	XCHG MOV	AH,AL ; BUFFER HEADER HDR_LENGTH,AX
427		26: C6 06 0002 R 01	С	HOV	HDR_TYPE, HT_ID ; INDICATE THIS IS A PC ATTACHMENT
428 429	0070	FP 6343 B	C	<b>C</b> (1)	SCHO ON COMPLETE & ACCUPULEDCE THE POLICE ON PESET
430	0039	E8 02A3 R	с с	CALL	SEND_OP_COMPLETE ; ACKNOWLEDGE THE POWER ON RESET
431	003C	F6 06 0000 R 80	С	TEST	FBA_STATE,F_READ_PENDING ; IS THERE A READ REQUEST
432		7/	C		; PENDING?
433 434	0041	74 OA	с с	JZ	POR_1 ; NO, CONTINUE
435	0043	80 DE 0001 R 40	C	OR	FBA_PENDING_OP,F_POR ; YES, INDICATE FOR RECEIVED
436	0048	80 26 0000 R BF	ç	C1A	FBA_STATE, NOT F_READ_IN_FROGRESS ; READ SEQUENCE IS NOW
437 438			с с		; CANCELLED
439	004D		C P	DR_1:	
440 441	0040	80 26 0000 R DF	с с	C1A C1A	<pre>FBA_STATE,NOT F_WRITE_IN_PROGRESS ; WRITE SEQUENCE IS</pre>
442			c		; CANCELLED
443	0052		С	CALL	PROC_INTERRUPTS ; PROCESS OTHER PENDING INTERRUPTS
444 - 445	0055	FB	с с	STI	
446	0056	B4 02	c	HOV	AH, REASON_POWER_ON_RESET ; INDICATE POR RECEIVED
447			С		
448 449	0058 005C	8D 36 000F R E8 02C2 R	с с	LEA CALL	SI, DATA_XFER_VECTOR ; INFORM ALL INTERRUFT Reset_Euffer_counts ; handlers that a foxer on reset
450		E8 02AD R	c	CALL	INFORM_APPL ; HAS OCCURRED AND RESET ALL INPUT
451			C		; BUFFER COUNTERS
452 453	0062	8D 36 0020 R E8 02C2 R	с с	LEA CALL	SI, SCREEN_CAPTURE_VECTOR RESET_DUFFER_COUNTS
454		EB 02AD R	c	CALL	INFORM_AFPL
455			ç		
456 457	006C 0070	8D 36 0031 R E8 02C2 R	c c	LEA CALL	S1,WRAP_TEST_VECTOR RESET BUFFER COUNTS
458		E8 02AD R	С	CALL	INFORM_APPL
459	0076	90 7/ 00/0 0	c c	154	CT OVERY STATE VECTOR
460 461		8D 36 0042 R E8 02C2 R	c	LEA CALL	SI,QUERY_STATE_VECTOR RESET BUFFER COUNTS
462		EB 02AD R	С	CALL	INFORM_AFPL
463	0080	<b>6</b> 3	с с	DET	I PROCESS OTHER PENDING INTERRUPTS
464 465	0000	13	c	RET	; PROLESS DINER PENDING INTERNOTIO
466			С		
467 465			с с		19.5、9、9、9、9、9、9、9、9、9、9、9、9、9、9、9、9、9、9、
469			C ;+ C ;+		· · · · · · · · · · · · · · · · · · ·
470			С ;	Procedure	Name: READ_RECEIVED
471 472			C ;	H H Exposion:	To process a Start Operation command from the 3274
473			C 31		which indicates Read
474			С;	H	
475 476			C ;: C ;:		, ,
477			C i		ENDING = 1 if a read request is pending
478			C ;		1
479 480			C 3		i
481			C 3		1 *
462			C 3		N_PROGRESS = 1
483 484			C 3		要我我就帮家家来我就就要要我我就要我我就要要你不是我就要要帮我我有我家家就要要我我要要要我就要要没有要要做吗。 。
485			c ,		
486	0081		CR	EAD_RECEIVED:	
487 488		F6 06 0000 R 80 74 0B	C C	TEST JZ	FBA_STATE,F_READ_PENDING ; IS THERE A READ PENDING? READ_1 ; NO, SEND A CLEARED BUFFER TO THE 3274
489	0000		c	JL .	TO CANCEL THE READ REQUEST
490			С		
491 492	0088 008D	80 0E 0001 R 80 80 0E 0000 R 40	с с	OR OR	FBA_PENDING_OP,F_READ ; INDICATE READ RECEIVED FBA_STATE,F_READ_IN_PROGRESS ; INDICATE READ SEQUENCE
492	0000	50 91 9000 R 40	c	UK	; IS NOW IN PROGRESS
494	0092	C3	C	RET	FROCESS NEXT PENDING INTERRUPT
495			£		

496					
	0093		C RE	AD_1:	
497	0093	BA R	С	- MOV	DX,FBA_BUFFER ; CLEAR THE FBA BUFFER
498	0096	BE C2	С	HOV	ES,DX
499	0098	BF 0000	С	MOV	DI,0
500		B9 0080	č	HOV	CX,SIZE_FBA_BUFFER/2
501		33 C0	č		
502	DOAD		c	XOR	AX,AX
503				CLD	
-	COAL	F3/ AB	C	REP	STOSW
504			C		
505	DOA3	EB DZ9A R	С	CALL	HAIT_FOR_POLL_ACK ; HAIT FOR THE PREVIOUS POLL REQUEST
506			C		; TO BE ACKNOWLEDGED
507	ODA6	BA 03E4	С	MOV	DX,COMMAND_PORT ; TELL THE 3274 TO READ THE CLEARED
508	ODA9	B0 02	С	MOV	AL,READ_COMPLETE ; BUFFER
509	ODAB	EE	С	OUT	DX,AL
510			с		
511	DDACO	C3	c	RET	3 PROCESS OTHER PENDING INTERRUPTS
512			č		,
513			č		
514			č		
515					就我我就想要你手要把出来这些不是你不能要要要要要要做你的这些我就要要要不是不是不是是不能要要有不能帮
516				*********	***************************************
517			C ;*		
			C ;*	Procedure N	ame: WRITE_RECEIVED
518			C ;*		
519			C ;*	Function: T	o process a Start Operation command from the 3274
520			C ;*	lei lei	hich indicates Write
521			C ;*		
522			C ;*	Input:	
523			C ;*		
524			C ;*		R contains message from the 3274
525			Č ;+		-
526			C 3*	Output:	
527			C ;*		
528			C ;*	5 LIDITE 7	N_PROGRESS = 1
529			C ;*		N_PROGRESS - 1 VE_VECTOR = address of interrupt vector
530					
531			C ;*	cata save	d in input buffer for the specified data type
532			C ;*		<b>家在在武器官家在这条在这条在这条的主要在这里就在不是不是不是这个是这个不是是不是没有不能没有不能是不是不是不是不</b>
				*********	
533			С		
534	DADO			ITE_RECEIVED:	THE REPORT OF THE
535			С	ASSUME	ES:FBA_BUFFER ; ESTABLISH ADDRESSABLITY TO THE
536	DADO	BA R	С	HOV	DX,FBA_BUFFER   FEATURE BUS ADAPTER BUFFER
537	0080	8E C2	С	MOV	ES,DX
538			С		
539	0082	80 DE 0000 R 20	C	OR	FBA_STATE, F_WRITE_IN_PROGRESS ; INDICATE WRITE SEQUENCE
540			С		IS IN FROGRESS
541	0087	8D 36 000F R	Ċ	LEA	SI, DATA_XFER_VECTOR ; DETERMINE WHICH INTERRUPT
542		26: 80 3E 0002 R 03	č	CMP	HDR_TYPE, HT_SF_FROM_HOST ; VECTOR TO USE
543		74 25	č	JE	WRITE_1
544		14 23	č	52	
545	0003	80 36 0020 R	c	154	SI, SCREEN_CAPTURE_VECTOR
546		26: 80 3E 0002 R 05		LEA	
547			c	CMP	HDR_TYPE,HT_SC_DATA
	0000	74 7F	c	JE	WRITE_SC
548			C		
549	OOCF	8D 36 0031 R	C	LEA	SI, WRAP_TEST_VECTOR
550	0003		C	CMP	HDR_TYPE,HT_WRAP_FROM_3276
551	0009	74 DD	С	JE	WRITE_1
552			С		
	0006	8D 36 0042 R	С	LEA	SI,QUERY_STATE_VECTOR
553	0000		С	CMP	HDR_TYPE, HT_KBD_STATE
553 554	OODF				
	ODDF				
554	ODDF	26: 80 3E 0002 R 07	С	JE	WRITE_1
554 555	ODDF	26: 80 3E 0002 R 07 74 01	с с	JE	WRITE_1 ; INVALID MESSAGE HEADER - IGNORE THE
554 555 556 557	00DF 00£5	26: 80 3E 0002 R 07 74 01	С С С		WRITE_1 ; INVALID MESSAGE HEADER - IGNORE THE
554 555 556 557 558	00DF 00£5	26: 80 3E 0002 R 07 74 01	0 0 0 0 0	JE	WRITE_1 ; INVALID MESSAGE HEADER - IGHORE THE ; START DEERATION - THE 3274 WILL TIM
554 555 556 557 558 559	00DF 00£5	26: 80 3E 0002 R 07 74 01	00000	JE	WRITE_1 ; INVALID MESSAGE HEADER - IGNORE THE
554 555 556 557 558 559 560	00DF 00E5 00E7	26: 80 3E 0002 R 07 74 01	000000	JE RET	WRITE_1 ; INVALID MESSAGE HEADER - IGHORE THE ; START DEERATION - THE 3274 WILL TIM
554 555 556 557 558 559 560 561	00DF 00E5 00E7 00E8	26: 80 3E 0002 R 07 74 01 C3	000000 KR	JE RET ITE_1:	WRITE_1 ; INVALID MESSAGE HEADER - IGNORE THE ; START OFERATION - THE 3274 WILL TIM ; OUT AND TAKE THE 3278/3279 OFFLINE
554 555 556 557 558 559 560 561 562	00DF 00E5 00E7 00E8	26: 80 3E 0002 R 07 74 01	CCCCCCC	JE RET	WRITE_1 ; INVALID MESSAGE HEADER - IGHORE THE ; START DEERATION - THE 3274 WILL TIM ; OUT AND TAKE THE 3276/3279 DEFLINE HORD PTR [SI].BUFFER_ADDRESS+2.0 ; WAS AN INPUT BUFFER
554 555 556 557 558 559 560 561 562 562 563	00DF 00E5 00E7 00E8 00E8	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00		JE RET ITE_1: CMP	WRITE_1 ; INVALID MESSAGE HEADER - IGNORE THE ; START DFERATION - THE 3274 HILL TH ; OUT AND TAKE THE 3278/3279 DFFLINE WORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER ; SPECIFIED FOR THIS MESSAGE TYPE?
554 555 556 557 558 559 560 561 562 563 563 564	00DF 00E5 00E7 00E8 00E8 00E8	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45	<b>ссссс</b> ссс	JE RET ITE_1: CMP JE	WRITE_1 ; INVALID MESSAGE HEADER - IGNORE THE ; START OFERATION - THE 3274 HILL TIM ; OUT AND TAKE THE 3278/3279 OFFLINE HORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER ; SFECIFIED FOR THIS MESSAGE TYPE? WRITE 4 ; NO, INDIC/TE DATA DVERSUN
554 555 556 557 558 559 560 561 562 563 564 565	00DF 00E5 00E7 00E8 00E8 00E8	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00	с <u>сс</u> осососососососососососососососососо	JE RET ITE_1: CMP	WRITE_1 ; INVALID MESSAGE HEADER - IGNORE THE ; START DFERATION - THE 3274 HILL TIM. ; DUT AND TAKE THE 3278/3279 DFFLINE HORD PTR [SI].BUFFER_ADDRESS42.0 ; HAS AN INPUT BUFFER ; SFECIFILD FOR THIS MESSAGE TYPE? WRITE_4 ; NO, INDIC/TE DATA DVERRUN [SI].MESSAGE_FLAGS,MF_IN_USL ; IS THE INPUT BUFFER IN
554 555 557 558 559 560 561 562 563 564 565 565 565	00DF 00E5 00E7 00E8 00E8 00E8 00EC	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20		JE RET ITE_1: CMP JE TEST	WRITE_1 ; INVALID MESSAGE HEADER - IGHORE THE ; START DFERATION - THE 3274 WILL TIM. ; OUT AND TAKE THE 3278/3279 DFFLINE HORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER ; SFECIFIED FOR THIS MESSAGE TYPE? WRITE_4 ; NO, INDIC/TE DATA OVERRUN [SI].HESSAGE_FLAGS.HF_IN_USL ; IS THE INPUT BUFFER IN ; USE?
554 555 557 558 559 560 561 562 563 564 565 564 565 566 566 567	00DF 00E5 00E7 00E8 00E8 00E8 00EC	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45		JE RET ITE_1: CMP JE	WRITE_1 ; INVALID MESSAGE HEADER - IGNORE THE ; START DFERATION - THE 3274 HILL TIM. ; DUT AND TAKE THE 3278/3279 DFFLINE HORD PTR [SI].BUFFER_ADDRESS42.0 ; HAS AN INPUT BUFFER ; SFECIFILD FOR THIS MESSAGE TYPE? WRITE_4 ; NO, INDIC/TE DATA DVERRUN [SI].MESSAGE_FLAGS,MF_IN_USL ; IS THE INPUT BUFFER IN
554 555 557 558 559 560 561 562 563 564 565 564 565 566 567 568	00DF 00E5 00E7 00E8 00E8 00E6 00EC 00EE D0F2	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F	аааааааааааааа Ж	JE RET ITE_1: CMP JE TEST JNZ	WRITE_1       ; INVALID MESSAGE HEADER - IGNORE THE         ; START OFERATION - THE 3274 HILL TIM.         ; OUT AND TAKE THE 3278/3279 OFFLINE         WORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         KRITE_4       ; NO, INDIC/TE DATA OVERRUN         [SI].MESSAGE_FLAGS,MF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4       ; YES, INDICATE DATA OVERRUN
554 555 556 557 558 557 558 557 558 560 561 563 564 565 566 566 567 568 569	00DF 00E5 00E7 00E8 00E8 00E6 00EC 00EE D0F2	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20	аааааааааааааааааааааааааааааааааааа	JE RET ITE_1: CMP JE TEST	WRITE_1       ; INVALID MESSAGE HEADER - IGNORE THE         ; START DFERATION - THE 3274 HILL TIM         ; OUT AND TAKE THE 3278/3279 DFFLINE         HORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         WRITE_4       ; NO. INDICITE DATA DVERRUN         ISI].HESSAGE_FLAGS.HF_IN_USL ; IS THE INPUT BUFFER IH         ; USE?         WRITE_4       ; YES, INDICATE DATA OVERRUN         FBA RECEIVE VECTOR,SI ; SAVE THE ADDRESS OF THE
554 555 557 558 557 558 557 558 560 561 563 564 563 564 565 566 566 566 567 568 569 568 569 570	00DF 00E5 00E7 00E8 00E8 00E6 00EC 00EE D0F2	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F	ааааааааааааааааааа ж	JE RET ITE_1: CMP JE TEST JNZ	WRITE_1       ; INVALID MESSAGE HEADER - IGNORE THE         ; START OFERATION - THE 3274 HILL TIM.         ; OUT AND TAKE THE 3278/3279 OFFLINE         WORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         KRITE_4       ; NO, INDIC/TE DATA OVERRUN         [SI].MESSAGE_FLAGS,MF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4       ; YES, INDICATE DATA OVERRUN
554 555 556 557 558 557 558 557 558 560 561 563 564 565 566 566 567 568 569	00DF 00E5 00E7 00E8 00E8 00E6 00EC 00EE D0F2	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F	вававааааааааааааааа Ж	JE RET ITE_1: CMP JE TEST JNZ	WRITE_1       ; INVALID MESSAGE HEADER - IGNORE THE         ; START OFERATION - THE 3274 HILL TIM.         ; DUT AND TAKE THE 3278/3279 DFFLINE         WORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         KRITE_4       ; NO. INDIC/TE DATA DVERRUN         [SI].MESSAGE_FLAGS.MF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         HRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR,SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE
554 555 557 558 557 558 557 558 560 561 563 564 563 564 565 566 566 566 567 568 569 568 569 570	00DF 00E5 00E7 00E8 00E8 00E6 00E6 00E6 00E7 00F2 00F4	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F	ааааааааааааааааааааааааааааааааааааа	JE RET ITE_1: CMP JE TEST JNZ MOV	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START OFERATION - THE 3274 HILL TIM.         ; OUT AND TAKE THE 3278/3279 OFFLINE         WORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SPECIFILD FOR THIS MESSAGE TYPE?         KRITE_4 ; NO, INDIC/TE DATA OVERRUN         [SI].MESSAGE_FLAGS.MF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4 ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR,SI ; SAVE THE ADDRESS OF THE         ; INTERRUFT VECTOR FOR THIS MESSAGE         CX.HOR_LENGTH ; PICK UP LENGTH OF THIS BLOCK
554 555 557 558 557 558 560 561 562 563 564 565 564 565 566 567 568 567 568 567 568 567 571	00DF 00E5 00E7 00E8 00E8 00E6 00E6 00E6 00E7 00F2 00F4	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R	вававааааааааааааааа Ж	JE RET ITE_1: CMP JE TEST JNZ MOV	WRITE_1       ; INVALID MESSAGE HEADER - IGNORE THE         ; START OFERATION - THE 3274 HILL TIM.         ; DUT AND TAKE THE 3278/3279 DFFLINE         WORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         KRITE_4       ; NO. INDIC/TE DATA DVERRUN         [SI].MESSAGE_FLAGS.MF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         HRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR,SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE
554 555 556 557 558 557 558 560 561 562 563 564 565 566 566 566 566 567 566 569 570 571 572	00DF 00E5 00E7 00E8 00E8 00E8 00E2 00E2 00F2 00F4	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 8B 0E D000 R		JE RET ITE_1: CMP JE TEST JNZ MOV	WRITE_1       ; INVALID MESSAGE HEADER - IGNORE THE         ; START OFERATION - THE 3274 HILL TIM.       ; OUT AND TAKE THE 3278 HILL TIM.         ; OUT AND TAKE THE 3278/3279 OFFLINE         HORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         KRITE_4       ; NO. INDIC/TE DATA DVERRUN         ISI].HESSAGE_FLAGS.HF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         HRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR.SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX.HOR_LENGTH       ; PICK UP LENGTH OF THIS BLOCK         CH.CL
554 555 557 558 557 558 557 558 550 561 563 564 565 564 566 566 566 566 567 568 567 568 570 571 572 573	00DF 00E5 00E7 00E8 00E8 00E8 00E2 00E2 00F2 00F4	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 8B 0E D000 R	аарарараасаарарар Ж	JE RET ITE_1: CMP JE TEST JNZ MOV	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START OFERATION - THE 3274 HILL TIM.         ; OUT AND TAKE THE 3278/3279 OFFLINE         WORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SPECIFILD FOR THIS MESSAGE TYPE?         KRITE_4 ; NO, INDIC/TE DATA OVERRUN         [SI].MESSAGE_FLAGS.MF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4 ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR,SI ; SAVE THE ADDRESS OF THE         ; INTERRUFT VECTOR FOR THIS MESSAGE         CX.HOR_LENGTH ; PICK UP LENGTH OF THIS BLOCK
554 555 556 557 558 560 561 562 563 564 565 566 566 567 570 570 571 572 573 575	00DF 00E5 00E7 00E8 00E8 00E8 00E8 00E8 00E2 00F2 00F4 00F8 00FD	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 8B 0E D00D R 86 E9	аааааааааааааааааааааааааааааааааааа	JE RET ITE_1: CMP JE TEST JNZ MOV XCHG	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START OFERATION - THE 3274 HILL TIM.         ; OUT AND TAKE THE 3278/3279 OFFLINE         HORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         KRITE_4       ; NO, INDIC/TE DATA DVERRUN         [SI].HESSAGE_FLAGS,MF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR,SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX,HOR_LENGTH ; PICK UP LENGTH OF THIS BLOCK         CH,CL         DI,KORD PTR [SI].EUFFER_ADERESS ; PICK UP ADDRESS OF
554 555 557 558 557 560 561 563 564 565 566 566 566 566 567 566 570 571 572 573 574 575 576	00DF 00E5 00E7 00E8 00E8 00E8 00EE 00F2 00F4 00F6 00F6	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 8B 0E 0000 R 86 E9 8B 7C 04	ឧបកកកកកកកកកកកកកកកកកកកកកកកកកកកកកកកកកកកក	JE RET ITE_1: CMP JE TEST JNZ MOV XCHG	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START OPERATION - THE 3274 HILL TIM         ; OUT AND TAKE THE 3278/3279 OFFLINE         KORD PTR [SI].BUFFER_ADDRESS+2.0 ; WAS AN INPUT BUFFER         ; SFECIFILD FOR THIS MESSAGE TYPE?         KRITE_4       ; NO, INDIC/TE DATA DVERRUN         [SI].HESSAGE_FLAGS,MF_IN_USL ; IS THE INPUT BUFFER IN         ; VESE?         WRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR,SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX,HOR_LENGTH ; PICK UP LENGTH OF THIS BLOCK         CH,CL         DI,WORD PTR [SI].BUFFER_ADDRESS ; PICK UP ADDRESS OF         ; INPUT DUFFER
554 555 557 558 557 558 560 561 563 564 565 564 565 566 566 567 568 566 571 573 574 575 577	00DF 00E5 00E7 00E8 00E8 00E8 00EE 00F2 00F4 00F6 00F6	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 8B 0E D00D R 86 E9	оаоааааааааааааааааааааааа Ж	JE RET ITE_1: CMP JE TEST JNZ MOV XCHG MDV	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START OPERATION - THE 3274 HILL TIM.         ; OUT AND TAKE THE 3278/3279 OFFLINE         KORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         KRITE_4       ; NO, INDIC/TE DATA DVERRUN         ISI].HESSAGE_FLAGS.HF_IN_USL ; IS THE INPUT BUFFER IN         ; WSE?         WRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR,SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX,HDR_LENGTH       ; PICK UP LENGTH OF THIS BLOCK         CH,CL       ; INPUT DUFFER         ADJRESS ; PICK UP ADDRESS OF         ; INPUT DUFFER         AX,(SI].BUFFER_COUNT ; PICK UP NUMBER OF BYTES CURRENTL
554 555 556 557 558 560 561 562 563 564 566 566 566 566 570 570 572 573 575 578 578	00DF 00E5 00E7 00E8 00E8 00E6 00EE 00F2 00F4 00F8 00F5 00FF	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 88 0E 0000 R 86 E9 88 7C 04 88 44 0A	ооооооооооооооооооооооооооооооооооооо	JE RET ITE_1: CMP JE TEST JNZ MOV XCHG MOV MOV	<pre>WRITE_1     ; INVALID MESSAGE HEADER - IGNORE THE     ; START OFERATION - THE 3274 HILL TIM     ; DUT AND TAKE THE 3278/3279 DFFLINE     ; SFECIFIED FOR THIS MESSAGE TYPE?     KRITE_4 ; SFECIFIED FOR THIS MESSAGE TYPE?     kRITE_4 ; NO, INDIC/TE DATA DVERRUN     ISIJ.MESSAGE_FLAGS.MF_IN_USL ; IS THE INPUT BUFFER IN</pre>
554 555 557 558 557 560 561 563 564 565 566 566 566 566 567 570 571 572 575 576 577 578 579	00DF 00E5 00E7 00E8 00E8 00E8 00E2 00F2 00F4 00F6 00F6 00FF 0102 0105	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 88 0E 0000 R 86 E9 88 7C 04 88 44 DA 03 F8		JE RET ITE_1: CMP JE TEST JNZ MOV XCHG MOV XCHG MOV ADD	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START OPERATION - THE 3274 HILL TIM         ; OUT AND TAKE THE 3276/3279 OFFLINE         KORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         KRITE_4       ; NO. INDIC/TE DATA OVERRUN         ISI].HESSAGE_FLAGS.MF_IN_USL ; IS THE INPUT BUFFER IN         FRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR.SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX.HOR_LENGTH       ; PICK UP LENGTH OF THIS BLOCK         CH.CL       DI,KORD PTR [SI].BUFFER_ADERESS ; PICK UP ADDRESS OF         j INPUT DUFFER       ALCLULATE WHERE THE NEW DATA GOES
554 555 557 558 557 560 561 563 564 565 566 566 566 566 571 573 576 577 578 577 578 577 578 577 578 578 578	00DF 00E5 00E7 00E8 00E8 00E6 00EE 00F2 00F4 00F8 00F5 00FF	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 88 0E 0000 R 86 E9 88 7C 04 88 44 DA 03 F8		JE RET ITE_1: CMP JE TEST JNZ MOV XCHG MOV MOV	<pre>WRITE_1     ; INVALID MESSAGE HEADER - IGNORE THE     ; START OFERATION - THE 3274 HILL TIM     ; DUT AND TAKE THE 3278/3279 DFFLINE     ; SFECIFIED FOR THIS MESSAGE TYPE?     KRITE_4 ; SFECIFIED FOR THIS MESSAGE TYPE?     kRITE_4 ; NO, INDIC/TE DATA DVERRUN     ISIJ.MESSAGE_FLAGS.MF_IN_USL ; IS THE INPUT BUFFER IN</pre>
554 555 557 558 557 558 560 561 562 563 564 565 566 567 568 567 571 572 574 577 577 577 577 577 577 577 577 578 579 561	00DF 00E5 00E7 00E8 00E6 00EE 00F2 00F4 00F8 00F7 00FF 0102 0105 0107	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 88 0E 0000 R 86 E9 88 7C 04 88 44 0A 03 F8 03 C1	аааааааааааааааааааааааааааааааааааа	JE RET ITE_1: CMP JE TEST JNZ MOV XCHG MOV MOV MOV ADD	WRITE_1       ; INVALID MESSAGE HEADER - IGNORE THE         ; START OFERATION - THE 3274 HILL TIM.       ; OUT AND TAKE THE 3278 HILL TIM.         ; OUT AND TAKE THE 3278/3279 OFFLINE         WORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         KRITE_4       ; NO. INDIC/TE DATA DVERRUN         ISI].MESSAGE_FLAGS.MF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4       ; YES, INDIC/ATE DATA OVERRUN         FBA_RECEIVE_VECTOR.SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX.HOR_LENGTH       ; PICK UP LENGTH OF THIS BLOCK         CH.cL       ; INPUT DUFFER         AX.(SI].BUFFER_COUNT ; PICK UP ADDRESS ; PICK UP ADDRESS OF         ; IN THE INPUT BUFFER         AX.(SI].BUFFER_COUNT ; PICK UP RUTABER OF BYTES CURRENTL         ; IN THE INPUT BUFFER         DI.AX       ; CALCULATE WHERE THE NEW DATA GOES         AX.CX       ; UPDATE NUMBER OF BYTES RECEIVED
554 555 557 558 557 558 560 561 566 566 566 566 566 566 570 571 572 578 5778 5778 5778 5778 5778 5778 5	00DF 00E5 00E7 00E8 00E8 00E8 00E2 00F2 00F4 00F6 00F6 00FF 0102 0105	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 88 0E 0000 R 86 E9 88 7C 04 88 44 DA 03 F8	опппопппппппппппппппппппппппппппппппп	JE RET ITE_1: CMP JE TEST JNZ MOV XCHG MOV XCHG MOV ADD	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START DFERATION - THE 3274 HILL TIM         ; OUT AND TAKE THE 3276/3279 OFFLINE         KORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         WRITE_4       ; NO. INDICITE DATA OVERRUN         ISI].HESSAGE_FLAGS.HF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR.SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX.HOR_LENGTH ; PICK UP LENGTH OF THIS BLOCK         CH.cL         DI,KORD PTR [SI].BUFFER_ADDRESS ; PICK UP ADDRESS OF         ; INPUT DUFFER         AX,[SI].BUFFER_COUNT ; PICK UP RUMBER OF BYTES CURRENTL         ; INPUT DUFFER         DI.AX       ; CALCULATE WHERE THE NEW DATA GOES         AX,(SI].BUFFER_LENGTH ; DOES THE NUMBER OF BYTES RECEIVED         AX,(SI].BUFFER_LENGTH ; DOES THE NEW DATA FIT IN THE
554 555 557 558 560 561 562 563 564 566 566 566 566 571 573 577 578 577 577 578 577 578 577 578 577 578 577 578 577 578 560 561 557 557 560 561 565 566 567 566 567 566 567 566 566 567 566 567 566 567 566 567 566 567 566 567 566 567 566 567 566 567 566 567 566 567 566 567 566 567 566 567 566 566	00DF 00E5 00E7 00E8 00E8 00E2 00F2 00F4 00F4 00F6 00FF 0102 0105 0107 0109	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 88 0E 0000 R 86 E9 88 7C 04 88 44 0A 03 F8 03 C1 38 44 08	а о о а а о о а о о а о о а о о о о о о	JE RET ITE_1:CMP JE TEST JNZ MOV XCHG MOV XCHG MOV ADD ADD CMP	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START OPERATION - THE 3274 HILL TIM         ; OUT AND TAKE THE 3276/3279 OFFLINE         WORD PTR [SI].BUFFER_ADDRESS+2.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         KRITE_4       ; NO, INDIC/TE DATA DVERRUN         [SI].HESSAGE_FLAGS.HF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR.SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX.HDR_LENGTH ; PICK UP LENGTH OF THIS BLOCK         CH.GL         DI,KORD PTR [SI].BUFFER_ADDRESS ; PICK UP ADDRESS OF         ; INPUT DUFFER         AX.(SI].BUFFER_COUNT ; PICK UP NUMBER OF BYTES CURRENTL         ; IN THE INPUT DUFFER         AX,(SI].BUFFER_LENGTH ; DOES THE NUMBER OF BYTES CURRENTL         ; AX, (SI].BUFFER_LENGTH ; DOES THE NUMBER OF BYTES RECEIVED         AX.(SI].BUFFER_LENGTH ; DOES THE NUMBER OF BYTES RECEIVED         AX.(SI].BUFFER_LENGTH ; DOES THE NUM DATA FIT IN THE         ; BUFFEP?
554 555 557 558 560 561 562 564 565 566 566 566 566 566 570 571 572 577 577 577 577 578 570 577 578 570 580 581 582 584	00DF 00E5 00E7 00E8 00E8 00E2 00F2 00F4 00F4 00F6 00FF 0102 0105 0107 0109	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 88 0E 0000 R 86 E9 88 7C 04 88 44 0A 03 F8 03 C1	ааопапаааааааааааааааааааааааааааааааа	JE RET ITE_1: CMP JE TEST JNZ MOV XCHG MOV MOV MOV ADD	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START DFERATION - THE 3274 HILL TIM         ; OUT AND TAKE THE 3276/3279 OFFLINE         KORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         WRITE_4       ; NO. INDICITE DATA OVERRUN         ISI].HESSAGE_FLAGS.HF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR.SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX.HOR_LENGTH ; PICK UP LENGTH OF THIS BLOCK         CH.cL         DI,KORD PTR [SI].BUFFER_ADDRESS ; PICK UP ADDRESS OF         ; INPUT DUFFER         AX,[SI].BUFFER_COUNT ; PICK UP RUMBER OF BYTES CURRENTL         ; INPUT DUFFER         DI.AX       ; CALCULATE WHERE THE NEW DATA GOES         AX,(SI].BUFFER_LENGTH ; DOES THE NUMBER OF BYTES RECEIVED         AX,(SI].BUFFER_LENGTH ; DOES THE NEW DATA FIT IN THE
554 555 557 558 557 558 560 561 562 566 566 566 566 567 570 570 577 577 577 578 577 578 577 578 577 578 579 580 12 579 578 578 578 578 578 578 578 560 561 566 567 578 566 567 566 566 567 566 567 566 566 567 566 566	00DF 00E5 00E7 00E8 00E8 00E2 00F2 00F4 00F8 00F4 00F8 00F7 0102 0105 0107 0109 010C	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 8B 0E 0000 R 86 E9 8B 7C 04 8B 44 0A 03 F8 03 C1 3B 44 08	ааааааааааааааааааааааааааааааааааааа	JE RET ITE_1: CMP JE TEST JNZ MOV XCHG MOV XCHG MOV XCHG MOV XCHG MOV XCHG MOV XCHG MOV XCHG MOV	WRITE_1       ; INVALID MESSAGE HEADER - IGNORE THE         ; START OPERATION - THE 3274 WILL TIM         ; OUT AND TAKE THE 3276/3279 OFFLINE         KORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SFECIFIED FOR THIS MESSAGE TYPE?         WRITE_4       ; NO. INDICITE DATA OVERRUN         ISI].HESSAGE_FLAGS.MF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR,SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX.HOR_LENGTH ; PICK UP LENGTH OF THIS BLOCK         CH,CL         DI,KORD PTR [SI].BUFFER_ADERESS ; PICK UP ADDRESS OF         ; INPUT DUFFER         AX,[SI].BUFFER_COUNT ; PICK UP NUMBER OF BYTES CURRENTL         ; INPUT DUFFER         DI,KORD PTR [SI].BUFFER_ADERESS ; PICK UP ADDRESS OF         ; INPUT DUFFER         DI,KORD PTR [SI].BUFFER_MORTH ; PICK UP NUMBER OF BYTES CURRENTL         ; INPUT DUFFER         DI,AX       ; CALCULATE WHERE THE NEW DATA GOES         AX,(SI].BUFFER_LENGTH ; DOES THE NEW DATA FIT IN THE         ; BUFFER?       ; YES, CONTINUE
554 555 557 558 560 561 562 564 565 566 566 566 566 566 570 571 572 577 577 577 577 578 570 577 578 570 580 581 582 584	00DF 00E5 00E7 00E8 00E8 00E2 00F2 00F4 00F8 00F4 00F8 00F7 0102 0105 0107 0109 010C	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 88 0E 0000 R 86 E9 88 7C 04 88 44 0A 03 F8 03 C1 38 44 08		JE RET ITE_1:CMP JE TEST JNZ MOV XCHG MOV XCHG MOV ADD ADD CMP	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START OPERATION - THE 3274 HILL TIM         ; OUT AND TAKE THE 3276/3279 OFFLINE         WORD PTR [SI].BUFFER_ADDRESS+2.0 ; WAS AN INPUT BUFFER         ; SFECIFILD FOR THIS MESSAGE TYPE?         KRITE_4       ; NO, INDIC/TE DATA DVERRUN         ISI].HESSAGE_FLAGS.HF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR,SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX.HDR_LENGTH ; PICK UP LENGTH OF THIS BLOCK         CH.CL         DI,KORD PTR [SI].BUFFER_ADDRESS ; PICK UP ADDRESS OF         ; INPUT DUFFER         AX,[SI].BUFFER_COUNT ; PICK UP NUMBER OF BYTES CURRENTL         ; IN THE INPUT DUFFER         AX,[SI].BUFFER_LENGTH ; DOES THE NEW DATA GOES         AX,[SI].BUFFER_LENGTH ; DOES THE NEW DATA FIT IN THE         ; BUFFER?         WRITE_2       ; YES, CONTINUE         [SI].MESSAGE_FLAGS.MF_DATA_TRUNC ; NO, THE DATA IS
554 555 557 558 557 558 560 561 562 566 566 566 566 567 570 570 577 577 577 578 577 578 577 578 577 578 579 580 12 579 578 578 578 578 578 578 578 560 561 566 567 578 566 567 566 566 567 566 567 566 566 567 566 566	00DF 00E5 00E7 00E8 00E8 00E2 00F4 00F4 00F4 00F6 00F7 0102 0105 0107 0105 0107	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 88 0E 0000 R 86 E9 88 7C 04 88 44 0A 03 F8 03 C1 38 44 08 76 00 80 4C DF 60	апеаропаралананананананананананан Ж	JE RET ITE_1: CMP JE TEST JNZ MOV XCHG MOV XCHG MOV MOV ADD ADD CMP JNA DR	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START OPERATION - THE 3274 HILL TIM.         ; OUT AND TAKE THE 3278/3279 OFFLINE         HORD PTR [SI].BUFFER_ADDRESS42.0 ; WAS AN INPUT BUFFER         ; SPECIFILD FOR THIS MESSAGE TYPE?         KRITE_4       ; NO, INDIC/TE DATA DVERRUN         ISI].HESSAGE_FLAGS.HF_IN_USL ; IS THE INPUT BUFFER IN         ; WSE;       INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR.SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX,HOR_LENGTH ; PICK UP LENGTH OF THIS BLOCK         CH,CL         DI,KORD PTR [SI].BUFFER_ADDRESS ; PICK UP ADDRESS OF         ; INPUT DUFFER         AX,(SI].BUFFER_COUNT ; PICK UP RUMBER OF BYTES CURRENTL         ; IN THE INPUT BUFFER         DI.AX       ; CALCULATE WHERE THE NEW DATA GOES         AX,(SI].BUFFER_LENGTH ; DOES THE NUMBER OF BYTES RECEIVED         AX,(SI].BUFFER_LENGTH ; DOES THE NUMBER OF BYTES RECEIVED         AX,(SI].BUFFER_LENGTH ; DOES THE NUMBER OF BYTES RECEIVED         AX,(SI].BUFFER_LENGTH ; DOES THE NUM DATA FIT IN THE         ; BUFFER?         KRITE_2       ; YES, CONTINUE         [SI].MESDAGE_FLAGS.MF_DATA_IRUNC ; NO, THE DATA IS       ; EEINJ TRUMATED
554 555 557 558 561 563 564 565 566 566 566 566 566 571 573 576 578 577 577 578 577 578 577 578 577 577	00DF 00E5 00E7 00E8 00E8 00E2 00F4 00F4 00F4 00F6 00F7 0102 0105 0107 0105 0107	26: 80 3E 0002 R 07 74 01 C3 83 7C 06 00 74 45 F6 44 0F 20 75 3F 89 36 0055 R 26: 8B 0E 0000 R 86 E9 8B 7C 04 8B 44 0A 03 F8 03 C1 3B 44 08		JE RET ITE_1: CMP JE TEST JNZ MOV XCHG MOV XCHG MOV XCHG MOV XCHG MOV XCHG MOV XCHG MOV XCHG MOV	WRITE_1         ; INVALID MESSAGE HEADER - IGNORE THE         ; START OPERATION - THE 3274 HILL TIM         ; OUT AND TAKE THE 3276/3279 OFFLINE         WORD PTR [SI].BUFFER_ADDRESS+2.0 ; WAS AN INPUT BUFFER         ; SFECIFILD FOR THIS MESSAGE TYPE?         KRITE_4       ; NO, INDIC/TE DATA DVERRUN         ISI].HESSAGE_FLAGS.HF_IN_USL ; IS THE INPUT BUFFER IN         ; USE?         WRITE_4       ; YES, INDICATE DATA OVERRUN         FBA_RECEIVE_VECTOR,SI ; SAVE THE ADDRESS OF THE         ; INTERRUPT VECTOR FOR THIS MESSAGE         CX.HDR_LENGTH ; PICK UP LENGTH OF THIS BLOCK         CH.CL         DI,KORD PTR [SI].BUFFER_ADDRESS ; PICK UP ADDRESS OF         ; INPUT DUFFER         AX,[SI].BUFFER_COUNT ; PICK UP NUMBER OF BYTES CURRENTL         ; IN THE INPUT DUFFER         AX,[SI].BUFFER_LENGTH ; DOES THE NEW DATA GOES         AX,[SI].BUFFER_LENGTH ; DOES THE NEW DATA FIT IN THE         ; BUFFER?         WRITE_2       ; YES, CONTINUE         [SI].MESSAGE_FLAGS.MF_DATA_TRUNC ; NO, THE DATA IS

			4.	641,262
		87		88
		07		
589		2B 4C 0A	C SUB	CX.[SI].BUTFER_COUNT
590	0118	88 44 08	c nov	AX, ISI). BUFFER_LENGTH
591 592	011B		C C WRITE_2:	
593	0118	89 44 OA	C MOV	ISII.BUFFER_COUNT, AX ; SAVE THE UPDATED RECEIVE COUNT
594			C	
595 596	011E	88 54 06	C HOV	DX.WORD PTR [SI].BUFFER_ADDRESS+2 } MOVE THE SI.HDR DATA } DATA FROM THE FBA BUFFER TO THE
= 76 597	0121 0126	26: 8D 36 0004 R 1E	C LEA C PUSH	SI,HDR_DATA ; DATA FROM THE FBA BUFFER TO THE DS ; INPUT BUFFER
598	0127	06	C PUSH	ES
599	0128	1F	C POP	DS
600		8E C2	C HOV	ES,DX
601 602	0128	FC F3/ A4	C CLD C REP	MOVSB
603	012E	1F	C POP	DS
604		-	c	
605	012F		C WRITE_3:	THE ATTACK AND THE MERCICE
606 607	012F	E8 02A3 R	C CALL	SEND_OP_COMPLETE ; ACKNOWLEDGE THE MESSAGE
608	0132	63	C RET	PROCESS ANY OTHER PENDING INTERRUPTS
609			r	
610			C ; DATA	HAS BEEN RECEIVED WHEN THERE IS NO BUFFER AVAILABLE TO
611				IT - INDICATE DATA OVERRUN
612 613	0133		C C WRITE_4:	
614	0133	F6 44 10 80	C TEST	ISI].STATUS_FLAGS,SF_PROCESSING ; IS THE INTERRUPT
615			C	HANDLER BUSY?
616	0137	75 OF	C JHZ	WRITE_5 ; YES, REMEMBER OVERRUN CONDITION ; NO, CALL THE INTERRUPT HANDLER
617 618	0139	E8 02A3 R	C ; C CALL	SEND_OP_COMPLETE ; ACKNOWLEDGE THE MESSAGE
619	9124	LU VLNU K	C CALL	
620	013C	56	C PUSH	SI
621	013D	E8 0000 E	C CALL	PROC_INTERRUPTS ; PROCESS OTHER PENDING INTERRUPTS
622	0140	55	C FOP C STI	SI
623 624	0141	FB	C STI	
625	0142	B4 03	с ноч	AH, REASON_DATA_OVERRUN & INDICATE DATA OVERRUN
626	6144	ES CZAD R	C CALL	INFORM_APPL ; CALL THE INTERRUPT HANDLER
627	0147	C3	C RET	; PROCESS OTHER PENDING INTERRUPTS
628 629	0148		C C WRITE_5:	
630	0148	80 4C 10 4D	C OR	[SI].STATUS_FLAGS, SF_OVERRUN ; INDICATE DATA WAS LOST
631	014C	EB E1	C JI:P	WRITE_3 ; CLEAN UP AND EXIT
632			C NEDAT	E THE HEADER IN THE SCREEN CAPTURE BUFFER
633 634			C ; UPDAT	E THE READER IN THE SCREEN CAPTORE BOTTER
635	014E		C WRITE_SC:	
636	014E	26: F6 06 0003 R 80	C TEST	HDR_FLAGS, HDR_F_START ; IS THIS THE START OF A NEW
637		7/ 15		; BUFFER SECTION? WRITE_SC_1 ; NO, CONTINUATION OF PREVIOUS SECTION
638 639	0154	74 15	C JZ	WRITE_SC_1 ; NO, CONTINUATION OF PREVIOUS SECTION
640	0156	8B 54 04	C HOV	DX,WORD PTR ISILBUFFER_ADDRESS ; CALCULATE THE ADDRESS
641	0159	03 54 OA	C 40D	DX.(SI).BUFFER_COUNT ; OF THE LENGTH FIELD
642		89 16 0057 R	C NOV	WORD PTR SC_SECTION_LENGTH,DX ; SAVE THE ADDRESS OF THE DX,WORD PTR [SI].BUFFER_ADDRESS+2 ; BUFFER SECTION
643 644	0160 0163	88 54 06 89 16 0059 R	C HOV C HOV	WORD PTR SC_SECTION_LENGTH+2,DX ; LENGTH FIELD
645		83 44 0A 02	C ADD	ISIJ.BUFFER_COUNT, 2 ; POINT PAST THE LENGTH FIELD
646			C	
647	016B		C WRITE_SC_1: C FUSH	DS ; SAVE ADDRESS OF DATA AREA
64B 449	016B	1E C5 1E 0057 P	C LDS	BX,SC_SECTION_LENGTH ; PICK UP THE ADDRESS OF THE
649 650	0100	C5 1E 0057 R	C C	BUFFER SECTION LENGTH FIELD
651			C	
652	0170	26: F6 06 0003 R 80	C TEST	HDR_FLAGS,HDR_F_START ; IS THIS THE START OF A BUFFER
653		74 04	C JZ	; SECTION? WRITE_SC_2 ; NO, CONTINUE
654 655	0176	/- /4	C 32	
656	0178	C7 07 0000	C NOV	WORD PTR DS:[BX],0 ; YES, CLEAR THE LENGTH FIELD
657			C IDITE CO. A.	
658	0170		C WRITE_SC_2:	CV UND 1 FUCTU 1 DUFFED FEFTAL
659		26: 88 DE COOD R 86 E9	C MOV C XCHG	CX.HDF_LENGTH : BUFFER SECTION CHACL
660 661		01 OF	CCA 3	WORD PTR DS:[EX].CX
662	0100		C	
663	0185	1F	C FOP	DS ; RESTORE ADDRESS DF DATA AREA
664	<b>.</b>		C 190	WITE 1 ; MOVE THE DATA INTO THE INPUT BUFFEP
665 666	0186	E9 DOE8 R	C JKP	KRITE_1 ; MOVE THE DATA INTO THE INPUT EUFFEP
667		и		DE NDSPSOP2.ASH
668			С	
669			c c	
670 671			.C. }₩₩₩₩₩₩₩₩₩₩₩	4.1.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4
672			C ;#	*
673			C ;* Procedure	Name: EXECUTE_RECEIVED • *
674			C ;*	To process a Start Operation command from the 3274
675 676			C ;* Function: C ;*	which indicates Execute
677			C ;# ·	*
678			C ;* Input:	¥ 
679				IN_PROGRESS = 1 if a message is being sent to the 3274 *
680 681			C 3* F_READ C 1* F_WRITE	IN_PROGRESS = 1 if a message is being received from #
682			C 1*	the 3274 *
-				

				4.	641,262
		89		,	ý <b>90</b>
		07			
683			C		VECTOR = address of interrupt vector for message
684 685			C	;*	being received from the 3274 *
686			C C	;* ;* Output:	•
687			č	j# 000p000	*
688			С	;* If a mes	sage is being sent to the 3274 *
689 690			c		
691			C C	\$* F_READ	_IN_PROGRESS = 0 *
692			č		sage is being received from the 3274:
693			C	;* F_HRIT	E_IN_PROGRESS = 0
694 695			ç		upt handler is called with reason code indicating
696			C C	;* Mess ;* Ifam	age Received From 3274 essage is received while the interrupt handler is busy, #
697			č	j# it i	s called again with reason code indicating Data Overrun #
698			С	1¥	-
699 700			c	****	我名写不好 乳状球化含液体扩充液水液液 常然像就会这些保证是没有没有有多多多多多的是有些有多多多多的有些有些有多少的 建水子 化化化化化化
701	0189		C C	EXECUTE_RECEIV	FD:
702	0189	E8 02A3 R	č	CALL	SEND_OP_COMPLETE ; ACKNOWLEDGE THE START OPERATION
703			C		
704 705	0180	F6 06 0000 R 20	C C	TEST	<pre>FBL_STATE,F_WRITE_IN_PROGRESS ; IS A WRITE SEQUENCE IN ; PROGRESS?</pre>
706	0191	74 67	č	JZ	EXECUTE 5 ; NO, CONTINUE
707			č	•2	
708	0193	80 26 0000 R DF	C	CrtA.	FBI_STATE, NOT F_WRITE_IN_PROGRESS ; YES, THE WRITE
709			c		; SEQUENCE IS NOW COMPLETE SI,FBA_RECEIVE_VECTOR ; PICK UP THE ADDRESS OF THE
710 711	0198	88 36 0055 R	C C	MOV	; INTERRUPT VECTOR FOR THE LAST
712			c		BLOCK RECEIVED
713			С		·
714	019C	OB F6	ç	OR	SI,SI ; WAS THE LAST BLOCK RECEIVED ASSIGNED
715 716	0105	74 59	C C	17	; TO AN INTERRUPT VECTOR? EXECUTE 4 ; NO, THE DATA WAS DISCARDED
717	019E		c	JZ	
718	DIAO	80 7C 0E 04	С	CHP	[SI].MESSAGE_TYPE,HT_SET_SC_OPTIONS ; IS THIS A SCREEN
719			С		; CAPTURE HESSAGE?
720	0144	75 06	c	JNE	EXECUTE_2 ; NO, CONTINUE
721 722	01A6	BA 03E4	C C	MOV	DX,CCHMAND_PORT ; YES, INDICATE SCREEN CAPTURE IS IN
723	0149	B0 05	č	MOV	AL, SCREEN_CAPTURE ; PROGRESS
724	DIAB	EE	C	OUT	DX,AL
725			Ċ		
726	DIAC	F/	C C	EXECUTE_2:	CT .
727 728	01AC 01AD	E8 0000 E	č	PUSH	SI PROC_INTERRUPTS ; PROCESS OTHER PENDING INTERRUPTS
729	0180	5E	č	POP	SI
730	01B1	FB	С	STI	
731			C		
732		B4 01	C C	HOV	AH,REASON_MESSAGE_RECEIVED ; INDICATE MESSAGE RECEIVED Al,ISII.MESSAGE_FLAGS ; PICK UP MESSAGE FLAGS
733 734	01B4 01B7	8A 44 OF 24 80	c	GIA	AL, ME DATA TRUNC ; ISOLATE THE DATA TRUNCATED FLAG
735	0189	6B 4C DA	Ē	HOV	CX, ISI ]. EUFFER_COUNT ; PICK UP THE NUMBER OF BYTES
736			C		; IN THE MESSAGE
737	DIBC	C4 54 04	C	LES	DX, (SI). BUFFER_ADDRESS ; PICK UP THE ADDRESS OF THE
738 739			с с		; INPUT BUFFER
740	DIBF	C7 06 0055 R 0000	č	HOV	FBA_PECEIVE_VECTOR,0 ; INDICATE THE MESSAGE WAS
741			Ċ		; PROCESSED
742		E8 02C2 R	C	CALL	FERET CUFFER COUNTS ; RESET BUFFER COUNTERS
743 744	0108	80 4C OF 20	C C	<b>DR</b>	ICTUREDINGE_FLAGS, HF_IN_USE ; INDICATE HESSAGE BUFFER
745			č		
746	0100	E8 D2AD R	С	CALL	INFORM_APPL ; CALL THE INTERRUPT HANDLER TO PROCESS
747			С		THE MESSAGE
748	0107	DD 44 DE DE	c c		[SI].HESSAGE_FLAGS, NOT HF_IN_USE ; INDICATE MESSAGE
749 750	0104	80 64 OF DF	С С	AND	: BUFFER IS AVAILABLE
751	0103	83 7C 06 D0	C	CHP	[SI].BUFFER_ADDRESS+2,0 ; IS THERE A MESSAGE BUFFER?
752	0107	74 OF	С	JZ	EXECUTE 3 ; NO, CONTINUE
753	0109	80 7C 0E 04	ç	CMP	ISTI.HESSAGE_TYPE, HT_SET_SC_OPTIONS ; WAS THIS & SCREEN
754	0100	75 09	C C	JNE	; CAPTURE MESSAGE? EXECUTE_3 ; NO, CONTINUE
755 756	0100	75 09	c	JNE	-
757	01DF	E8 029A R	č	CALL	WAIT_FOR_POLL_ACK ; WAIT FOR THE PREVIOUS POLL REQUEST
758		•	С		TO BE ACKNOWLEDGED
759		BA 03E4	c	HOV	DX,COMMAND_PORT ; A SCREEN CAPTURE MESSAGE WAS JUST AL,SCREEN_CAPTURE_COMPLETE   PROCESSED AND THE BUFFER
760		B0 04	C C	NOV	DX,AL ; IS AVAILABLE - SCREEN CAPTURE
761 762	01E7		c	001	PROCESSING HUST BE COMPLETE
763			С		
764	01E8		С	EXECUTE_3:	The server at the ap eventuel . The work birt brockture
765	01E8	F6 44 10 40	C	TEST	[SI].STATUS_FLAGS,SF_OVERRUN ; WAS MORE DATA RECEIVED ; WHILE THE INTERRUPT HANDLER WAS
766			C C		; WHILE THE INTERRUPT HANDLER WAS ; BUSY?
767 768	0150	74 0B	c	JZ	EXECUTE 4 ; NO, CONTINUE
769		80 64 10 BF	č	AND	ISIJ.STATUS_FLAGS, NOT SF_OVERRUN ; YES, INFORM THE USER
770		84 03	С	HOV	AH, REASON DATA OVERRUN ; INDICATE DATA LOST
771		E8 02AD R	C	CALL	INFORM APPL I CALL THE INTERRUPT HANDLER AGAIN
772	01F7	EB EF	ç	JMP	EXECUTE_3 ; TEST FOR OVERRUN AGAIN
773 774	01F9		C C	EXECUTE_4:	
775	01F9 01F9	C3	c	RET	; PROCESS OTHER PENDING INTERSUPTS
776		-	· Č		

.

4,641,262

.

777	01FA		c	EXECUTE_5: TEST FBA_STATE,F_READ_IN_PROGRESS ; IS A READ SEQUENCE IN
778 779	01FA	F6 06 0000 R 40	C C	; PROGRESS?
760	01FF	74 F8	C	JZ EXECUTE_4 ; NO, IGNORE THE EXECUTE
781			c	OR FBA_PENDING_DP,F_EXECUTE ; YES, INDICATE EXECUTE WAS
782	0201	87 DE DOD1 R 20	C C	OR FBA_FENDING_OP,F_EXECUTE ; TES, INDICATE EXECUTE WAS ; RECEIVED
783 784	0206	80 26 0000 R BF	č	AND FBA_STATE, NOT F_READ_IN_PROGRESS ; THE READ SEQUENCE IS
785			C	; NOW COMPLETE
786	020B		c	RET ; PROCESS OTHER PENDING INTERPUPTS
787 788		)	C C	
789			č	· · · · · · · · · · · · · · · · · · ·
790			c	* 装装装饰水管盖装造装饰装饰盖制的小子的每个个部分,并不能能能够把某个工作,并不不能有处只能是不能要要做个需要要要要要要要要要要要要
791 792			C C	;* ;* Procedure Name: ABORT_E_RECEIVED
793			č	14
794			C	* Function: To process a Start Operation command from the 3274
795			C	# which indicates Abort Transmission
796 797			C C	;* ;* Input:
798			č	12
799			C	F READ IN PROGRESS = 1 if a message is being sent to the 3274
800			C C	
801 802			c	H RECEIVE VECTOR = address of interrupt vector for message
803			ċ	
804			C	
805 806			C C	
807			č	j* If a message is being sent to the 3274
808			C	;* F_ABORT_E = 1
809			C C	
810 811			c	;* If a message is being received from the 3274:
812			C	;* F_WRITE_IN_PROGRESS = 0
813			ç	
814 815			C C	
816			č	
817	02 <b>0C</b>		c	
818 819	D20C	E8 02A3 R	C	_
820	020F	F6 06 0000 R 20	č	TEST FBA STATE,F_WRITE_IN_PROGRESS ; IS A WRITE SEQUENCE IN
821			С	
822	0214	74 OD	C C	
823 824	0216	80 26 0000 R DF	с С	AND FBA STATE,NOT F_WRITE_IN_FROGRESS ; YES, CANCEL THE
825			č	: WRITE SEQUENCE
826	021B	88 36 0055 R	C	
827			0	: RECEIVED
828 829	021F	E8 02C2 R	Č	CALL RESET BUFFER COUNTS ; RESET THE BUFFER COUNTERS
830	D222		C	RET ; PROCESS OTHER PENDING INTERROPTS
831			<u> </u>	
832 833	0223	F6 06 0000 R 40	C C	
834			C	; PROGRESS?
835	0228	74 OA ·	0	
836 837	0224	80 0E 0001 R 10	C C	
838			č	TRANSMISSION RECEIVED
839	022F	80 26 0000 R BF	(	
84D 841			( (	
841 842	0234			ABODT F 2.
843	0234	C3	C	RET ; PROCESS OTHER PENDING INTERROPTS
844			(	
845 846			C C	
847				
848			c	*
849 850			с с	
851			0	
852			c	3*
853			0	· · · · ·
854 855			C	· · · · · · · · · · · · · · · · · · ·
856			0	
857			c	3* F_WRITE_IN_FROGRESS = 1 if a message is being received from
858 859			0	
860			0	
861			c	*
862			C	
863 864			0	
865			C C	The second second second second second
866			c	1 ( ) · · · · · · · · · · · · · · · · · ·
867			C	
868 869			с с	and all date received since receiving the about other operations
870			C	command indicating verify is discalded
871			c	
•				

## 

		93	94	
872			C	
873	0235		C ABORT_V_RECEIVED:	
874 875	0235	E8 02A3 R	C CALL SEND_OP_COMPLETE ; ACKNOWLEDGE THE START OP	
876	0238	F6 06 0000 R 20	C C TEST FBA_STATE,F_WRITE_IN_PROGRESS ; IS A WRITE SEQUENCE IN	я
877 -			C ; PROGRESS?	
878	0230	74 15	C JZ ABORT_V_1 ; NO, CONTINUE	
879 880	023F	88 36 0055 R	C MOV SI,FBA_RECEIVE_VECTOR ; PICK UP THE ADDRESS OF THE	
881		00 30 0033 K	C ; INTERRUPT VECTOR FOR THE LAST BLOW	CK
882			C ; RECEIVED	
883 884	0243	8B 44 0C	C MOV AX, [SI]. VERIFY COUNT ; SET THE NUMBER OF BYTES RECEIVE	20 5
885	0246	89 44 DA	C MOV ISII.BUFFER_COUNT,AX ; TO THE NUMBER OF VERIFIED BYTE:	-
886	0249	F6 44 DF 40	C TEST [SI].MESSAGE_FLAGS,MF_VERIFY_TRUNC ; WAS THE LAST VER	IFY
887			C ; PAST THE END OF THE BUFFER?	
888 889	024D 024F	75 16 80 64 DF 7F	C JNZ ABDRT_V_2 ; YES, THE DATA IS STILL TRUNCATED C AND ISIJ.MESSAGE_FLAGS.NOT MF_DATA_TRUNC ; NO, THE VERIFI	ED
890	0247	60 64 UF 77	C AND [SI].MESSAGE_FLAGS,NOT MF_DATA_TRUNC ; ND, HE VERIFIED C ; DATA ALL FITS IN THE BUFFER	
891	0253	C3	C RET ; PROCESS OTHER PENDING INTERRUPTS	
892 893	0254			
894	0254	F6 06 0000 R 40	C ABORT_V_1: C TEST FBA_STATE,F_READ_IN_PROGRESS ; IS A READ SEQUENCE IN	
895			C ; PROGRESS?	
896	0259	74 OA	C JZ ABORT_V_2 ; NO, IGNORE THE ABORT TO LAST VERIFY	
897 898	025B	80 OE 0001 R 08	C DR FBA_PENDING_DP,F_ABORT_V ; YES, INDICATE ABORT TO LAS	т
899	2620	55 82 9991 N 80	C FRA_PERGING_DF, TABORT_V, TES, INDICATE RESACE WIL	
900			C ; BE RETRANSMITTED	
901 902	0260	80 26 0000 R BF	C AND FBA_STATE, NOT F_READ_IN_PROGRESS ; THE READ SECUENCE C ; HAS TO BE RESTARTED	
902			C ; HAS TO BE RESTARTED	
904	0265		C ABORT_V_2:	
905	0265	C3	C RET ; PROCESS OTHER PENDING INTERRUPTS	
906 907			C	
905 905			с , с	
909			C ) ***********************************	÷¥
910				*
911 912			C ;* Procedure Name: ERROR_RECEIVED C ;*	*
913			C ;* Function: To process a Start Operation command from the 3274	*
914			C 1* which indicates Error	*
915 916				*
917			C ;* Input: C ;*	×
918			C :* F_READ_IN_FROGRESS = 1 if a message is being sent to the 3274	¥
919			C 3*	*
920 921			C ;* Output: C ;*	*
922			C 1* If a message is being sent to the 3274	*
923			C :* F_ERROR = 1	*
924 925			C ]*####################################	£##
926			C	
927	0266		C ERROR_RECEIVED:	
928 929	0266	E8 02A3 R	C CALL SEND_OP_COMPLETE ; ACKNOWLEDGE THE START OP	
930	0269	F6 06 0000 R 40	C TEST FBA_STATE,F_READ_IN_PROGRESS ; IS A READ SEQUENCE IN	
931			C ; PROGRESS?	
932 933	026E	74 OA	C JZ ERROR_1 ; NO, IGNORE THE ERROR	
934	0270	80 0E 0001 R 04	C OR FBA_PENDING_OP,F_ERROR ; YES, INDICATE ERROR RECEIVED	
935	0275		C AND FBA_STATE,NOT F_READ_IN_PROGRESS ; THE READ SEQUENCE J	(S
936 937			C ; NOW COMPLETE	
938	027A		C C ERROR_1:	
939	027A	сз •	C RET ; PROCESS OTHER PENDING INTERRUPTS	
940 941			с с .	
941			c	
943			C = 344xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	+
944				
945 546			C ;* Procedure Name: VERIFY_RECEIVED C ;*	
940			C ;* Function: To process a Start Operation command from the 3274	
948			C :* which indicates Verify	
949 950			C ;∦ C ;≭ Input:	
950			C ;*. Input: C ;*	
952			C ;* F_WRITE_IN_PROGRESS = 1 if a message is being received from	
953			C ;* the 3274	
954 955			C ;* RECEIVE_VECTOR = address of interrupt vector for message C ;* being received from the 3274	
956			C 1*	
957			C 3* Output:	
958 ·			C ;* C ;* If a message is being received from the 3274:	
959 960			C ;* If a message is being received from the 3274: C ;* the current data count is saved for use during a subsequent	
961			C ;* Abort To Last Verify	
962			C ;*	**
963 964			C ;************************************	

95

1057

96

VERIFY\_RECEIVED: 965 027B С SEND\_OP\_COMPLETE ; ACKNOWLEDGE THE START OP 0278 E8 0243 R C C C CALL 966 967 FBA\_STATE.F\_WRITE\_IN\_PROGRESS ; IS A WRITE SEQUENCE IN TEST 027E F6 06 0000 R 20 968 PROGRESS? 969 с с с ; PROGRESS? ; NO, IGNORE THE VERIFY COMMAND 970 971 JZ VERIFY 1 0283 74 14 SI,FBA\_RECEIVE\_VECTOR ; PICK UP THE ADDRESS OF THE C C C 972 0285 88 36 0055 R MOV INTERRUPT VECTOR FOR THE LAST BLOCK 973 974 AX, [SI]. BUFFER\_COUNT ; SAVE THE NUMBER OF VERIFIED [SI]. VEPIFY\_COUNT, AX ; DAT/ BYTES [SI].MESSAGE\_FLAGS, MF\_DATA\_THUNC ; WAS THE DATA ; TRUNCATEL ? 975 0289 88 44 0A C C MOV MOV 028C 89 44 0C 028F F6 44 0F 80 976 977 C TEST ; TRUNCATEL ? VERIFY\_1 ; NO, CONTINUE (SI).MESSAGE\_FLAGS,MF\_VERIFY\_TRUNC ; YES, INDICATE THE 978 C C 0293 74 04 JZ 979 980 0295 80 4C DF 40 OR ; VERIFIED DATA WAS TRUNCATED 981 С 982 VERIFY\_1: 983 0299 С ; PROCESS OTHER PENDING INTERRUPTS č RET 0299 C3 984 С 985 986 C C 987 "我长期发发来长年年年年末就要要放在这些实现在我们的,,不能不不不不不不不是在我这些不不要能是不要不是不能不是我们也不是不能不是不能不是不能不 988 С 989 1# ;\* Procedure Name: WAIT\_FOR\_POLL\_ACK 990 991 ;× Function: To wait for the previous poll request to be acknowledged by the 3274 j\* 992 993 ;# :# 994 995 ;\* Input: None 3# 996 997 ;\* Output: 998 ;\* C ;× AL and DX are unpredictable 999 1000 с с :\* ;\*\* 1001 с с 1002 AL,DX ; INDICATES THAT A POLL REQUEST IS AL,ST\_POLL\_ACTIVE ; NOT PENDING WAIT\_FOR\_POLL\_ACK WAIT\_FOR\_POLL\_ACK PROC NEAR 1003 0294 HOV 1004 Ċ 029A BA 03E4 с с 1005 029D EC AB 40 TEST 1006 1007 029E 02A0 75 F8 С 3N7 С 1008 ; RETURN TO CALLER 1009 Ċ RET DZAZ C3 WAIT FOR POLL\_ACK ENDP 1010 1011 с с 0243 」 像水波像像黑水灰起意地发发在水水灰在水水灰在灰灰的灰地的灰色的灰色的灰色的灰色的水头在水水水水水水水水水水水和浓水的水和浓水的水和浓水的水和浓水的 1012 С c c 1013 ;× ;\* Procedure Name: SEND\_OP\_COMPLETE 1014 1015 3 K С С С С ;\* Function: To send an Operation Complete reply to the 3274 1016 1017 ;\* ;# Input: None 1018 ;\* 1019 0000000 ;\* Output: 1020 ;\* 1021 AL and DX are unpredictable 1022 :\* 1023 "实法扩展资格在考虑实施有不成实现的实在在有关实在有关实现在在不是有关的实际的不可不可。"为大学不不少,并不不不可以是不是在不是在不是不是不是不是不是不可能。 1024 1025 SEND\_OP\_COMPLETE PROC NEAR 1026 02A3 с с с WAIT\_FOR\_POLL\_ACK ; WAIT FOR THE FREVIOUS POLL REQUEST ; TO BE ACKNOWLEDGED DX,COMMAND\_PORT ; SEND OP COMPLETE TO THE 3278/3279 CALL 1027 02A3 E8 029A R 1028 BA 03E4 С MOV 1029 02A6 AL, OP\_COMPLETE ; ADAPTER HOV 02A9 02AB B0 01 EE с с 1030 1031 OUT DX.AL ; RETURN TO CALLER 1032 D2AC C3 С RET С SEND\_OP\_COMPLETE ENDP 02AD 1033 1034 С c 1036 0000000 \$ 我家家老师我家家家家家家家家家家家家家家家家家家,有个个个个个个个人的正是有些家父亲家家家家家家家家家家家家家家家家家家家家家家家 1037 \* Procedure Name: INFORM\_APPL 1039 ;+ 1040 ;\*
;\* Function: To call a user supplied interrupt handler 1041 11 1042 C ;\* Input: 1043 с с 1044 ;× SI = address of interrupt vector ;\* 1045 1046 C C C ;× ;# Dutput: 1047 1048 ;\* All registers other than DS and SI are unpredictable 1049 C C :\* ; H 1050 \* 李老家家被拿家来我来来我这些我我我就没要你这个家族就是我是我有我不是这些我不能要我不能要不是这个要要我能够不能是我不能是我不是我 ¢ 1051 1052 C INFORM\_APPL PROC NEAR C 1053 02AD WORD PTR [SI].INTERRUPT\_VECTOR+2,0 ; IS THERE AN с с 1054 02AD 83 7C 02 00 CMP INTERRUPT HANDLER SPECIFIED? ; INTERRUPT HANDLER S INFORM APPL\_1 ; NO, RETURN TO CALLER 1055 0281 74 0E с с JZ 1056

1050       60 0. 10 00       C       DR       ISILSTADS_FLASS_PROCESSIME TABLATE THE INTERPUT INAULES INTERPUT NAULER         1050       6207       14       C       PATH       DS       I CALL THE LEFTS INTERPUT NAUDLER         1051       6207       14       C       PATH       DS       I CALL THE LEFTS INTERPUT NAUDLER         1051       6207       15       CALL ISILSTATUS_FLASS.PT PROCESSIME I DOTLATE THE       I NOTE         1051       1051       INTERPUT NAULER IS NOT WADE       PATH       I NOTE         1050       0201       C       PATH       I NOTE       I NOTE         1050       0201       C       PATH       I NOTE       I NOTE         1050       0201       C       PATH       I NOTE       I NOTE         1050       0201       INTOR_APPL_1       I RET       I NOTE       I NOTE         1050       0201       INTOR_APPL_1       I RET       I NOTE       I NOTE         1050       0201       INTOR_APPL_1       I NOTE       I NOTE       I NOTE         1050       0201       INTOR_APPL_1       I NOTE       I NOTE       I NOTE         1050       0201       I NOTE       I NOTE       I NOTE       I NOTE <td< th=""><th></th><th></th><th></th><th></th><th></th><th>97</th><th></th><th></th><th>.,</th><th></th><th>98</th></td<>						97			.,		98
1000       1 MADER 15 BURY         1000       0207 16         1000       0207 16         1000       0207 16         1000       0207 16         1000       0207 16         1000       0207 16         1000       0207 16         1000       0207 16         1000       0207 16         1000       0207 17         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       0201 10         1000       020	1058	02B3	80	40	<b>`</b> 10		с		08	ISTL STATUS FI	
164       CED 7 IF       C       PUSH DS       ; CALL THE USER'S INTERPOY HANGLEB         164       CED 6 IC       C       PUSH DS       ; CALL THE USER'S INTERPOY HANGLEB         165       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         165       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         165       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         165       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         165       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         165       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         166       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         167       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         167       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         167       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         167       CED 6 IC       C       POP DS       ; CALL THE USER'S INTERPOY HANGLEB         167       CED 6 IC       C	1059			16			С		UK	(511.51×105_11	; HANDLER IS BUSY
1855       0289       FF 1C       C       CALL       ISI. JAYTERPUT_VECTOR         000       050       050       050       1       MODELLET       INFORMATION         000       051       050       050       1       MODELLET       INFORMATION         000       051       051       1       MODELLET       INFORMATION       INFORMATION         0000       051       051       1       MITTERPERATION       INFORMATION       INFORMATION         0000       051       051       051       INFORMATION       INFORMATION       INFORMATION         0000       051       051       051       051       051       051       051         0000       0520       052       052       050       050       050       050       051       051       051       051       051       051       051       051	1060	02B7	1E						PUSH	DS	; CALL THE USER'S INTERRUPT HANDLER
1044       0280       55       C       POP       51         1045       0280       60       64       10       77       AD       ISILITATUS_FLOSSING STREEDT HANDES INDERATE THE INTERPORT SANDES STATUSY         1050       0200       60       64       10       77       C       AD       ISILITATUS_FLOSSING STREEDT HANDES INTERPORT SANDES STATUSY         1050       0200       C       C       INTORN_APPL_ENDP       INTORN_APPL_ENDP         1051       0200       C       C       INTORN_APPL_ENDP       INTORN_APPL_ENDP         1050       C       C       INTORN_APPL_ENDP       INTORN_APPL_ENDP         1051       C       INTORN_APPL_ENDP       INTORN_APPL_ENDP         1052       C       INTORN_APPL_ENDP       INTORN_APPL_ENDP         1053       C       INTORN_APPL_ENDP       INTORN_APPL_ENDP         1054       C       INTORN_APPL_ENDP       INTORN_APPL_ENDP         1055       C       INTORN_APPL_ENDP       INTORN_APPL_ENDP         1056       C       INTORN_APPL_ENDP       INTORNAPPL_ENDP         1057       SC       SC       INTORN_APPL_ENDP         1050       SC       SC       INTORN_APPL_ENDP         1050       SC	1062										
1066 1070       0750       00       1511.5TATUS_FLAGS.HOT SF processing 1_DEFLATE THE 	1064										TECTOR
1067       0280       60 4 6 10 7F       C       AD       ISILISATUS_LOSSING INDUCTS SHAT THE INDUCATION THE INDUCTS SHAT THE INT INTO SHAT THE INTERPORT SHAT THE INTERPORT SHAT THE INT	1065	02BC	1 F						POP		
1990       CC1       CS2	1067	026D	80	64	10	7F	С		AND	[SI].STATUS_FI	
1072       e2C1       C1	1068										; INTERRUPT HANDLER IS NOT BUST
1972       0202       DIFGRE_APPLEEDP         1973       C       Procedure Name: RESET_BUFFER_COUNTS         1974       C       Procedure Name: RESET_BUFFER_COUNTS         1975       C       Procedure Name: RESET_BUFFER_COUNTS         1976       C       Procedure Name: RESET_BUFFER_COUNTS         1977       C       Procedure Name: RESET_BUFFER_COUNTS         1987       C       Procedure Name: RESET_BUFFER_COUNTS         1987       C       Procedure Name: RESET_BUFFER_COUNTS         1987       C       POSE         1988       C       POSE         1989       CCC       C         1989       CCC       C         1989       CCC       C       POSE         1989 <td< td=""><td>1070</td><td></td><td>- 7</td><td></td><td></td><td></td><td></td><td>INFO</td><td></td><td></td><td>DETIMIN TO CALLER</td></td<>	1070		- 7					INFO			DETIMIN TO CALLER
1076       C         1077       C         1081       C         1081       C         1083       C         1084       C         1085       C         1086       C         1087       C         1088       C         1089       C         1080       C         1081       C         1082       C         1084       C         1085       C         1086       C         1086       C         1087       C         1088       C         1089       C         1080       C <td< td=""><td>1072</td><td></td><td>5</td><td></td><td></td><td></td><td>С</td><td>INFO</td><td></td><td>P</td><td>, REIGRA TO CALLER</td></td<>	1072		5				С	INFO		P	, REIGRA TO CALLER
1076       C         1076       C         1077       C         1078       C         1079       C         1079       C         1079       C         1079       C         1070       C         1071       C         1072       C         1073       C         1074       C         1075       C         1081       C         1082       C         1083       C         1084       C         1085       C         1086       C         1087       C         1088       C         1089       C         1080       C         1081       C         1082       C         1083       C         1084       C         1085       C         1086       C         1087       C         1088       C         1089       C         1080       C         1081       C         1081       C <td< td=""><td>1073 1074</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>	1073 1074										
1077       C       #       Procedure Hama: RESET_BUFFEP_COUNTS         1080       C       #       Function: To rest the receive data counts and status flags for an interrupt vector         1081       C       #       Function: To rest the receive data counts and status flags for an interrupt vector         1085       C       #       Function: To rest the receive data counts and status flags for an interrupt vector         1086       C       #       SI = address of interrupt vector         1086       C       #       SI = address of interrupt vector         1086       C       #       SI = address of interrupt vector         1086       C       #       SI = address of interrupt vector         1087       CCC 50       C       RESIL_DUFFE_COUNT AX; rests interfies Not Flags         1088       CCC 50       C       MOV       ISI.LUTPE_COUNT AX; rests interfies Not Flags         1089       CCC 50       C       MOV       ISI.LUTPE_COUNT AX; rests interfies Not Flags         1089       CCC 60       C       MOV       ISI.LUTPE_COUNT AX; rests interfies Not Flags         1080       CCC 60       C       REST_BUFFE_COUNT AX; rests interfies Not Flags         1081       CCD 1       NDSP_SOF INDP       ISI.EXECEPT AX         1082	1075						с				2—2 222 14 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2
1079       C       Function: To rest the receive data counts and status flags for an interrupt vector         1083       C       Imput:         1084       C       Imput:         1085       C       Imput:         1086       C       Imput:         1086       C       Imput:         1086       C       Imput:         1086       C       Imput:         1087       Octo       Dutput: None         1088       C       Imput:       Imput:         1089       Octo       C       POS         1089       Octo       C       POS       Ax         1089       Octo       C       POS       Imput:         1089       Octo       C       POS       Imput:       Imput:         1089       Octo       C       POS       Imput:       Imput:       Imput:         1089       Octo       C       POS       Imput:       Imput: <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>**********</td> <td></td> <td>***************************************</td>									**********		***************************************
1080       C   # Function: To reset the receive data counts and status flags for         1081       C   # Function: To reset the receive data counts and status flags for         1083       C   # Function: To reset the receive data counts and status flags for         1084       C   # SI = address of interrupt vector         1085       C   # SI = address of interrupt vector         1086       C   # SI = address of interrupt vector         1086       C   # SI = address of interrupt vector         1087       DCC:       C   # SI = address of interrupt vector         1088       C   # SI = address of interrupt vector         1089       DCC:       C   # SI = address of interrupt vector         1089       DCC:       C   # SI = address of interrupt vector         1089       DCC:       C   # SI = address of interrupt vector         1089       DCC:       C   # Dutpt:         1089       DCC:       C   # Dutpt:         1089       DCC:       C   # Dutpt:         1080       DCC:       REST_BUFFE_COUNT.X: & REST NUMER of PATES NECEVED         1080       C   # Intervent X: REST NUMER of PATES NECEVED         1080       C   # Dutpt:       I # RETURN TO CALLER         1080       C   # Intervent X: REST NUMER of PATES NECEVED         1081       C   # Forceion:									Procedure 1	lame: RESET_BUFF	ER_COUNTS
1065       C       i Input:         1066       SI = address of interrupt vector         1066       C       i Dutput:         1067       C       i Dutput:         1068       C       i Dutput:         1069       CCC       C         1060       C       NOV         1061       C       NOV         1061       C       NDSP_SOP ENDP         1061       C       NDSP_SOP IND         1061       C       NDSP_SOP IND         1061       C       NDSP_SOP IND         1061       C	1080						C	;×			
1065       C       i       SI = address of interrupt vector         1066       C       Dutput: Home         1067       C       Dutput: Home         1068       C       FEST_BUFFER_CONTS PROC HEAR         1069       CCC SS C       C         1060       C       C         1060       C       C         1061       C       REST_BUFFER_CONTS FROC HEAR         1060       C       C         1061       C       NOV         1062       C       MOV         1063       C       REST_BUFFER_CONTS FROC HEAR         1064       C       NOV         1065       C       NOV         1066       C       NOV         1067       C       NOSP_AST PACONTACKS FROC HEAR         1068										in interrupt vec	tor
1055       C       i       S.I. # address of interrupt vector         1057       Dutput: None         1059       02C2       C       PESET_BUFFER_CONTS PROC MAR         1059       02C2       C       REST_BUFFER_CONTS PROC MAR         1059       02C2       C       C       XI, XI, YEETY CONTERS IN JEED OF DYES RECEIVED         1059       02C2 60       C       NO       ISI, NEETY CONT.XI   REST NUMBER OF DYES RECEIVED         1059       02C2 64 44 0C       C       NOV       ISI, NEETASAE_FLASS, I REST NUMER OF DYES RECEIVED         1059       02C2 64 44 0C       C       NOV       ISI, NEETASAE, FLASS, I REST NUMER OF DYES RECEIVED         1059       02C1       C       RET       RET       RET         1050       02C1       C       REST_BUFFER_CONTS ENDP       IRETURN TO CALLER         1050       02C1       NDSP_SOP ENDP       IRETURN TO CALLER         1051       SUBTL Procedure Name: NOSP_API       I         1051       INSP_SOP ENDP       IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII									Input:		
107       C       i* Output: None         1080       C       i************************************	1085						С	;#	SI = əddr	ess of interrup	t vector
1086       C         1087       05C2         1089       05C2         1090       05C3         1090       05C4         1090       05C5         1090       05C5         1090       05C6         05C6       64         05C7       05C7         1090       05C8         1090       05C8         1090       05C8         1090       05C9         05C9       05C3         05C9									Output: Nor	ne .	
1090       0CC       C       RESET_BUFFE_COUNTS FROC NEAR         1090       0CC 50       C       PUSH_AX       IET COUNTERS TO ZERO         1090       0CC 53       C       XX       IET COUNTERS TO ZERO         1090       0CC 53       C       XX       IET COUNTERS TO ZERO         1090       0CC 544 0C       C       MOV       ISIJ_MEDTEY_COUNT, AX : REST NUMBER OF BYTES VERIFIED         1090       0CC 55       C       HOV       ISIJ_MEDTEY_COUNT, AX : REST NUMBER OF BYTES VERIFIED         1090       0CC 55       C       HOV       ISIJ_MEDTEY_COUNT, AX : REST NUMBER OF BYTES VERIFIED         1090       0CC 55       C       HOV       ISIJ_MEDTEY_COUNT, AX : REST NUMBER OF BYTES VERIFIED         1090       0CD 1       HOV       ISIJ_MEDTEY_COUNT, AX : REST NUMBER OF BYTES VERIFIED         1090       C       C       HOV       ISIJ_MEDTEY_COUNT, AX : REST NUMBER OF BYTES VERIFIED         1090       C       C       HOV       ISIJ_MEDTEY_COUNT, AX : REST NUMBER OF BYTES VERIFIED         1090       C       C       HOV       ISIJ_MEDTEY_COUNT, AX : REST NUMBER OF BYTES VERIFIED         1001       C       NOSP_SOP ENDP       ISIJ_MEDTEY       ISIJ_MEDTEY         1001       Fraction: To process a 3278/3279 Atta							с		•		<b>王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王</b>
1092         02C2         50         C         XX         XX         SET COUNTERS TO ZERO           1094         02C5         69         44 0.0         C         NOV         ISIL BUFFER COUNT, XX : BEST I MURRER OF BYTES RECEIVED           1095         02C5         69         44 0.0         C         NOV         ISIL BUFFER COUNT, XX : BEST I MURRER OF BYTES RECEIVED           1095         02C6         69         44 0.0         C         NOV         ISIL BUFFER COUNT, XX : BEST I MURRER OF BYTES RECEIVED           1097         02C7         54         40 OF DD         NOV         ISIL BUFFER COUNTS ENDP           1099         02D1         NDSP_SOP ENDP         :         RESET MURRER OF BYTES         RESET MURRER OF BYTES           1006         C         RESET_BUFFER_COUNTS ENDP         :         RESET_BUFFER_COUNTS ENDP           1007         C         RESET_BUFFER_COUNTS ENDP         :         :           1008         C         RESET_BUFFER_COUNTS ENDP         :         :           1009         C         RESET_BUFFER_COUNTS ENDP         :         :           1009         C         RESET_BUFFER_COUNTS ENDP         :         :           1009         C         RESET_BUFFER COUNTS ENDP         :	1070						С	; ***	*********		*****
1095       02C3       33       C0       C       X0R       X1,XX       1 SET CONTREP TO ZERO         1096       02C6       89 44 00       C       HOV       ISI.BERFE, LONT, XX       RESET NUMBER OF BYTES PECEDIVED         1096       02C6       89 44 00       C       HOV       ISI.BERFE, LONT, XX       RESET NUMBER OF BYTES PECEDIVED         1096       02C6       64 40 00       C       HOV       ISI.BERGE, LASSO       REST HERSAGE         1096       02C1       C       HOV       ISI.BERGE, LASSO       REST HERSAGE       REST HERSAGE         1097       02C1       NDSP_SOP ENDP       ISI.BERGE, LASSO       REST HERSAGE       REST HERSAGE         1106       C       SUBTIL Process a 3276/3279 Attachment function request       ISI.BERGE       ISI.BERGE         1106       ISI.BERGE       SUBTIL Process a 3278/3279 Attachment function request       ISI.BERGE         1107       ISI.BERGE       ISI.BERGE       ISI.BERGE       ISI.BERGE         1108       ISI.BERGE       ISI.BERGE       ISI.BERGE       ISI.BERGE         1107       ISI.BERGE       ISI.BERGE       ISI.BERGE       ISI.BERGE         1108       ISI.BERGE       ISI.BERGE       ISI.BERGE       ISI.BERGE			50					RESE			
1095         02C6         89 4 0C         C         HOV         ISI.VERITY_COUNT,AX : REST HIESSAF VERIFIED           1096         02CF 56         56         C         POP         AX         ; RETURN TO CALLER           1090         02CF 56         C         POP         AX         ; RETURN TO CALLER           1091         02CF 56         C         POP         AX         ; RETURN TO CALLER           1091         C         REST_BUFFER_COUNTS ENDP	1093	02C3	33	C0			C		XOR	AX.AX	; SET COUNTERS TO ZERO
1097       02CF       58       C       POP       AX         1096       02D1       C       RET       ; RETURN TO CALLER         1000       C       NDSP_SOP ENDP         1101       SUBTL       Process a 3278/3279 Attachment function request         1105       SUBTL       Process a 3278/3279 Attachment function request         1107       Image: NDSP_SOP ENDP         1108       SUBTL       Process a 3278/3279 Attachment function request         1109       Image: NDSP_SOP ENDP         1100       Image: NDSP_SOP ENDP         1100       Image: NDSP_SOP ENDP         1101       Image: NDSP_SOP ENDP         1109       Image: NDSP_SOP ENDP         1100       Image: NDSP_SOP ENDP         1110       Image: NDSP_SOP ENDP         1110       Image: NDSP_SOP ENDP         1110       Image: NDSP_SOP ENDP         1111       Image:										ISI].VERIFY_CC	UNT, AX ; RESET NUMBER OF BYTES VERIFIED
1096       020       C3       C       RET       ; RETURN TO CALLER         1099       0201       C       RESET_BUFFER_COUNTS ENDP         1001       C       NDSP_SOP ENDP         1005       C201       NDSP_SOP ENDP         1006       SUBTL       Process = 3278/3279 Attachment function request         1007       Impose Substructure Hame: NDSP_AFI         1008       Impose Substructure Hame: NDSP_AFI         1019       Impose Substructure Hame: NDSP_AFI         1010       Impose Substructure Hame: NDSP_AFI         1011       Impose Substructure Hame: NDSP_AFI         1011       Impose Substructure Hame: NDSP_AFI         1011       Impose Substructure Hame: NDSP_AFI         1018       Impose Substructure Hame: NDSP_AFI         1019       Impose Substructure Hame: NDSP_AFI         1011       Impose Substructure Hame: NDSP_AFI         1011       Impose Substructure Hame: NDSP_AFI         1011       Impose Substructure Hame: NDSP_AFI         1019       Impose Substructure Hame: NDSP_AFI         1011       Impose Substructure Hame: NDSP_AFI         1019       Impose Substructure Hame: NDSP_AFI         1011       Impose Substructure Hame: NDSP_AFI         1012       Impose Substr					OF	00					LAGS,0 ; RESET MESSAGE FLAGS
1100       C         1101       C         1102       CC         1103       CCD1         1104       SUBTL Process a 3278/3279 Attachment function request         1105       SUBTL Process a 3278/3279 Attachment function request         1106       Import         1111       Import         1112       Imput:         1113       Imput:         1114       Imput:         1115       Imput:         1116       Imput:         1118       Imput:         1119       Imput:         1119       Imput:         1119       Imput:         1110       Imput:         1111       Imput:         1112       Imput:         1119       Imput:         1110       Imput:         1111       Imput:         1111       Imput:         1111       Imput:         1111       Imput:         1111       Imput:         1111       Imput:         1112       Imput:         1121       Imput:         1122       Imput:         1123       Imput:	1098	02D0					C		RET		; RETURN TO CALLER
1102     DZD1     NDSP_SOP ENDP       1105     SUBTL Process a 3276/3279 Attachment function request       1106     Image: Procedure Hame: NDSP_API       1107     Image: Procedure Hame: NDSP_API       1108     Image: Procedure Hame: NDSP_API       1119     Image: Procedure Hame: NDSP_API       1111     Image: Procedure Hame: NDSP_API       1111     Image: Procedure Hame: NDSP_API       1112     Image: Procedure Hame: NDSP_API       1114     Image: Procedure Hame: NDSP_API       1115     Image: Procedure Hame: NDSP_API       1116     Image: Procedure Hame: NDSP_API       1114     Image: Procedure Hame: NDSP_API       1115     Image: Procedure Hame: NDSP_API       1116     Image: Process A 3278/3279 Attachment function request       1117     Image: Process A 3278/3279 Attachment Function Request       1121     Image: Process A 3278/3279 Attachment Function Request       1122     Image: Process A 3278/3279 Attachment Function Request       1131     Image: Process A 3278/3279 Attachment Function Request       1132     Image: Process A 3278/3279 Attachment Function Request       1133     Image: Process A 3278/3279 Attachment Function Request       1133     Image: Process A 3278/3279 Attachment Function Request       1133     Image: Process A 3278/3279 Attachment Function Request       1134 </td <td></td> <td>0201</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>RESE</td> <td>T_BUFFER_CO</td> <td>JUNIS ENUP</td> <td>· ·</td>		0201						RESE	T_BUFFER_CO	JUNIS ENUP	· ·
103       DCD1       NDSP_SOP ENDP         106       SUBITL Process a 3278/3279 Attachment function request         107       IIII         108       IIIII         109       IIIIII         100       IIIIIIIIII         110       IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII							С				
106       SUBTL Process a 3278/3279 Attachment function request         107       108         108       109         109       11         100       11         110       11         1110       11         1111       11         1112       11         1111       11         1112       11         1113       11         1114       11         1115       11         1116       11         1117       11         1118       11         1117       11         1118       11         1119       11         1111       11         1111       11         1111       11         1111       11         1111       11         1111       11         1111       11         1112       11         1112       11         1112       11         1112       11         1121       11         1122       11         1123       11         1124       11	1103	02D1						NDSP	SOP ENDP		
1107         1108         1109         1110         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1111         1112         1112 <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>SUBTTL</td><td>Process a 3278</td><td>/3279 Attachment function request</td></t<>									SUBTTL	Process a 3278	/3279 Attachment function request
1108       image: NDSP_API         1110       image: NDSP_API         1111       image: NDSP_API         1112       image: NDSP_API         1112       image: NDSP_API         1112       image: NDSP_API         1120       image: NDSP_API         1121       image: NDSP_API         1122       image: NDSP_API         1123       image: NDSP_API         1124       image: NDSP_API         1125       image: NDSP_API         1126       image: NDSP_API         1127       image: NDSP_API         1128       image: NDSP_API         1129       image: NDSP_API											
1110       IP Procedure Hame: NDSP_API         1111       IP Function: To process a 3278/3279 Attachment function request         1113       IP function: To process a 3278/3279 Attachment function request         1114       IP function: To process a 3278/3279 Attachment function request         1115       IP function: To process a 3278/3279 Attachment function request         1116       IP function code         1117       IP function failed         1120       IP function failed         1121       IP function failed         1122       IP function failed         1123       IP function failed         1124       IP function failed         1125       IP function failed         1126       IP function failed         1127       IP function failed         1128       IP function failed         1129       IP function failed         1120       IP function failed         1131       IP function failed         1132       IP function failed         1133       IP function failed         1134       IP function failed         1135       IP function failed         1136       IP function failed         1137       IP function failed         113	1108								******	**************	-
iiii       iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii								; <del>*</del>	Procedure I	lame: NDSP_API	*
1113       ;*       Input:         1114       ;*       Input:         1115       ;*       AH = function code         1116       ;*       Output:         1119       ;*       Output:         1120       ;*       CF = 0 if function was processed successfully         1121       ;*       1 if function failed         1122       ;*       AL = return code if function failed         1123       ;*       AL = return code if function failed         1124       ;*       Include NDSPAPI1.ASH         1125       ;*       Process A 3278/3279 Attachment Function Request         1131       C       ;*         1132       C       SIT         1133       C       ;*         1134       C       ;*         1135       C       pUSH BX       ; SAVE ALL REGISTERS EXCEPT AX         1136       C       pUSH DS       ;         1137       C201 FB       C       PUSH SI         1138       C202 53       C       PUSH SI         1139       C203 51       C       PUSH DS         1140       0205 66       C       PUSH DS         1141       0205 66       <								;* ;*	Function:	To process a 32	78/3279 Attachment function request
1115       i*       AH = function code         1117       i*       Output:         1118       i*       Output:         1119       i*       CF = 0 if function was processed successfully         1120       i*       I if function failed         1121       i*       AL = return code if function failed         1122       i*       AL = return code if function failed         1126       i*       i*         1126       i*       i*         1126       i*       i*         1127       0201       NDSP_API PROC FAR         1130       C       i*         1131       C       i*         1132       i*       Process A 3278/3279 Attachment Function Request         1133       C       i*         1134       C       push bx       i SAVE ALL REGISTERS EXCEPT AX         1135       C       PUSH bX       i SAVE ALL REGISTERS EXCEPT AX         1136       C       PUSH bX       i SAVE INPUT PARAMETER IN IX         1140       0205 56       C       PUSH bX       i SAVE INPUT PARAMETER IN IX         1141       0205 56       C       PUSH bS       i SAVE INPUT PARAMETER IN IX         1142	1113							;*			
1116       i* AH = function code         1117       i*         1118       i* Output:         1119       i*         1120       i*         1121       i*         1122       i*         1123       i*         1124       i*         1125       i*         1126       i*         1127       02D1         NDSP_API PROC FAR         1129       c         1120       i*         1121       i*         1125       i*         1126       i*         1127       02D1         NDSP_API PROC FAR         1130       c         1131       c         1132       c         1133       c         1134       r*         1135       c         1136       c         1137       02D1 FB         02D2 F33       C         02D4 F32       C         02D5 55       c         02D6 57       C         02D7 1E       C         02D6 57       C         02D6 677       C											*
1116       1* Dutput:         1119       **         1120       **         1121       **         1122       1 if function was processed successfully         1121       **         1122       **         1123       **         1124       **         1125       **         1126       **         1127       02D1         NDSP_API PROC FAR         1126       **         1127       02D1         NDSP_API PROC FAR         1130       C         1131       C         1132       C         1133       C         1134       C         1135       C         1136       C         1137       02D1 FB         C       STI         1138       C2D2 53         C       PUSH BX       ; SAVE ALL REGISTERS EXCEPT AX         1139       02D3 51       C         1140       02D6 57       C         1139       02D7 1E       C         1140       02D6 677       C         1141       02D8 66       C									AH = fun	ction code	
1120       ;*       CF = 0 if function Nas processed successfully         1121       ;*       1 if function failed         1122       ;*       AL = return code if function failed         1124       ;*       AL = return code if function failed         1125       ;*       NDSP_API PROC FAR         1126       C       INCLUDE NDSPAPI1.ASH         1127       02D1       NDSP_API PROC FAR         1128       C       INCLUDE NDSPAPI1.ASH         1130       C       ;************************************	1118							1#	Output:		*
1121       i* 1 if function failed         1122       i* AL = return code if function failed         1124       i* AL = return code if function failed         1125       i************************************									CF = 0 i	f function was	processed successfully *
1123       ;* AL = return code if function failed         1124       ;*         1125       ;*         1126       ;*         1127       0201         1128       ;*         1129       C         1130       C         1131       C;*         1132       ;*         1133       C;*         1134       C;*         1135       ;*         1136       C;*         1137       C201 FB         C       STI         1136       C         1137       C201 FB         C       STI         1138       C202 53         C       PUSH         1139       0202 53         C       PUSH         1140       0204 52         C       PUSH         1141       0205 56         C       PUSH         1142       0206 57         C       PUSH         1144       0209 60         1145       C         1146       0209 8C DB         C       MOV       SLSUS         1146       0208 8E R <td>1121</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>•</td> <td></td> <td></td> <td>EG *</td>	1121							•			EG *
1125       ;************************************	1123							;*			ction Tailed *
1126       NDSP_API PROC FAR         1129       C         1129       C         1130       C         1131       C         1132       C         1133       C         1134       C         1135       C         1136       C         1137       C2D1 FB         1138       C2D2 53         1139       C         1130       C         1137       O2D1 FB         1138       C2D2 53         1139       C         1140       D2D4 52         C       PUSH         1141       D2D5 56         1142       C2D6 57         1144       O2D6 57         1144       O2D8 06         C       PUSH         1143       O2D9 8C DB         C       MOV       SI,BIOSDATA         1144       O2D6 50         C       PUSH         ANDV       SI,BIOSDATA         1144       O2D6 50         C       PUSH         C       HOV         SI       SILICY FAR         C       HOV <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>;* ;**</td> <td>****</td> <td>***</td> <td>***************</td>								;* ;**	****	***	***************
1128       INCLUDE NDSPAPI1.ASH         1130       INCLUDE NDSPAPI1.ASH         1131       INCLUDE NDSPAPI1.ASH         1131       INCLUDE NDSPAPI1.ASH         1131       INCLUDE NDSPAPI1.ASH         1132       INCLUDE NDSPAPI1.ASH         1133       INCLUDE NDSPAPI1.ASH         1134       INCLUDE NDSPAPI1.ASH         1135       INCLUDE NDSPAPI1.ASH         1136       INCLUDE NDSPAPI1.ASH         1137       O2D1 FB       INCLUDE NDSPAPI1.ASH         1135       INCLUDE NDSPAPI1.ASH       INCLUDE NDSPAPI1.ASH         1136       O2D1 FB       INCLUDE NDSPAPI1.ASH         1137       O2D1 FB       INCLUDE NDSPAPI1.ASH       INCLUDE NDSPAPI1.ASH         1140       O2D5 60       INC       PUSH DS		0201						NDS	P API PROC	FAR	
1130       C         1131       C         1132       C         1133       C         1134       C         1135       C         1136       C         1137       0201         1138       C         1134       C         1135       C         1136       C         1137       0201         1138       C202         1139       0202         0201       FB         C       PUSH         BX       ; SAVE ALL REGISTERS EXCEPT AX         1139       0203         0204       52         C       PUSH         1140       0204         0205       56         C       PUSH         1141       0205         1142       0206         C       PUSH         1143       0207         1144       0208         0208       06         C       PUSH         1144       0208         0209       8C DB         C       MOV         1145       C <tr< td=""><td>1)28</td><td></td><td></td><td></td><td></td><td></td><td>-</td><td></td><td>-</td><td></td><td></td></tr<>	1)28						-		-		
1132       C       ;*         1133       C       ;*         1134       C       ;*         1135       C       ;*         1136       C       ;*         1137       0201 FB       C       STI       ; Allow HARDWARE INTERRUPTS         1136       C       STI       ; Allow HARDWARE INTERRUPTS         1137       0201 FB       C       STI       ; Allow HARDWARE INTERRUPTS         1138       0202 53       C       PUSH       BX       ; SAVE ALL REGISTERS EXCEPT AX         1139       0203 51       C       PUSH       DX         1140       0204 52       C       PUSH DX         1141       0205 56       C       PUSH DX         1142       0206 .57       C       PUSH DS         1144       0208 06       C       PUSH ES         1145       C       MOV       SI.BIOSDATA       ; ESTABLISH ADDRESSABILITY TO WORK AREA         1149       020E 8E DE       C       MOV       SI.BIOSDATA       ; USE THE FUNCTION CODE IN (AH) TO FOUTE         1150       C       PUSH AX       INDV       SI.SI       INTY TO WORK AREA         1150       D2E1 8A C4       C <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<>							_				
1133       C ;* Process A 3278/3279 Attachment Function Request         1134       C ;*         1135       C ;*         1136       C ;*         1137       0201 FB         1138       C 201 FB         1138       C 201 FB         1138       C 201 FB         1139       0202 53         1139       0203 51         1140       0204 52         1141       0205 56         1142       C PUSH DX         1143       0207 1E         1144       0208 06         1145       C         1146       0209 8C DB         1147       C         1148       0203 BE R         1149       0202 68 DE         1149       0202 60         1149       0208 8E R         1149       0208 8E R         1149       0208 6E DE         1150       C         1151       0220 50         1151       0220 50         1151       0220 50         1151       0220 50         1151       0220 50         1151       0221 50         1152       0221 8A C4											
1135       C       ;************************************	1133						С	;*			
1136       C       STI       ; ALLOW HARDWARE INTERRUPTS         1137       0201       FB       C       STI       ; ALLOW HARDWARE INTERRUPTS         1138       0202       53       C       PUSH       BX       ; SAVE ALL REGISTERS EXCEPT AX         1139       0203       51       C       PUSH       DX       ;         1140       0204       52       C       PUSH       DX       ;         1141       0205       56       C       PUSH       DX       ;         1142       0206       57       C       PUSH       DI       ;         1143       0207       1E       C       PUSH       DS       ;       ;         1144       0208       06       C       PUSH       ES       ;       ;       SAVE INPUT PARAMETER IN IX         1145       C								;* ;**	****	****	<b>家生我和我能够做这些我就没能能能能能能能要我的我们的不能不能不能不能不能不能要能做的。</b>
1137       0201       10       C       PUSH       BX       ; SAVE ALL REGISTERS EXCEPT AX         1138       0203       51       C       PUSH       BX       ; SAVE ALL REGISTERS EXCEPT AX         1139       0203       51       C       PUSH       CX         1140       0204       52       C       PUSH       DX         1141       0205       56       C       PUSH       SI         1142       0206       57       C       PUSH       SI         1144       0207       1E       C       PUSH       ES         1145       C       PUSH       ES       Itsi       Itsi         1146       0208       06       C       PUSH       ES         1146       0209       8C DB       C       MOV       BX,DS       ; SAVE INPUT PARAMETER IN ItX         1147       C       MOV       SI,BIOSDATA       ; ESTABLISH ADDRESSABILITY TO WORK AREA         1148       020B       BE R       C       MOV       SI,SI         1150       C       Itsi       020E       50       C       PUSH       AX         1152       02E1       8A C4       C       HOV </td <td>1136</td> <td></td> <td></td> <td>2</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>ALLOW HARDWARE INTERRUPTS</td>	1136			2							ALLOW HARDWARE INTERRUPTS
1140       0204       52       C       PUSH       DX         1141       0205       56       C       PUSH       SI         1142       0206       57       C       PUSH       DI         1143       0207       1E       C       PUSH       DS         1144       0208       06       C       PUSH       DS         1144       0208       06       C       PUSH       ES         1145       C       C       1145       C         1146       0208       8C       DB       C       MOV       BX,DS       ; SAVE INPUT PARAMETER IN BX         1147       C       C       1147       C       C       1147       C         1148       020B       BE       R       C       MOV       SI,BIOSDATA       ; ESTABLISH ADDRESSABILITY TO WORK AREA         1149       020E       8E       DE       C       MOV       DS,SI       DS         1150       C       C       PUSH       AX       ; USE THE FUNCTION CODE IN (AH) TO ROUTE         1152       02E1       8A       C       HOV       AL,AH       ; CONTEDU TO THE PROPER PRCCESSING		02D2	53	3			C		PUSH		; SAVE ALL REGISTERS EXCEPT AX
1141       02D5       56       C       PUSH       SI         1142       02D6       57       C       PUSH       DI         1143       02D7       1E       C       PUSH       DS         1144       02D8       06       C       PUSH       ES         1144       02D8       06       C       PUSH       ES         1145       C       C       MOV       BX,DS       ; SAVE INPUT PARAMETER IN BX         1145       C       C       MOV       BX,DS       ; SAVE INPUT PARAMETER IN BX         1146       02D8       BE      R       C       MOV       SI,BIOSDATA       ; ESTABLISH ADDRESSABILITY TO NORK AREA         1149       02DE       6E       DE       C       MOV       DS,SI         1150       C       C       PUSH       AX       ; USE THE FUNCTION CODE IN (AH) TO ROUTE         1152       02E1       8A       C4       C       MOV       AL,AH       ; USE THE FUNCTION CODE IN (AH) TO ROUTE											
1143       02D7       1E       C       PUSH       DS         1144       02D8       06       C       PUSH       ES         1144       02D8       06       C       PUSH       ES         1145       C       MOV       BX.DS       ; SAVE INPUT PARAMETER IN NX         1146       02D9       8C DB       C       MOV       BX.DS       ; ESTABLISH ADDRESSABILITY TO WORK AREA         1147       C       1147       C       1147       TO WORK AREA         1147       C       MOV       DS.SI       C       1149       02DE       8E DE       C         1149       02DE       8E DE       C       MOV       DS.SI       1150       C         1150       C       PUSH       AX       IUSE THE FUNCTION CODE IN (AH) TO ROUTE       1152       02E1       8A C4       C       MOV       AL.AH       I USE THE FUNCTION CODE IN (AH) TO RECESSING	1141	02D5	50	5			C		PUSH		
1144       02D8       06       C       PUSH       ES         1145       C       C       MOV       BX,DS       ; SAVE INPUT PARAMETER IN BX         1146       02D9       8C DB       C       MOV       BX,DS       ; SAVE INPUT PARAMETER IN BX         1147       C       1147       C       1148       02DB       BER       C       MOV       SI,BIOSDATA       ; ESTABLISH ADDRESSABILITY TO WORK AREA         1149       02DE       8E DE       C       MOV       DS,SI       1150         1150       C       PUSH       AX       IUSE THE FUNCTION CODE IN (AH) TO ROUTE         1152       02E1       8A C4       C       MOV       AL,AH       ; USE THE FUNCTION CODE IN (AH) TO ROUTE							C		PUSH	DS	
1146     02D9     8C DB     C     MOV     BX,DS     ; SAVE INPUT PARAMETER IN DA       1147     C       1148     02DB     BE     R     C     MOV     SI,BIOSDATA     ; ESTABLISH ADDRESSABILITY TO WORK AREA       1149     02DE     8E DE     C     MOV     DS,SI       1150     C     1151     02E0     50     C     PUSH     AX       1152     02E1     8A C4     C     MOV     AL,AH     ; USE THE FUNCTION CODE IN (AH) TO ROUTE	1144								PUSH	٤S	
1148     02DB     BE    R     C     MOV     SI,BIOSDATA     ; ESTABLISH ADDRESSABILITY TO FORE AREA       1149     02DE     8E     DE     C     MOV     DS,SI       1150     C     PUSH     AX       1151     02E1     8A     C4     C     MOV     AL,AH     ; USE THE FUNCTION CODE IN (AH) TO ROUTE       1152     02E1     8A     C4     C     MOV     AL,AH     ; CONTROL TO THE PROPER PROCESSING	1146	02D9	8	C DE	3		C		MOV	BX,DS	
1149     02DE     8E     C     MOV     DS,SI       1150     C       1151     02E0     50     C     PUSH     AX       1152     02E1     8A     C4     C     MOV     AL,AH     ; USE THE FUNCTION CODE IN (AH) TO ROUTE       1152     02E1     8A     C4     C     MOV     AL,AH     ; CONTROL TO THE PROPER PROCESSING		0205	в	ε.		R			HOV	SI,BIOSDATA	; ESTABLISH ADDRESSABILITY TO WORK AREA
1151     02E0     50     C     PUSH     AX       1152     02E1     8A     C4     C     MOV     AL,AH     ; USE THE FUNCTION CODE IN (AH) TO ROUTE       1152     02E1     8A     C4     C     MOV     AL,AH     ; CONTROL TO THE PROPER PRCCESSING	1149						С			DS,SI	
TISE CONTROL TO THE PROPER PROCESSING							C				. USE THE EINCTION CODE IN (AH) TO ROUTE
	1152 1153										CONTROL TO THE PROPER PROCESSING

L

-

	C CBH
02E5 98	
02E5 98 02E6 88 F0	C MOV SI.AX ; ROUTINE
02E8 3D 0014 90	C CHP AX, BIOS_JMP_TABLE_LENGTH
02EC 58	
02ED 72 19	
02EF B0 01	C HOV AL, RC_INVALID_FUNC ; INVALID FUNCTION CODE
02F1 EB 1A 90	C JMP BIOS_ERROR
	c
02F4	C BIOS_JMP_TABLE LABEL WORD
02F4 032D R	C DW SET_MODE
02F6 0367 R	C DW SEND_KEYSTROKES
02F8 0438 R	C DH SET_INTERRUPT_VECTOR C DW RESET_INTERRUPT_VECTOR
02FA 0434 R	
02FC 0484 R	
02FE 04C3 R	C DW SEND_MESSAGE C DW CANCEL_MESSAGE
0300 062B R	C DH MODIFY_XLAT_IN
0302 0648 R 0304 0653 R	C DW MODIFY_XLAT_OUT
0304 0653 R 0306 065E R	C DW GUERY_KEYBOARD_IDENTIFIER
0500 0052 4	C C
= 0014	C BIOS_JMP_TABLE_LENGTH EQU \$-BIOS_JMP_TABLE
	C
0308	C BIOS_ROUTE:
0308 2E: FF A4 02F4 R	C JMP BIOS_JMP_TABLE(SI)
	C
	C C
	C ; ***********************************
	C ;# Procedure Name: BIOS_ERROR
	C ;*
	C 1* Function: To return to the caller with an error course and
	C ;#
	C ;* Input:
	C 3#
	C ;* (AL) = error code
	C ;*
	C ;* Output:
	C ;* C ;* CF = 1
	C ;* CF = 1 C ;* (AL) = error code
	C
	c .
030D	C BIOS_ERROR: ; INDICATE AN ERROR HAS OCCURRED
030D F9	C STC , INDICATE CALLER
030E EB 04 90	
	C ;* C ;* Procedure Name: BIOS_EXIT
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;*
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was C ;* Function: To return to the caller indicating the function was
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was C ;* completed successfully
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* Function: To return to the caller indicating the function was C ;* Function: To return to the caller indicating the function was C ;* completed successfully C ;*
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was C ;* Completed successfully C ;* C ;* Input: None
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was C ;* Function: To return to the caller indicating the function was C ;* Completed successfully C ;* C ;* Input: None C ;*
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was C ;* Completed successfully C ;* C ;* Input: None
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;*
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was C ;* completed successfully C ;* C ;* Input: None C ;* C ;* Output: C ;* C ;* CF = 0 C ;* (AL) = 0
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was C ;* completed successfully C ;* C ;* Input: None C ;* C ;* Output: C ;* C ;* CF = 0 C ;* (AL) = 0
	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;*
	C ;* C ;*
0311	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was C ;* completed successfully C ;* C ;* Input: None C ;* C ;* Output: C ;* C ;* CF = 0 C ;* C ;* (AL) = 0 C ;* C ;* C ;* C ;* Output: C ;* C ;* (AL) = 0 C ;* C ; C ; C ; C ; C ; C ; C ; C ;
0311 F8	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* C ;* Function: To return to the caller indicating the function was c ;* completed successfully C ;* C ; C ; C ; C ; C ; C ; C ; C ;
	C ;* C ; ; C ; C ; C ; C ; C ; C ; C
0311 F8 0312 32 C0	C ;* C ; C ; C ; C ; C ; C ; C ; C ;
0311 F8 0312 32 C0 0314	C ;* C ;* C ;* C ;* Function: To return to the caller indicating the function was c ;* C ; C ; C ; C ; C ; C ; C ; C ;
0311 F8 0312 32 C0 0314 0314 07	C ;* C ;* C ;* Function: To return to the caller indicating the function was C ;* C ; C ; C ; C ; C ; C ; C ; C ;
0311 F8 0312 32 C0 0314 0314 07 0315 1F	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was c ;* completed successfully C ;* C ; C ; C ; C ; C ; C ; C ; C ;
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F	C ;* C ;* C ;* Function: To return to the caller indicating the function was c ;* C ; C ; C ; C ; C ; C ; C ; C ;
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was c ompleted successfully C ;* C
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was c ;* completed successfully C ;* C ;* Dutput: C ;* C ;* Output: C ;* C ;* (AL) = 0 C ;* C BIOS_EXIT: C CLC ; INDICATE SUCCESSFUL COMPLETION C CLC ; INDICATE SUCCESSFUL COMPLETION C XOR AL,AL C BIOS_RETURN: C POP ES ; RESTORE REGISTERS C POP DI C POP SI C POP SI C POP CX
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 55 0317 55 0319 5A 0319 59	C ;* C ;* C ;* C ;* Function: To return to the caller indicating the function was c ;* C ; ; INDICATE SUCCESSFUL COMPLETION C ;* C ; ; INDICATE SUCCESSFUL COMPLETION C ;* C ; ; RESTORE REGISTERS C ; POP DS C ; POP DS
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A 0319 59 0314 5B	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was c ompleted successfully C ;* C
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A 0319 59 0314 5B	C ;* C ;* C ;* Function: To return to the caller indicating the function was completed successfully C ;* C ; * C ; *
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A 0319 59 0318 55	C ;* C ;* C ;* Function: To return to the caller indicating the function was completed successfully C ;* C ; * C
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A 0319 59 0318 55 0318 55 0318 55 0318 6B EC 0316 72 07	C ;* C ;* C ;* C ;* Function: To return to the caller indicating the function was completed successfully C ;* C ; * C
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A 0319 59 0318 55 0318 55 0316 68 EC 0318 72 07 0320 60 66 06 FE	C ;* C ;* Procedure Name: BIOS_EXIT C ;* Function: To return to the caller indicating the function was completed successfully C ;* C ;* Input: None C ;* C ;* Output: C ;* C ;* CF = 0 C ;* C ;* (AL) = 0 C ;* C BIOS_EXIT: C CLC ; INDICATE SUCCESSFUL COMPLETION C CLC ; INDICATE SUCCESSFUL COMPLETION C CLC ; INDICATE SUCCESSFUL COMPLETION C CLC ; RESTORE REGISTERS C POP DS C POP DS C POP DS C POP DS C POP SI C PO
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A 0319 59 0318 55 0318 55 0318 65 EC 0318 72 07 0320 60 66 06 FE	C ;* C ;* C ;* Function: To return to the caller indicating the function was completed successfully C ;* C ; * C
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0317 5E 0318 5A 0319 59 0318 55 0316 6B EC 0318 72 07 0320 60 66 06 FE 0324 EB 05 90 0327	C ;* C ;* C ;* Function: To return to the caller indicating the function was completed successfully C ;* C ; * C ; *
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A 0319 59 0318 55 0312 6B EC 0318 55 0312 6B EC 0318 72 07 0320 60 66 06 FE 0324 EB 05 90	C ;* C ;* C ;* Function: To return to the caller indicating the function was completed successfully C ;* C ; ; INDICATE SUCCESSFUL COMPLETION C ;* C ;* C ; ; INDICATE SUCCESSFUL COMPLETION C ;* C ;* C ;* C ; ; INDICATE SUCCESSFUL COMPLETION C ;* C ;* C ;* C ; ; RESTORE REGISTERS C ; POP DS C ; POP SI C ; POP SI C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP DX C ; POP SI C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C ; POP BX C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C ; POP BX C ; SO THE USER CAN TEST THE CARRY FLAG IN THE STACK C ; POP BX C
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A 0319 59 0318 55 0310 6B EC 0318 55 0310 6B EC 0318 72 07 0327 80 4E 06 01 0328	C ;* C ;* Procedure Name: BIOS_EXIT C ;* C ;* Function: To return to the caller indicating the function was c completed successfully C ;* C BIOS_EXIT: C CLC ; INDICATE SUCCESSFUL COMPLETION C CLC ; S RESTORE REGISTERS C POP ES ; RESTORE REGISTERS C POP ES ; RESTORE REGISTERS C POP DI C POP SI C POP DI C POP BX C POP BX C POP BX C POP BX C SO THE USER CAN TEST THE CARR? FLAG IN THE STACK: MOV BP,SP ; SO THE USER CAN TEST THE CARR? FL C JC BIOS_RETURN_1 ; TO SEE IF THE OPERATION SUCCEEDED AND BYTE PTR [BP+6],0FH JHP BIOS_RETURN_2: BIOS_RETURN_2:
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0317 5E 0318 5A 0319 59 0318 55 0316 68 EC 0318 72 07 0320 60 66 06 FE 0324 EB 05 90 0327 0327 80 4E 06 01	C ;* C ;* C ;* Function: To return to the caller indicating the function was completed successfully C ;* C ; (AL) = 0 C ;* C ;* C ; (AL) = 0 C ;* C ;* C ; (AL) = 0 C ; (AL) = 0 C ;* C ; (AL) = 0 C ; (AL) =
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A 0319 59 0318 55 031C 68 EC 0318 72 07 0320 60 66 06 FE 0327 80 4E 06 01 0328 0 325 5D	<pre>C ;* C ;* C ;* C ;* C ;* C ;* Function: To return to the caller indicating the function was C ;* C ;*</pre>
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 55 0318 55 0318 55 0312 68 EC 0318 55 0312 68 EC 0318 72 07 0320 60 66 06 FE 0324 EB 05 90 0327 80 4E 06 01 0328 0 0325 5D 0 322 CF	<pre>C ;* C ;*</pre>
0311 F8 0312 32 C0 0314 0314 07 0315 1F 0316 5F 0317 5E 0318 5A 0319 59 0318 55 031C 68 EC 0318 72 07 0320 60 66 06 FE 0327 80 4E 06 01 0328 0 325 5D	<pre>C ;* C ;* C ;* C ;* C ;* C ;* Function: To return to the caller indicating the function was C ;* C ;*</pre>

101	4,641,262 102
101	
	C ;*
	C ;*
	C ;* Function: To change the mode of the display and/or keyboard * C ;*
	C ;* Input: C ;*
	C ;* (AL) = 0 - switch to host computer mode C ;* (AL) = 1 - switch to personal computer mode
	C ;* (AL) = 2 - display host computer screen image
	C ;* (AL) = 3 - display personal computer screen image C ;*
	C i# Dutput: None * C i#
	C ;ииииииииииииииииииииииииииииииииииии
032D 032D 0A CO	C SET_MODE: C OR AL,AL ; IS MODE = 0?
D32F 75 03 D331 EB 1A 90	C JNZ SET_MODE_1 ; NO, CONTINUE
334	C SET_MODE_1:
0334 FE C8 0336 75 03	C DEC AL ; IS MODE = 1? C JNZ SET_MODE_2 ; NO, CONTINUE
0338 EB 20 90 0338	C JHP SET_HODE_TO_PC ; YES, SWITCH TO PERSONAL COMPUTER MODE C SET_HODE_2:
33B FE C8 330 75 03	C DEC AL ; IS MODE = 2? C INZ SET MODE 3 ; NO, CONTINUE
133F EB 11 90	C JMP SET_MODE_DISPLAY_HOST ; YES, SWITCH DISPLAY TO HOST MODE
342 FE C8	C SET_MODE_3: C DEC AL ; IS MODE = 3? C DEC AL ; IS MODE = 3?
1344 75 03 1346 EB 17 90	C JHZ SET_HODE_4 ; NO, CONTINUE C JMP SET_HODE_DISPLAY_PC ; YES, SWITCH DISPLAY TO FC MODE
1349	C SET_MODE_4:
1349 BO 02 1348 EB CO	C MOV AL,RC_INVALID_MODE ; INDICATE INVALID MODE VALUE C JMP BIOS_ERROR
	C ; PUT THE KEYBOARD AND DISPLAY INTO HOST COMPUTER MODE
	C J THE 3278/3279 DISPLAY WILL SHOW THE SCREEN IMAGE FROM THE
	C ; 3274/3276 CONTROL UNIT C ; ALL KEYSTROKES WILL BE ROUTED TO THE 3278/3279 DISPLAY
34D	C SET_HODE_TO_HOST:
34D 80 0E 0000 R 80	C OR TERH_MODE,T_HDST # INDICATE TERMINAL IN HOST MODE
	C ; PUT THE DISPLAY INTO HOST COMPUTER MODE C ; THE 3278/3279 DISPLAY WILL SHOW THE SCREEN IMAGE FROM THE
	C ; 3274/3276 CONTROL UNIT C
352 352 BA 03E3	C SET_HODE_DISPLAY_HOST: C HOV DX,CHD_6255 ; SWITCH THE DISPLAY TO HOST MODE
355 BO OF	C MOV AL, HOST_HODE
357 EE	C DUT DX,AL C
358 EB B7	C JMP BIOS_EXIT ; RETURN TO CALLER C
	C ; PUT THE KEYBOARD AND DISPLAY INTO PERSONAL COMPUTER MODE C ; The 3278/3279 DISPLAY WILL SHOW THE SCREEN "HAGE FROM THE
	C ; PERSONAL COMPUTER C ; ALL KEYSTROKES WILL BE ROUTED TO THE PERSONAL COMPUTER
35A	C C SET_HODE_TO_PC:
35A 80 26 0000 R 7F	C AND TERM_MODE,NOT T_HOST ; INDICATE TERMINAL IS IN PC MODE
	C
	C ; THE 3278/3279 DISPLAY WILL SHOW THE SCREEN IMAGE FROM THE
	C ; PERSONAL COMPUTER C
35F BÁ 03E3	C SET_MODE_DISPLAY_PC: C Mov dx,cmd_d255 ; Switch the display to personal
0362 80 DE 0364 EE	C MOV AL, PO_NODE ; COMPUTE MODE C OUT DX, AL
365 EB AA	C C C JMP BIOS EXIT ; RETURN TO CALLER
	C
	C ;*
	C ;* Procedure Name: SEND_KEYSTROKES C ;*
	C ;* Function: To send a series of keystrokes to the 3276/3279 C ;*
	C ;* Inpu(: C ;*
	C ;* BX:DX = address of encoded keystrokes
	C ;*
	C ;* Output: None C ;*
	C ;************************************

.

4,641,262

### 102

.

104

103

1338 1339 SEND\_KEYSTROKES: 0367 000 FBA\_STATE,F\_ENABLED : IS THE FEATURE BUS ENABLED? SEND\_KEYSTROKES\_3 ; YES, INSURE KEYSTROKES ARE RECEIVED F6 06 0000 R 02 75 09 1340 0367 TEST 1341 036C JNZ 1342 C C C SEND\_ENCODED\_KEYSTROKES\_T0\_3278 ; DECODE AND SEND 1 KEYSTROKES SEND\_KEYSTROKES\_2 ; INVALID KEYSTROKE ENCODING 1343 036F F8 0000 F CALL 1344 1345 0371 72 02 С JC 1346 C 1347 0373 ċ SEND\_KEYSTROKES\_1: ; KEYSTROKES WERE VALID BIOS\_EXIT 0373 EB 9C 1348 С JMP 1349 С 1350 0375 C C SEND\_KEYSTROKES\_2: ; KEYSTROKES WERE INVALID 0375 EB 96 BIOS\_ERROR 1351 JHP 1352 SEND\_KEYSTROKES\_3: с с с 1353 0377 ; FIND THE FIRST ENCODED KEYSTROKE 1354 ESIBX 0377 8E C3 MOV 1355 0379 88 F2 MOV SI.DX AL, ES: [SI] 1356 000000 037B 26: 8A 04 MOV ; IS IT THE RESET KEY? CMP 1357 037E 3C 88 AL.88H SEND\_KEYSTROKES\_4 ; YES, RESET SHOULD UNLOCK THE ; KEYBOARD IF IT IS LOCKED 74 15 1358 0380 JE 1359 1360 QUERY\_KBD\_STATE ; GET CURRENT KEYBOARD STATE 0382 E8 03C9 R CALL 1361 1362 0000 PUSH 1363 0385 51 CX,DELAY\_100\_MS ; WAIT FOR 100 HILLISECONDS 89 681A E2 FE 1364 0386 HOV LOOP 1365 0389 Ś č POP сх 1366 59 D38B 1367 C AL,RC\_OK ; WAS QUERY ACCEPTED BY THE 3274? SEND\_KEYSTROKES\_2 ; NO, INFORM USER OF THE PROBLEM KBD\_STATE,00H ; YES, IS KEYBDARD UNLOCKED? SEND\_KEYSTROKES\_5 ; NO, KEYBDARD IS LOCKED CHP 038C 3C 00 1368 С JNÉ 1369 038E 75 E5 CHP 1370 1371 80 3E 005C R 00 75 25 0390 с с 0395 JNE 1372 C SEND\_KEYSTROKES\_4 Ċ 1373 0397 SEND\_ENCODED\_KEYSTROKES\_TO\_3278 ; DECODE AND SEND 1374 0397 E8 0000 E C C CALL KEYSTROKES 1375 1376 SEND\_KEYSTROKES\_2 ; INVALID KEYSTFOKE ENCODING 039A 72 D9 C JC 1377 C C C CX,DELAY\_100\_HS/10 ; WAIT FOR 10 HILLISECONDS ноу 039C B9 0A69 1379 039F E2 FE 100P 1380 с с QUERY\_KBD\_STATE ; GET NEW KEYBOARD STATE AL,RC\_OK ; WAS QUERY ACCEPTED BY 3274? SEND\_KEYSTROKES\_2 ; NO, INFORM USER OF THE PROBLEM KBD\_STATE,00H ; IS THE KEYBOARD INHIBITED? SEND\_KEYSTROKES\_1 ; YES, RETURN TO CALLER 1381 E8 03C9 R CALL D3A1 CHP с с 1382 03A4 3C 00 1383 03A6 75 CD JHE 1384 0348 80 3E 005C R 00 С СКР 74 C4 č JE 1385 03AD AL,RC\_AID\_SENT ; INDICATE AID IS NOW PENDING KBD\_STATE,OCOH ; IS THE KEYBOARD INHIBITED BECAUSE OF ; AN ATTENTION INTERRUPT? SEND\_KEYSTROKES\_2 ; YES, RETURN TO CALLER AL,RC\_MESSAGE\_LOST ; NO, INDICATE PART OF THE MESSAGE ; WAS LOST DURING TRANSMISSION SEND\_KEYSTROKES\_2 ; RETURN TO CALLER 1386 C 1367 MOV 03AF 80 0C C C 0381 80 3E 005C R CO CHP 1389 C č JE 0366 74 BD 1390 1391 0388 B0 0D С HOV 1392 С 1393 JMP 03BA EB B9 С 1394 c 5: AL,RC\_AID\_PERDING ; INDICATE AID IS PENDING KBD\_STATE,OCOH ; IS THE KEYBOARD INHIBITED BECAUSE OF KBD\_STATE,OCOH ; A PREVIOUS ATTENTION INTERRUPT? SEND\_KEYSTRCKES\_2 ; YES, RETURN TO CALLER AL,RC\_KEIBDARD\_INHIBITED ; NO, INDICATE KEYBOARD IS ; INHIBITED FOR ANOTHER REASON SEND\_KEYSTRCKES\_2 ; RETURN TO CALLER SEND KEYSTROKES\_5: 1396 0350 С 03BC MOV BO OA С 1397 038E 80 3E 005C R CO CHP 1398 1399 С JE 1400 03C3 74 B0 C c нον 1401 03C5 80 0B 1402 С JMP 1403 D3C7 EB AC C C 1404 С 1405 1406 如金家家能放这些家子多家的生活就都有在这个不不不是要使就在这些这些的是不是在这个个人,不一都有的这些不是没有必要要要要要要要要要要要要要要要要要 1407 1.8 1408 1409 Procedure Name: QUERY\_KBD\_STATE ;\* 1410 :\* Function: To get the current keyboard state from the 3274 **;**\* 1411 1412 ;\* j# Input: None 1413 1414 ; H Output: 1415 С 1416 ;× AH is unpredictable 1417 C C ;× (AL) = error code KBD\_STATE = current keyboard state . . 1418 С ;× 1419 1420 C 3 "你就来出来你跟你来这家家我在跟我就这是我的这些我们就能要我没有这个我们就没有这些不是不是不是不是这个我们不是有这些不是没有不是有不是有不是不是不是不是不是 C C C 1421 1422 QUERY\_KBD\_STATE PROC NEAR 1423 0309 C 1424 ; SAVE WORK REGISTERS PUSH сх 0309 51 1425 0000 PUSH nx 03CA 52 1426 1427 PUSH 1428 1429 0308 1**E** AH,FUNC\_SET\_INT\_VECTOR ; INITIALIZE QUERY KEYBOARD AL,VECTOR\_QUERY\_KBD\_STATE ; STATE INTERRUPT VECTOR DX,OFFSET QUERY\_STATE\_INT\_HAND С HOV 0300 84 02 HOV С 1430 03CE B0 03 MOV BA 0410 R C C 0300 1431 PUSH cs 0E 1432 0303

•

106

•

-

1433 1434 1435	03D4 1F 03D5 CD 50 03D7 1F	C POP DS C INT NDSP_INT_CODE C POP DS	
1436 1437 1438 1439 1440 1441	03D8 B4 04 03DA B0 03 03DC 80 16 005C R 03E0 B9 0001 03E3 CD 50	C MOV AH,FUNC_SET_BUFFER_ADDRESS ; INITIALIZE QUERY KEYBOARD C MOV AL.VECTOR_GUERY_KBD_STATE ; STATE INPUT BUFFER ADDRESS C LEA DX;KBD_STATE C MOV CX,1 C INT NDSP_INT_CODE	2
1442 1443 1444	03E5 80 0E 0000 R 10	C OR FBA_STATE,F_QUERY_IN_PROGRESS ; INDICATE QUERY KEYBDAR C ; STATE IS IN PROGRESS	0
1445 1446 1447 1448 1449 1450 1451 1452	D3EA B4 05 03EC B0 03 03EE B9 0001 03F1 BD 16 005B R 03F5 CD 50 03F7 72 0A	C HOV AH,FURC_SEND_MESSAGE; SEND THE DUMMY QUERY KEYBOARD C MOV AL,VECTOR_QUERY_KBD_STATE; MESSAGE TO THE 3274 C MDV CX,1 C LEA DX,QUERY_KED_STATE_MSG C INT ND5P_INT_CODE C JC QUERY_KED_STATE_2; ERROR WHILE SENDING MESSAGE	
1453 1454 1455	03F9 03F9 F6 06 0000 R 10 03FE 75 F9	C C QUERY_KBD_STATE_1: C TEST FBA_STATE,F_QUERY_IN_PROGRESS ; HAIT FOR QUERY REPLY C JNZ QUERY_KED_STATE_1 ; TO BE RECEIVED	
1456 1457 1458	0400 AD 005D R	C MOY AL, QUERY_RC ; SET UP RETURN CODE FOR CALLER	
1459	0403 0403 84 C8	C QUERY_KBD_STATE_2: C MOV CL,AL ; SAVE RETURN CODE	
1461	D405 B4 D3	C C C MOV AH,FUNC_RESET_INT_VECTOR ; RESET QUERY KEYBOARD	
1463 1464	0407 B0 03 0409 CD 50	C HOV AL,VECTOR_QUERY_KBD_STATE ; STATE INTERRUPT VECTOR C INT NDSP_INT_CODE	
1465 1466 1467	040B 84 Cl 040D 54	C MOV AL,CL ; RESTORE RETURN CODE C POP DX ; RESTORE NORK REGISTERS	
1468 1469	040E 59 040F C3	C POP CX C RET I RETURN TO CALLER	
1470 1471	0410	C C QUERY_KBD_STATE ENDP	
1472 1473		c	
1474 1475		C :************************************	**
1476 1477		C ;*	*
1478		C 3* Procedure Name: QUERY_STATE_INT_HAND C 3*	*
1479 1480		C }* Function: To process a Query Keyboard State reply interrupt C }*	*
1481 1482		C ;* Input:	*
1483		C (AH) = reason interrupt handler was entered	¥
1484 1485		C ;* C ;* Output:	*
1486 1487		C ;* C ;* QUERY_RC = return code	*
1488 1489		C ;* K6D_STATE = current keyboard state C ;*	*
1490 1491	·	C ;************************************	**
1492 1493 1494 1495	0410 0410 BB R 0413 8E DB	C QUERY_STATE_INT_HAND PROC FAR C MOV BX,BIOSDATA ; ESTABLISH ADDRESSABILITY TO DATA ARE/ C MOV DS,BX	A.
1496 1497	0415 C6 06 0050 R 00 041A 80 FC 01	C MOV QUERY_RC,RC_OK ; SET UP RETURN CODE C CHP AH,REASON MESSAGE_RECEIVED ; IS THIS THE REPLY?	
1498	041D 74 OF	C JE QUERY_STATE_INT_HAND_1 ; YES, RETURN TO CALLER	
1499 1500	041F C6 06 005D R 06 0424 80 FC 02	C MOV QUERY_RC,RC_POWER_ON_RESET; NO, SEE WHAT IT IS C CMP AH,REASON_POWER_ON_RESET; IS THIS A POWER ON RESET?	
1501 1502	0427 74 05 0429 C6 06 005D R 07	C JE QUERY_STATE_INT_HAND_1 ; YES, RETURN TO CALLER C MOV QUERY_RC,RC_PROTOCOL_VIOLATION ; NO, SOME OTHER ERROR	
1503	042E	C QUERY_STATE_INT_HAND_1:	
1505	042E 80 26 0000 R EF	C AND FBA_STATE, NOT F_QUERY_IN_PROGRESS ; INDICATE QUERY KB	D
1506 1507	0433 CB	C ; STATE REPLY WAS RECEIVED C RET ; RETURN TO BIOS	
1508 1509	0434	C QUERY_STATE_INT_HAND ENDP	
1510 1511		C INCLUDE NDSPAP12.ASH	
1512		c	
1513		C C C	
1515 1516		C ;K	+# #
1517 1518		C ;* Procedure Name: RESET_INTERRUPT_VECTOR	*
1519		C 1* Function: To reset a 3276/3279 Attachment interrupt vector	*
1520 1521		C ;* C ;* Input:	*
1522 1523		C ;* C ;* (AL) = interrupt vector code	*
1524 1525		C ;*	*
		C ;* Output:	• .
*			

-	<b>^</b>
	INV.

	107	108
		C ;# C ;# BX and DX = 0
		C 34
		C ;************************************
0434 0434 3 0436 3	53 DB 53 D2	C RESET_INTERRUPT_VECTOR: C XOR BX,BX ; SET INTERRUPT HANDLER ADDRESS TO ZERO C XOR DX,DX
		C ;####################################
		C ;* C ;* Procedure Name: SET_INTERRUPT_VECTOR
		C ;# Function: To set a 3278/3279 Attachment interrupt vector C ;# Function: To set a 3278/3279 Attachment interrupt vector
		C ;* Input: C ;*
		C ;# (AL) = interrupt vector code
		C ;* BX:DX = address of interrupt handler C ;*
		C ;* Dutput: None
		C }************************************
<b>0</b> 438		C SET_INTERRUPT_VECTOR:
D438	E8 0664 R	r rall FIND VECTOR ; IDENTIFY WHICH VECTOR IS TO BE CHARGE
043B 043D	73 03 E9 030D R	C JNC SET_VECTOR_1 ; VECTOR WAS FOUND C JMP BIOS_ERROR ; ERROR - INVALID VECTOR WAS SPECIFIED
0440		C SET_VECTOR_1:
0440	FA	C C CLI ; TURN OFF INTERRUPTS SO HE ARE NOT C CLI ; DISTURBED WHILE CHANGING THE VECTOR
0441	89 15	MOVE HORD RTP LOTI THTEPRIPT VECTOR.DX ; SAVE ADDRESS OF
	89 15 89 50 02	C MOV WORD PTR (DI).INTERRUPT_VECTOR+2,BX ; INTERRUPT HANDLE
0446	C7 45 D4 D0D0	C NOV WORD PTR [DI].BUFFER_ADDRESS,0 ; CLEAR ADDRESS OF
	C7 45 06 0000	C HOV WORD PIR (DI).BUFFER_ADDRESS+2,0 ; INPUT BUFFER
0450	FB	C STI ; ALLOW INTERRUPTS NOW
9451	OB DB	C OR BX, EX ; IS THERE AN INTERRUPT HANDLER?
	74 10	C JZ SET_VECTOR_2 ; NO, THIS IS A RESET REQUEST
55	E8 029A R	C CALL WAIT_FOR_POLL_ACK ; WAIT FOR THE PREVIOUS POLL REQUEST
	BA 03E4	C ; TO BE ACKNOWLEDGED C MOV DX,COMMAND_FORT ; SIGNAL IPL COMPLETE TO THE 3274
5B	B0 0D	C MOV AL, IPL_COMPLETE
50 5E	EE 80 0E 0000 R 02	C OUT DX.AL C OR FBA_STATE.F_ENABLED ; INDICATE FEATURE EUS IS ENABLED
	EB 1C	C JHP SHORT SET_VECTOR_4 ; PETURN TO CALLER
465		C SET_VECTOR_2:
65	8B 16 0011 R 0B 16 0022 R	C MOV DX,DATA_XFER_VECTOR.INTERRUPT_VECTOR+2 ; IS THERE STI C DR DX,SCREEN_CAPTURE_VECTOR.INTERRUPT_VECTOR+2 ; AN ACTI
	0B 16 0033 R	C OR DX.WRAP_TEST_VECTOR.INTERRUPT_VECTOR+2 ; INTERRUPT
0471	75 OE	C JNZ SET_VECTOR_4 ; HANDLER? C
		C ; NO, DISABLE THE FEATURE EUS INTERPAL
0473	E8 029A R	C ; TO BE ACKNOWLEDGED
0674	BA 03E4	C HOV DELLOCHMAND_PORT ; DISABLE 3276/3279 ATTACHMENT
0479	B0 0C	C MOV AL, RESET_IPL_COMPLETE ; OPTION FEATURE BOS INTERFACE
047B 047C	EE 80 26 0000 R FD	C OUT DX,AL C AND FBA_STATE,NOT F_ENABLED ; INDICATE FEATURE BUS IS
		C I I NOT ENABLED
		c
481	E9 0311 R	C SET_VECTOR_4: C JMP BIOS_EXIT ; RETURN TO CALLER
	_,	C
		C Page C ;###+5#++
		C ;* Procedure Name: SET_BUFFER_ADDRESS
		C ;* C ;* Function: To set an input buffer address
		C ;* Input: C ;*
		C ;# (AL) = interrupt vector code C ;# BX:DX = address of input buffer
		C ;*
		C ;# Dutput: None C ;#
		C }₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
0484		C C SET_BUFFER_ADDRESS: C CALL FIND VECTOR ; IDENTIFY WHICH INPUT BUFFER IS BEING
	EB 0664 R	C CALL FIND_VECTOR ; IDENTIFY WHICH INFOR DOTTER TO DESCRI
0487	73 03	C JNC SET_BUFFER_ADDRESS_1 ; VALID IDENTIFIER
	E9 030D R	C JMP BIOS_ERROR ; ERROR - INVALID BUTTER DELIVER

.

1620	048C		С	SET_BUFFER_ADDRESS_1:
1621	048C	83 70 02 00	C	CHP WORD PTR [DI].INTERRUPT_VECTOR+2,0 ; IS THERE AN
1622			c	; INTERRUPT HANDLER FOR THIS TYPE OF ; INPUT?
1623 1624	0490	75 05	C C	; INPUT? JNE SET_BUFFER_ADDRESS_2 ; YES, CONTINUE
1625		80 05	č	MOV AL, RC_INVALID_VECTOR ; INDICATE INVALID BUFFER TYPE
1626		E9 030D R	č	JHP BICS_ERROR
1627			C	-
1628 1629	0497		ç	SET_BUFFER_ADDRESS_2: OR CX,CX ; IS THIS A ZERO LENGTH BUFFER?
1630	0497	0B C9 75 04	C C	OR CX,CX ; IS THIS A ZERU LENGTA DUFFER: JNZ SET_BUFFER_ADDRESS_3 ; NO, CONTINUE
1631	0498	8B D1	č	MOV DX,CX ; YES, ASSUME NO BUFFER WAS SPECIFIED
1632	0490	6B D9	С	MOV BX.CX
1633			C	···· ·····
1634 1635	049F 049F	FA	C C	SET_BUFFER_ADDRESS_3: CLI ; DISABLE INTERRUPTS SO WE ARE NOT
1636	• • • •		č	; DISTURBED WHILE CHANGING THE
1637			C	; BUFFER ADDRESS
1638 1639			c	HOV WORD PTR [DI].BUFFER_ADDRESS,DX ; SAVE THE ADDRESS
1640		89 55 04 89 5D 06	C C	MOV HOPP PTE [DI], EUFFER ADDRESS+2, BX ; OF THE INPUT BUFFEF
1641		89 4D 08	č	HOV (DILEUFFER LENGTH,CX ; SAVE THE LENGTH OF THE BUFFER
1642	04,49	E5 02C2 R	c	CALL RESET_EUFFER_COUNTS ; RESET THE RECEIVED DATA COUNTS
1643 1644	0/10	50	C	STI ; ALLOW INTERRUPTS TO BE PROCESSED
1645	OGAC	FB	c c	STI ; ALLOW INTERRUPTS TO BE PROCESSED
1646	0440	CB DB	ē	OR BX, BX ; IS THERE A BUFFER?
1647	04AF	74 DF	C	JZ SET_BUFFER_ADDRESS_4 ; NO, CONTINUE
1648	0481	80 7D 0E 04	ç	CMP [DI].MESSAGE_TYFE.HT_SET_SC_OPTIONS ; IS THIS THE SCREE ; CAPTURE BUFFER?
1649 1650	0485	75 09	C C	JNE SET_BUFFER_ADDRESS_4 ; NO, CONTINUE
1651			č	
1652	04B7	E8 029A R	C	CALL WAIT_FOR_POLL_ACK ; WAIT FOR THE PREVIOUS POLL REQUEST
1653 1654	86B/	BA 03E4	C C	1 TO BE ACKNOWLEDGED MOV DX,COMMAND_PORT; YES, INDICATE SCREEN CAPTURE
1655		B0 04	c	MOV AL, SCREEN_CAPTURE_COMPLETE ; FROCESSING IS COMPLETE
1656	04BI		c	OUT DX,AL
1657			C	/
1658 1659	0400	E9 0311 R	C C	SET_BUFFER_ADDRESS_4: JMP BIOS_EXIT ; RETURN TO CALLER
1660	0405	EY USI' R	č	Jur Bios_Exti , Ketoni to bacea
1661			С	Page
1662			c	2 赛我乐家常家里那家家家家做家家在的家主专业成本自由自自自由的分子自身有些有些不能能感要完成要就想不不可以不能找了这些不是这个不少有些要做家
1663 1664			C C	这些常有这个不是不是不是不是不是不是不是不是不是不是不是不是不是不是不是不是不不不是有,有有多少,有是有不多是是不是是不是是这些这些,如果还是这些这些是是是这一个。 3 第一
1665			č	* Procedure Name: SEND_MESSAGE
			~	3*
1666			c	
1667			C	;* Function: To send a message to the 3274 via the feature bus *
1667 1668			C C	;* Function: To send a message to the 3274 via the feature bus * ;*
1667			C	;* Function: To send a message to the 3274 via the feature bus *
1667 1668 1669 1670 1671				<ul> <li>;* Function: To send a message to the 3274 via the feature bus</li> <li>;*</li> <li>;* Input:</li> <li>;*</li> <li>;*</li> <li>(AL) = interrupt vector code</li> </ul>
1667 1668 1669 1670 1671 1672				<pre>;* Function: To send a message to the 3274 via the feature bus ;* ;* i* Input: ;* (AL) = interrupt vector code ;* BX:DX = address of message * * * * * * * * * * * * * * * * * * *</pre>
1667 1668 1669 1670 1671				<ul> <li>;* Function: To send a message to the 3274 via the feature bus</li> <li>;*</li> <li>;* Input:</li> <li>;*</li> <li>;*</li> <li>(AL) = interrupt vector code</li> </ul>
1667 1668 1669 1670 1671 1672 1673 1674 1675				<pre>;* Function: To send a message to the 3274 via the feature bus ;* ;* i* Input: ;* (AL) = interrupt vector code ;* BX:DX = address of message ;* (CX) = length of message ;* j* Output: None **</pre>
1667 1668 1669 1670 1671 1672 1673 1674 1675 1676				<pre>;* Function: To send a message to the 3274 via the feature bus ;* ;* i* i* ;* (AL) = interrupt vector code ;* (AL) = address of message ;* (CX) = length of message ;* i* i*</pre>
1667 1668 1669 1670 1671 1672 1673 1674 1675				<pre>;* Function: To send a message to the 3274 via the feature bus ;* ;* i* Input: ;* (AL) = interrupt vector code ;* BX:DX = address of message ;* (CX) = length of message ;* j* Output: None **</pre>
1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677	04C3			<pre>;* Function: To send a message to the 3274 via the feature bus ;* i# i#</pre>
1667 1668 1669 1670 1671 1672 1673 1673 1675 1676 1677 1678 1679 1680	04C3	E8 0664 R		<pre>;* Function: To send a message to the 3274 via the feature bus ;* iF Input: * * * * * * * * * * * * * * * * * * *</pre>
1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681	04C3 04C6	73 03		<pre>;* Function: To send a message to the 3274 via the feature bus ;* i* i*</pre>
1667 1668 1669 1670 1671 1672 1673 1673 1675 1676 1677 1678 1679 1680	04C3 04C6			<pre>;* Function: To send a message to the 3274 via the feature bus ;* iF Input: * * * * * * * * * * * * * * * * * * *</pre>
1667 1668 1670 1670 1671 1673 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684	04C3 04C6 04C8 04C8	73 03 E9 030D R		<pre>;* Function: To send a message to the 3274 via the feature bus ;* i* i*</pre>
1667 1668 1869 1670 1671 1672 1673 1674 1675 1676 1677 1677 1677 1678 1679 1680 1682 1683 1684 1685	04C3 04C6 04C8 04C8	73 03		<pre>;* Function: To send a message to the 3274 via the feature bus ;* image: i</pre>
1667 1668 1670 1670 1671 1673 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684	04C3 04C6 04C8 04C8	73 03 E9 030D R		<pre>;* Function: To send a message to the 3274 via the feature bus ;* i* i*</pre>
1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1677 1678 1679 1680 1682 1683 1684 1685 1686 1687	04C3 04C6 04C8 04C8 04C8 04CB	73 03 E9 030D R 83 7D 02 00 75 05		<pre>i* Function: To send a message to the 3274 via the feature bus i* if if i* i*</pre>
1667 1668 1669 1670 1671 1673 1674 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1685 1686 1687 1688	04C3 04C6 04C8 04C8 04CB 04CB 04CF 04CF	73 03 E9 030D R 83 7D 02 00 75 05 B0 05		<pre>i* Function: To send a message to the 3274 via the feature bus i* if if i* i*</pre>
1667 1668 1670 1671 1672 1673 1674 1673 1674 1677 1678 1677 1678 1680 1681 1683 1684 1683 1684 1685 1688 1688 1689 1690	04C3 04C6 04C8 04C8 04CB 04CB 04CF 04CF	73 03 E9 030D R 83 7D 02 00 75 05		<pre>i* Function: To send a message to the 3274 via the feature bus i* if if i* i*</pre>
1667 1668 1670 1671 1672 1673 1674 1675 1677 1678 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1688 1689 1691	04C3 04C6 04C8 04C8 04CB 04CB 04CF 04CF 04D1 04D3	73 03 E9 030D R 83 7D 02 00 75 05 B0 05		<pre>:* Function: To send a message to the 3274 via the feature bus :* Input: :* Input: :* * * * * * * * * * * * * * * * * *</pre>
1667 1668 1670 1671 1672 1673 1674 1673 1674 1677 1678 1677 1678 1680 1681 1683 1684 1683 1684 1685 1688 1688 1689 1690	04C3 04C6 04C8 04C8 04CB 04CB 04CF 04CF	73 03 E9 030D R 83 7D 02 00 75 05 B0 05		<pre>i* Function: To send a message to the 3274 via the feature bus i* i*</pre>
1667 1668 1670 1671 1672 1673 1674 1673 1674 1675 1677 1678 1678 1681 1682 1684 1685 1688 1688 1688 1688 1689 1691 1692 1694	04C3 04C6 04C8 04C8 04CB 04CB 04C7 04D1 04D3 04D6 04D6	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R		<pre>:* Function: To send a message to the 3274 via the feature bus :* Input: :* Input: :* * * * * * * * * * * * * * * * * *</pre>
1667 1668 1669 1670 1671 1673 1674 1674 1675 1675 1677 1678 1680 1681 1683 1684 1685 1688 1688 1688 1688 1688 1689 1699 1699	04C3 04C6 04C8 04CB 04CB 04CB 04CF 04D1 04D3 04D6 04D6 04D8	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05		<pre>i* Function: To send a message to the 3274 via the feature bus i* Input: i* Input: i* AL) = interrupt vector code i* BX:DX = address of message i* CALL = length of message i* CALL = length of message i* SEND_MESSAGE:     CALL = FIND_VECTOR ; IDENTIFY THE TYPE OF MESSAGE TO BE SENT     JNC = SEND_MESSAGE_0 ; VALID MESSAGE TYPE     JMP BIOS_ERROR ; ERROR - INVALID MESSAGE TYPE SEND_MESSAGE_0:     CHP HORD PTR [DI].INTERRUPT_VECTOR+2,0 ; IS THERE AN</pre>
1667 1668 1670 1671 1672 1673 1674 1675 1674 1677 1678 1678 1680 1681 1683 1684 1685 1688 1688 1688 1688 1688 1688 1688	04C3 04C6 04C8 04C8 04CB 04CB 04CF 04D1 04D3 04D4 04D4 04D6 04D6 04D8 04DA	73 03 E9 030D R 83 7D 02 00 75 05 E9 030D R 0B C9 75 05 B0 07		<pre>i* Function: To send a message to the 3274 via the feature bus i* Input: i* Input: i* i*</pre>
1667 1668 1670 1671 1673 1673 1674 1675 1675 1677 1678 1680 1681 1683 1684 1685 1688 1688 1688 1688 1689 1692 1692 1695 1695 1696 7 1698	04C3 04C6 04C8 04CB 04CB 04CF 04D1 04D3 04D6 04D6 04D6 04D6 04DA 04DA	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05		<pre>i* Function: To send a message to the 3274 via the feature bus i* Input: i* i*</pre>
$\begin{array}{c} 1667\\ 1668\\ 1670\\ 1671\\ 1673\\ 1674\\ 1677\\ 1674\\ 1676\\ 1677\\ 1678\\ 1678\\ 1680\\ 1681\\ 1683\\ 1684\\ 1686\\ 1687\\ 16886\\ 1687\\ 1699\\ 1699\\ 1699\\ 1699\\ 1697\\ 1697\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\$	04C3 04C6 04C8 04C8 04CB 04CB 04CF 04D1 04D3 04D4 04D4 04D6 04D6 04D8 04DA	73 03 E9 030D R 83 7D 02 00 75 05 E9 030D R 0B C9 75 05 B0 07		<pre>i* Function: To send a message to the 3274 via the feature bus i* Input: i* Input: i* i*</pre>
$\begin{array}{c} 1667\\ 1668\\ 1670\\ 1671\\ 1673\\ 1677\\ 1673\\ 1674\\ 1677\\ 1678\\ 1676\\ 1678\\ 1680\\ 1681\\ 1682\\ 1684\\ 1688\\ 1684\\ 1688\\ 1689\\ 16991\\ 1692\\ 1699\\ 16992\\ 1699\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 1697\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16989\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 16990\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 10000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000$	04C3 04C6 04C8 04CB 04CF 04D1 04D3 04D4 04D4 04D6 04D6 04D6 04DF	73 03 E9 030D R 83 7D 02 00 75 05 E9 030D R 0B C9 75 05 B0 07 E9 030D R		<pre>i* Function: To send a message to the 3274 via the feature bus i* Input: i* Input: i* i*</pre>
$\begin{array}{c} 1667\\ 1668\\ 1670\\ 1671\\ 1673\\ 1674\\ 1677\\ 1674\\ 1676\\ 1677\\ 1678\\ 1678\\ 1680\\ 1681\\ 1683\\ 1684\\ 1686\\ 1687\\ 16886\\ 1687\\ 1699\\ 1699\\ 1699\\ 1699\\ 1697\\ 1697\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1697\\ 899\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\$	04C3 04C6 04C8 04CB 04CF 04C1 04D3 04D6 04D6 04D6 04D6 04DC D4DF	73 03 E9 030D R 83 7D 02 00 75 05 E9 030D R 0B C9 75 05 B0 07		<pre>i* Function: To send a message to the 3274 via the feature bus i* Input: i* i*</pre>
$\begin{array}{c} 1667\\ 1668\\ 1670\\ 1671\\ 1673\\ 1677\\ 1673\\ 1674\\ 1677\\ 1678\\ 1676\\ 1678\\ 1680\\ 1681\\ 1682\\ 1684\\ 1685\\ 1684\\ 1686\\ 1686\\ 1689\\ 1691\\ 1693\\ 1694\\ 1695\\ 1694\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1698\\ 1697\\ 1697\\ 1697\\ 1698\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 1697\\ 100\\ 100\\ 100\\ 100\\ 100\\ 100\\ 100\\ 10$	04C3 04C6 04C8 04CB 04CF 04D1 04D3 04D4 04D4 04D6 04D6 04D6 04D6 04D6 04D7 04D7	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05 B0 07 E9 030D R B8 R 88 R		<pre>i* Function: To send a message to the 3274 via the feature bus i* Input: i* Input: i* i*</pre>
$1667 \\ 1668 \\ 1670 \\ 1671 \\ 1673 \\ 1674 \\ 1676 \\ 1677 \\ 1676 \\ 1677 \\ 1678 \\ 1680 \\ 1681 \\ 1683 \\ 1683 \\ 1685 \\ 1686 \\ 1687 \\ 1692 \\ 1692 \\ 1695 \\ 1699 \\ 1695 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 1699 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 169 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 10$	04C3 04C6 04C8 04C8 04C8 04C7 04D1 04D3 04D6 04D6 04D6 04D6 04D6 04D6 04D6 04D6	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05 B0 07 E9 030D R B8 R 85 C0 89 16 0002 R		<pre>i* Function: To send a message to the 3274 via the feature bus i* Input: i* i Input: i* i*</pre>
$\begin{array}{c} 1667\\ 1668\\ 1670\\ 1671\\ 1673\\ 1674\\ 1674\\ 1676\\ 1677\\ 1678\\ 1676\\ 1678\\ 1680\\ 1681\\ 1682\\ 1684\\ 1686\\ 1687\\ 1689\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1700\\ 1702\\ 1703\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\ 1705\\$	04C3 04C6 04C6 04C8 04CB 04CB 04D1 04D3 04D6 04D6 04D6 04D6 04D6 04D6 04D6 04D6	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05 B0 07 E9 030D R B8 R 88 R		<pre>i Function: To send a message to the 3274 via the feature bus i Function: To send a message to the 3274 via the feature bus i Input:     (AL) = interrupt vector code     BX:DX = address of message     (CX) = length of message     (CY) = length of message     (CY) = length of message     (CX) = length of message     (CX) = length of message     (CY) = length</pre>
$\begin{array}{c} 1667\\ 1668\\ 1670\\ 1671\\ 1673\\ 1673\\ 1674\\ 1676\\ 1677\\ 1676\\ 1677\\ 1678\\ 1680\\ 1681\\ 1682\\ 1683\\ 1685\\ 1685\\ 1687\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1695\\ 1699\\ 1702\\ 1702\\ 1705\\ 1705\\ 1707\\ 1705\\ 1707\\ 1707\\ 1705\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\ 1707\\$	04C3 04C6 04C8 04C8 04CB 04CB 04C7 04D1 04D3 04D4 04D3 04D6 04D6 04D6 04D6 04D6 04D6 04D6 04D7 04E2 04E4 04E8 04E2	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05 B0 07 E9 030D R B0 07 E9 030D R B8 R 85 C0 89 16 0002 R 89 16 0002 R		<pre>i Function: To send a message to the 3274 via the feature bus i Function: To send a message to the 3274 via the feature bus i (AL) = interrupt vector code i BX:DX = address of message i (CX) = length of me</pre>
$\begin{array}{c} 1667\\ 1668\\ 1670\\ 1671\\ 1673\\ 1674\\ 1674\\ 1676\\ 1677\\ 1678\\ 1676\\ 1678\\ 1680\\ 1681\\ 1682\\ 1683\\ 1684\\ 1686\\ 1687\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1700\\ 1700\\ 1706\\ 1706\\ 1706\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1708\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\ 1008\\$	04C3 04C6 04C6 04CB 04CB 04CB 04D1 04D3 04D4 04D6 04D6 04D6 04D6 04D6 04D6 04D6	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05 B0 07 E9 030D R 80 07 E9 030D R 88 R 85 C0 89 16 0002 R 89 16 0002 R 89 16 0004 R 89 0E 0006 R 84 45 0E A2 00E R		<pre>i Function: To send a message to the 3274 via the feature bus i Input:     Input:     (AL) = interrupt vector code     BX:DX = address of message     (CX) = length of message     (CY) = length     (CY) = length of message     (CY) = length of</pre>
$\begin{array}{c} 1667\\ 1668\\ 1670\\ 1671\\ 1673\\ 1677\\ 1673\\ 1674\\ 1677\\ 1678\\ 1677\\ 1678\\ 1680\\ 1681\\ 1683\\ 1684\\ 1686\\ 1687\\ 1697\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1000\\ 100\\ 1$	04C3 04C6 04C8 04CB 04CB 04CF 04D1 04D3 04D6 04D6 04D6 04D6 04D6 04D6 04DF 04DF 04E2 04E2 04E4 04E6 04E6 04E6	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05 B0 07 E9 030D R B8 R 88 c0 89 16 0002 R 89 16 0002 R 89 16 0002 R 89 16 0002 R 89 16 0006 R 84 45 0E		<pre>i Function: To send a message to the 3274 via the feature bus i Function: To send a message to the 3274 via the feature bus i (AL) = interrupt vector code i BX:DX = address of message i (CX) = length of me</pre>
$\begin{array}{c} 1667\\ 1668\\ 1670\\ 1671\\ 1673\\ 1677\\ 1673\\ 1674\\ 1677\\ 1676\\ 1677\\ 1676\\ 1676\\ 1680\\ 1682\\ 1683\\ 1684\\ 1685\\ 1686\\ 1689\\ 1692\\ 1695\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1700\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\ 1000\\$	04C3 04C6 04C8 04CB 04CB 04C7 04D1 04D3 04D4 04D3 04D6 04D6 04D6 04D6 04D6 04D6 04D6 04D6	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05 B0 07 E9 030D R 88 R 88 C0 89 16 0002 R 89 16 0002 R 89 16 0002 R 89 26 0006 R 84 45 0E A2 000E R 89 3E 0053 R		<pre>Function: To send a message to the 3274 via the feature bus Input: Input:</pre>
1667 1668 1670 1671 1673 1674 1673 1674 1677 1677 1678 1680 1681 1683 1684 1685 1687 1689 1699 1699 1699 1699 1699 1699 1699	04C3 04C6 04C8 04C8 04CB 04C7 04D1 04D3 04D4 04D6 04D6 04D6 04D6 04D6 04D6 04D6	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05 B0 07 E9 030D R B8 R 85 C0 89 16 0002 R 89 16 0002 R 89 16 0002 R 89 0E 0006 R 84 45 0E A2 0005 R 89 3E 0053 R C6 06 0001 R 00		<pre>:* Function: To send a message to the 3274 via the feature bus :* Input: I</pre>
$\begin{array}{c} 1667\\ 1668\\ 1670\\ 1671\\ 1673\\ 1677\\ 1674\\ 1677\\ 1676\\ 1677\\ 1678\\ 1676\\ 1677\\ 1678\\ 1680\\ 1681\\ 1682\\ 1683\\ 1684\\ 1686\\ 1687\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1699\\ 1701\\ 1702\\ 1700\\ 1700\\ 1700\\ 1700\\ 1711\\ 1711\\ \end{array}$	04C3 04C6 04C8 04C8 04CB 04C7 04D1 04D3 04D4 04D6 04D6 04D6 04D6 04D6 04D6 04D6	73 03 E9 030D R 83 7D 02 00 75 05 B0 05 E9 030D R 0B C9 75 05 B0 07 E9 030D R B8 R 85 C0 89 16 0002 R 89 16 0002 R 89 16 0002 R 89 16 0002 R 89 16 0005 R 84 45 0E A2 000E R 89 3E 0053 R		<pre>Function: To send a message to the 3274 via the feature bus Input: Input:</pre>

112

111

	111			
1714	0504 E8 029A R	с с с	CALL	WAIT_FOR_POLL_ACK ; WAIT FOR THE PREVIOUS POLL REQUEST
1715 1716	0504 E8 0294 R	с ,		; TO BE ACKNOHLEDGED
1717	0507 BA 03E4			DX, COMMAND_PORT ; REQUEST A READ FROM THE 3274
1718 1719	050A BO 03 050C EE		10V 7UT	AL, READ_REGUEST
1720	0300 22	C		
1721	050D	C RESTART_		DX,FEA_HESSAGE_START ; SET UP TO TRANSHIT MESSAGE
1722	050D 8B 16 0002 R 0511 89 16 0008 R		HOV HOV	FBA_MESSAGE,DX ; TO THE 3274
1724	0515 88 16 0004 R	C I	HOV	DX,FBA_MESSAGE_START+2
1725	0519 89 16 DOUA R		MOV MOV	FBA_MESSAGE+2+DX CX+FBA_MESSAGE_SIZE
1726 1727	0510 88 0E 0006 R 0521 89 0E DODC R		HOY	FBA_MESSAGE_LENGTH.CX
1728		C		NAIT_FOR_READ ; WAIT FOR THE 3274 TO ACKNOWLEDGE THE
1729 1730	0525 E8 0546 R	с (	CALL	REQUEST FOR A READ SEQUENCE
1731		C		HDR_FLAGS,HDR_F_START ; INDICATE START OF MESSAGE
1732 1733	0528 26: C6 06 0003 R 80	с I с	MOV	NUK_FLAGS, NUK_F_START & INSTERSE OFART OF HERE
1734	052E	C SEND_MES		CX.FEA_MTSSAGE_LENGTH ; PICK UP REMAINING NEESAGE LENGT
1735 1736	052E 88 0E 000C R	с : с	MOV	
1730	0532 81 F9 00FC	С	CHP	CX, SIZE HOR_DATA ; WE CAN ONLY SEND 252 BYTES FER BLOCK
1738	0536 76 03		JNA NOV	SEND_MESSAGE_3 CX,SIZE HDR_DATA
1739 1740	0538 B9 DOFC	č		
1741	053B	C SEND_MES		FBA_MESSAGE_LENGTH,CX ; UPDATE REMAINING MESSAGE LENGTH
1742	053B 29 0E 0DOC R 053F 75 06		SUB JNZ	PEND MERCAGE & FINTS TS NOT THE LAST DECUN OF THE TWO
1744	0541 26: 80 DE 0003 R 40	C	OR	HDR_FLAGS, HDR_F_END   INDICATE LAST BLOCK OF MESSAGE
1745	ar ( 7	C SEND_MES	SAGE 4:	
1746 1747	0547 0547 86 E9	с _	XCHG	
1748	0549 26: 89 0E 0000 R	•	MOV	AL,FBA_MESSAGE_TYPE ; PUT MESSAGE TYPE INTO HEADER
1749 1750	054E AO 000E R 0551 26: A2 0002 R		MOV	HDR_TYPE,AL
1751	0555 86 E9	С	XCHG	CH,CL
1752 1753	0557 8C DB	C C	HOV	BX,DS 3 SAVE DATA SEGMENT POINTER
1754	0559 C5 36 0008 R	С	LDS	SI, FBA_MESSAGE ; POINT TO MESSAGE TEXT DI.HDR DATA ; POINT TO DATA AREA IN BUFFER
1755	0550 26: 80 3E 0004 R	C C	LEA CLD	DI,HDR_DATA ; POINT TO DATA AREA IN BOTTOR ; MOVE MESSAGE TO BUFFER
1756 1757	0562 FC 0563 F3/ A4	c	REP	NOVER
1758	0565 &E DB	C	HOV	DS,BX ; RESTORE DATA SEGMENT POINTER DS,BX ; UPDATE MESSAGE STARTING ADDRESS FBA_MESSAGE,SI ; UPDATE MESSAGE STARTING ADDRESS
1759 1760	0567 89 36 0008 R	C C	HOV	
1761	0568 E8 029A R	С	CALL	WAIT_FOR_POLL_ACK ; WAIT FOR THE PREVIOUS POLL REQUEST ; TO BE ACKNOWLEDGED
1762 1763	056E BA 03E4	C C	MOV	DY. COMMAND PORT ; INDICATE THAT THE BUFFER HAT NOW DE
1764	0571 B0 02	C	MOV	AL,READ_COMPLETE ; READ BT THE 3274
1765 1766	0573 EE	с с	OUT	DX,AL
1767	0574 F6 06 0000 R 08	c	TEST	FBA_STATE,F_DEFERRED_CANCEL ; SHOULD THIS MESSAGE BE CANCELLED?
1768	0579 74 11	C C	JZ	THE REPORT OF THE CONTINUE
1769 1770	0579 74 11 0578 83 3E 000C R 00	С	CMP	SEND_RESSAGE_S ; NO; CONTINUE FBA_MESSAGE_LENGTH.0 ; YES, WAS THE ENTIRE MESSAGE SENT SEND_MESSAGE_S ; NO, CONTINUE
1771 1772	0580 75 DA	с с	JNZ	
1773	0582 80 26 0000 R 3F	C	AHD	FBA_STATE,NOT (F_READ_PENDING OR F_READ_IN_PROGRESS) AL.RC_HESSAGE_CANCELLED ; INDICATE HESSAGE CANCELLED
1774	0587 B0 08 0589 E9 030D R	C C	HOV JMP	BIOS_ERROR ; RETURN TO APPLICATION PROGRAM
1775 1776	U384 E4 0300 K	C		
1777	058C	C SEND_ME C	ESAGE_5 CALL	
1778 1779	058C E8 0546 R	c	0-00	; READ SEQUENCE OR TO SIGNAL END OF
1780		c		3 MESSAGE
1781 1782	058F F6 06 0001 R 20	c C	TEST	FBA_PENDING_OP,F_EXECUTE ; IS THIS THE END OF THE
1783		ç	JIIZ	HESSAGE? SEND_MESSAGE_6 ; YES, THE ENTIRE MESSAGE HAS BEEN
1784 1725	0594 75 08	с с	5112	; RECEIVED BY THE 3274
1766		с ·	NOV	HDR_FLAGS,0 ; CLEAR START AND END BLOCK FLAGS
1787 1788	0596 26: C6 06 0003 R 00 059C EB 90	C C	JNP	SEND_HESSAGE_2 ; SEND NEXT BLOCK TO THE 3274
1789		с.		
1790	059E 059E 80 26 0000 R 7F	C SEND_ME C	SSAGE_6	FBA STATE, NOT F_READ_PENDING ; INDICATE THE READ THE
1791 1792	0372 00 20 0000 1 21	. C		BEEN COMPLETED
1793	6543 50 0311 B	с с	JHP	BIOS_EXIT ; RETURN TO CALLER
1794 1795	05A3 E9 0311 R	С		
1796		C C		ES:NOTHING
1797 1798		C ;	HAIT F	OR THE 3274 TO SEND A START OPERATION COMMAND
1799		C C WAIT_FO	READ	FROC NEAR
1800 1801	05A6 05A6	C WAIT_FO	DR_READ_	
1802	0546 80 3E 0001 R 00	с С	СМР	FBA_PENDING_DP.0 ; HAS A START OPERATION COMMAND BEEN ; RECEIVED?
1803 1804	05AB 74 F9	C C	JZ	WAIT_FOR_READ_1 ; NO, WAIT FOR THE START OP
1805		С	TECT	FBA_PENDING_OP,F_POR ; WAS A POWER ON RESET RECEIVED?
1806	05AD F6 06 D0D1 R 40	C	TEST	

113

.

### 114

05B2					114	
0202	74 0B	c		JZ	WAIT_FOR_READ_2 ; NO, CONTINUE	
		с С		POWER OF	N RESET RECEIVED - THE READ SEQUENCE HAS BEEN CANCELLED	
0584	80 24 0000 0	c	:			
0589		R7F C		AND	FBA_STATE,NOT F_READ_PENDING ; READ SEQUENCE IS FINISHE	
05BA		C	:	HOV	AL, RC_POWER_ON_RESET ; INDICATE POWER ON RESET RECEIVED	
05BC	E9 030D R	C C		JHP	BIOS_ERROR	
05BF 05BF	F6 06 0001 F	20 C		_READ_2 TEST		
0504	74 03	C	:	JZ	FBA_PENDING_OP,F_EXECUTE ; WAS AN EXECUTE RECEIVED? WAIT_FOR_READ_3 ; NO, CONTINUE	
		с С		EXECUTE	RECEIVED - THE ENTIRE MESSAGE HAS BEEN RECEIVED BY	
		C	;	THE 3274		
0506	EB 5D 90	C C		JHP	WAIT_FOR_READ_8 ; RETURN TO CALLER	
0509		С С				
	F6 06 0001 R	2 10 C	-	TEST	FBA_PENDING_OP,F_ABORT_E ; WAS AN ABORT TRANSMISSION	
05CE	74 04	C C		JZ	; RECEIVED? WAIT_FOR_READ_4 ; ND, CONTINUE	
		с с			RANSMISSION RECEIVED - RESTART THE ENTIRE TRANSMISSION	
		c	;	SEQUENCE		
05D0	58	с с		POP	AX ; REMOVE RETURN ADDRESS FROM THE STACK	
<b>0</b> 5D1	E9 D4FA R	C		JHP	RESTART_SEND_1 ; RESTART THE READ SEQUENCE	
<b>0</b> 5D4		C C	WAIT_FOR	_READ_4		
<b>0</b> 5D4	F6 06 0001 R	208 C		TEST	<pre>FBA_PENDING_OP,F_ABORT_V ; WAS AN ABORT TO LAST VERIFY ; RECEIVED?</pre>	
0509	74 09	С		JZ	; RECEIVED? WAIT_FOR_READ_5 ; NO, CONTINUE	
		c c	;	ABORT TO	O LAST VERIFY RECEIVED - RETRANSMIT THE ENTIRE MESSAGE	
05DB	80 26 0001 R	c		AND	FBA_PENDING_DP,NOT F_ABORT_V ; CLEAR START OP STATUS	
<b>0</b> 5E0	58	c		POP	AX ; REMOVE RETURN ADDRESS FROM THE STACK	
05E1	E9 050D R	c c		JMP	RESTART_SEND_2 ; RETRANSMIT THE MESSAGE	
05E4	E4 D4 0001 -	c	NAIT_FOR			
05E4 05E9	F6 D6 D001 R 74 DB	C		TEST JZ	FBA_PENDING_OP,F_ERROR ; WAS AN ERROR RECEIVED? WAIT_FOR_READ_6 ; NO, CONTINUE	
		с с			ECEIVED - INDICATE FROTOCOL VIOLATION FOR THE MESSAGE	
	66 61 A	C				
05EB 05F0	80 26 0000 R 58	7F C C		AND Pop	FBA_STATE,NOT F_READ_PENDING ; READ SEQUENCE IS FINISHEL AX ; REMOVE RETURN ADDRESS FROM THE STACK	
05F1 05F3	B0 07 E9 0300 R	C C	I	MOV JMP	AL, RC_PROTOCOL_VIOLATION ; INDICATE PROTOCOL VIOLATION	
	2 h	C			-	
05F6 05F6	F6 06 0001 R		WAIT_FOR		: FBA_PENDING_OP,F_CANCEL ; DID THE USER REQUEST TO CANCE	
Q5FB	74 28	c c			; THIS MESSAGE HAIT_FOR_READ_8 ; NO, CONTINUE	
	. =-	c				
		c c	1	INE APPL	LICATION PROGRAM CANCELLED THIS MESSAGE	
05FD	FA	с с	I	CLI	; INSURE NO INTERRUPTS OCCUR WHILE ; CHANGING STATE FLAGS	
05FE	80 26 0001 R	FD C		AND	FBA_PENDING_DP, NOT F_CANCEL + INDICATE CANCEL PROCESSED	
0603	F6 06 0000 R	C		TEST	FBA_STATE,F_READ_IN_FROGRESS ; IS THE MESSAGE ALREADY ; BEING TRANSMITTED TO THE 3274?	
	74 OF 83 3E 000C R	00 C			WAIT_FOR_READ_7 ; ND, THE MESSAGE CAN BE CANCELLED FBA_MESSAGE_LENGTH,0 ; WAS THE WHOLE MESSAGE ALREADY	
		C			SEND TO THE 3274?	
060F	74 08	C C		JZ	WAIT_FOR_READ_7 ; YES, THE MESSAGE CAN BE CANCELLED	
	80 05 0000 R				FBA_STATE,F_DEFERRED_CANCEL ; NO, CAN'T CANCEL THE ; MESSAGE UNTIL THE ENTIRE MESSAGE HAS	
	FR	L L	:	STI	; EEEH SENT TO THE 3274	
0616		C		JMP	WAIT_FOR_READ_1 ; WAIT FOR THE NEXT START OPERATION	
0616	FB EB 8D					
D616 D617 0619	EB 8D	с с с	WAIT_FOR	READ_7:		
D616 D617 0619		с с с	WAIT_FOR	READ_7:	FLA_STATE, NOT (F_READ_PE:.DING OF F_READ_IN_PROSPESS) ; READ SEQUENCE IS FINISHED	
D616 D617 0619 0619 0615	EB 8D 20 26 0000 R FB	C C C 3F C C C	WAIT_FOP	READ_7:	FLA_CTATE,NOT (F_READ_PE:DING OF F_PEAD_IN_PROSPESS) ; READ SEQUENCE IS FINISHED ; ENABLE INTERRUPTS AGAIN	
D616 D617 0619 0619 0619 061E 061F D620	EB 8D 80 26 0000 R FB 58 80 08	C C C 3F C C C C C C C C C C C C C C C C	WAIT_FOP_	READ_7:	FLA_STATE, NOT (F_READ_PE:.DING OF F_READ_IN_PROSPESS) ; READ SEGUENCE IS FINISHED ; ENABLE INTERRUPTS AGAIN AX ; REMOVE RETURN ADDRESS FROM THE STACK AL,RC_MESSAGE_CANCELLED ; INDICATE MESSAGE CANCELLED	
D616 D617 0619 0619 0619 061E 061F D620	EB 8D 26 26 0000 R FB 58	C C C 3F C C C C C C C C C C C C C C C C	WAIT_FOP_	READ_7:	FLA_STATE:HOT (F_READ_PE:DING OF F_READ_IN_PROSPESS) ; READ SEQUENCE IS FINISHED ; ENABLE INTERRUPTS AGAIN AX ; REMOVE RETURN ADDRESS FROM THE STACK	
D616 D617 0619 0619 0619 0619 0615 0620 0622 0625	EB 8D 60 26 0000 R FB 58 80 08 E9 030D R	2 3 5 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	WAIT_FOP	READ_7:	FLA_STATE; NOT (F_READ_PE:DING OF F_READ_IN_PROSPESS) ; READ SEQUENCE IS FINISHED ; READ SEQUENCE IS FINISHED ; ENABLE INTERRUPTS AGAIN AX ; REMOVE RETURN ADDRESS FROM THE STACK AL,RC_MESSAGE_CANCELLED ; INDICATE MESSAGE CANCELLED EIOS_ERROR ; RETURN TO APPLICATION PROGRAM	
D616 D617 0619 0619 0619 0619 0615 0620 0622 0625	EB 8D 80 26 0000 R FB 58 80 08	C C C C C C C C C C C C C C C C C C C	WAIT_FOP	READ_7:	FLA_STATE; NOT (F_READ_PE:DING OR F_READ_IN_PROSPESS) ; READ SEQUENCE IS FINISHED ; ENABLE INTERRUPTS AGAIN AX ; REMOVE RETURN ADDRESS FROM THE STACK AL,RC_MESSAGE_CANCELLED ; INDICATE MESSAGE CANCELLED EIOS_ERROR ; RETURN TO APPLICATION PROGRAM	
0616 0617 0619 0619 0619 0619 0619 0619 0619 0620 0625 0625	EB 8D 20 26 0000 R FB 50 08 E9 030D R 80 26 0001 R	2 3 F C C C C C C C C C C C C C C C C C C	WAIT_FOR	READ_7:	FLA_STATE; NOT (F_READ_PE:DING OR F_READ_IN_PROSPESS) ; READ SEQUENCE IS FINISHED ; READ SEQUENCE IS FINISHED ; ENABLE INTERRUPTS AGAIN AX ; REMOVE RETURN ADDRESS FROM THE STACK AL,RC_MESSAGE_CANCELLED ; INDICATE MESSAGE CANCELLED EIOS_ERROR ; RETURN TO APPLICATION PROGRAM FBA_PENDING_OP; NOT F_READ ; YES, CAN COMPLETE READ ; SEQUENCE	
D616 D617 0619 0619 0619 0615 0615 0622 D625	EB 8D 20 26 0000 R FB 50 08 E9 030D R 80 26 0001 R	C C C C C C C C C C C C C C C C C C C	WAIT_FOR	READ_7:	FLA_STATE, NOT (F_READ_PE:DING OF F_READ_IN_PROSPESS) ; READ SEQUENCE IS FINISHED ; READ SEQUENCE IS FINISHED ; ENABLE INTERRUPTS AGAIN AX ; REMOVE RETURN ADDRESS FROM THE STACK AL,RC_MESSAGE_CANCELLED ; INDICATE MESSAGE CANCELLED EIOS_ERROR ; RETURN TO APPLICATION PROGRAM FBA_PENDING_OP,NOT F_READ ; YES, CAN COMPLETE READ ; SEQUENCE ; RETURN TO CALLER	
D616 D617 0619 0619 0616 0616 0620 0622 D625 0625	EB 8D 20 26 0000 R FB 50 08 E9 030D R 80 26 0001 R	C C C C C C C C C C C C C C C C C C C	WAIT_FOP	READ_7:	FLA_STATE, NOT (F_READ_PE:DING OF F_READ_IN_PROSPESS) ; READ SEQUENCE IS FINISHED ; READ SEQUENCE IS FINISHED ; ENABLE INTERRUPTS AGAIN AX ; REMOVE RETURN ADDRESS FROM THE STACK AL,RC_MESSAGE_CANCELLED ; INDICATE MESSAGE CANCELLED EIOS_ERROR ; RETURN TO APPLICATION PROGRAM FBA_PENDING_OP,NOT F_READ ; YES, CAN COMPLETE READ ; SEQUENCE ; RETURN TO CALLER	

4,641,262

115	4,641,262 <b>116</b>
	C ;####################################
	C ;* C ;* Procedure Name: CANCEL_MESSAGE
	C :* C :* C :* Function: To cancel a message waiting to be sent to the 3074
	C ;*
	C ;* Input: C ;*
	C ;* (AL) = interrupt vector code C ;*
	C ;* Output: None C ;*
	C 。 3 安远在水水就就在水水水水水水水水水水水水水水水水水水水水水水水水水水水水水水水水
62B	C C CANCEL_MESSAGE:
628 E8 0664 R	C CALL FIND_VECTOR ; IDENTIFY WHICH MESSAGE IS BEING C ; SPECIFIED
62E 73 03 630 E9 0311 R	C JNC CANCEL_MESSAGE_1 ; VALID IDENTIFIER C JMP BIOS_EXIT ; ERROR - INVALID MESSAGE IDENTIFIER C
3 3 F6 06 0000 R 80 6 74 08	C CANCEL_MESSAGE_1: C TEST FBA_STATE,F_READ_PENDING ; IS A READ REQUEST PENDING? C JZ CANCEL_MESSAGE_2 ; NO, IGNORE CANCEL REQUEST C
3A 3B 3E 0053 R 3E 75 05	C CMP DI,FBA_SEND_VECTOR ; IS THIS MESSAGE STILL BEING SENT? C JNE CANCEL_MESSAGE_2 ; NO, IT WAS ALREADY SENT TO THE 3274
0 80 DE 0001 R 02	C C OR FBA_PENDING_OP,F_CANCEL ; YES, INDICATE THAT THE USER C ; WANTS TO CANCEL THE MESSAGE
E9 0311 R	C C CANCEL_MESSAGE_2: C JMP BIOS_EXIT ; RETURN TO CALLER C
	C INCLUDE NDSPAPI3.ASM C C
	— C ]#****#*******************************
	C ;* Procedure Name: MODIFY_XLAT_IN
	C ;* C ;* Function: To modify the inbound keyboard translation table
	C ;* C ;* Input:
	C ;* C ;* (CH) = 3278/3279 relative key number
	C 1# (CL) = 3278/3279 keyboard case
	C (H (DH) = Fersonal Computer keyboard scan code or BIOS function cod C (H (DL) = Personal Computer keyboard case
	C )* C ;* Output:
	C ;# C ;# (AL) = error code if parameters are invalid
	C ;# CF = 0 if parameters are valid C ;# 1 if parameters are invalid
	C 3*
	С ; , , , , , , , , , , , , , , , , , ,
548 548 E8 0000 E	C MODIFY_XLAT_IN: C CALL MOD_XLAT_IN ; MODIFY KEYBOARD TRANSLATION TABLE
4B 72 03 4D E9 0311 R	C JC MODIFY_XLAT_IN_1 C JMP BIOS_EXIT ; PARAMETERS WERE VALID
50 50 E9 030D R	C MODIFY_XLAT_IN_1: C JMP BIOS_ERROR ; PARAMETERS WERE INVALID
	c _
	C ;*
	C ;* Procedure Name: MODIFY_XLAT_OUT C ;*
	C :* Function: To modify the outbound keyboard translation table C ;*
	C ;* Input: C ;*
	C ;* (CH) = encoded keystroke value
	C ;* (DH) = 3278/3279 relative key number C ;* (DL) = 3278/3279 keyboard case
	C ;* C ;* Output:
	C ;* C ;* (AL) = error code if parameters are invalid
	C :* CF = 0 if parameters are valid
	C ;* 1 if parameters are invalid C ;*
	C ;* 1 if parameters are invalid
	C ;* 1 if parameters are invalid C ;* C ; <del>****</del> ********************************
EA 0000 E 72 03	C ;* 1 if parameters are invalid C ;* C ;* C ;* C MODIFY_XLAT_OUT: C CALL MOD_XLAT_OUT ; MODIFY KEYBOARD TRANSLATION TABLE C JC MODIFY_XLAT_OUT_1
EN 0000 E 72 D3 E9 D311 R	C ;* 1 if parameters are invalid C ;* C ;* C ;**************************
3 3 ES 0000 E 6 72 03 8 E9 0311 R B B E9 030D R	C ;* 1 if parameters are invalid C ;* C ;* C ;**************************

~

-.

•

117	118
	C ;* C ;* Function: To return the identifier of the current kzyboard
	C ;* translation table
	C 3#
	C ;# Input: None C ;#
	C ;* Output:
	C 3M C 3*    BX on stack = keyboard translation table identifier
	C   3 我我我就我就会来来我我我我我我我我我我我我我我我我我我我我我我我我我我我我我我
5/ FT	C C QUERY_KEYBOARD_IDENTIFIER:
065E 065E E8 0000 E	
0661 E9 0311 R	C CALL QUERY_KEYBOARD_ID ; GET KEYBOARD TRANSLATION TABLE ID C JMP BIOS_EXIT : PETURN TO CALLER
	C
	c .
	C
	C 3*
	C ;* Procedure Name: FIND_VECTOR C ;*
	C ;* Function: To find an interrupt vector
	C ;*
	C 3* Input: C 3*
	C ;* (AL) = interrupt vector code
	C ;# Output: C ;#
	C }# (AL) = error code if interrupt vector code is invalid
	C ;# (DI) = address of interrupt vector C ;# CF = 0 if interrupt vector code is valid
	C ;* 1 if interrupt vector code is invalid or the feature
	C ;# bus adapter is incperative
	C }*** C }***
	C
0664	C FIND_VECTOR PROC NEAR C TEST FBA_STATE,F_FBA_NOT_DP ; IS THE FEATURE BUS ADAPTER
0664 F6 06 0000 R D1	C TEST FBA_STATE,F_FBA_NOT_DP; IS THE FEATURE BUS ADAFTER C ; OPERATIONAL?
0669 74 05	C JZ FIND_VECTOR_1 ; YES, CONTINUE
066B B0 03	C HOV AL,RC_FBA_NOT_DP ; NO, INDICATE FBA IS INOPERATIVE
066D EB 23 90	C JMP FIND_VECTOR_2 ; RETURN TO CALLER
	C
0670 0670 80 3E 000F R	C FIND_VECTOR_1: C LEA DI,DATA_XFER_VECTOR ; SET UP PROPER INTERRUPT
0674 0A CD	C OR AL,AL ; VECTOR ADDRESS
0676 74 1B	
0678 80 3E 0020 R 067C FE C8	C LEA DI,SCREEN_CAPTURE_VECTOR C DEC AL
067E 74 13	C JZ FIND_VECTOR_3
0680 8D 3E 0031 R	C LEA DI, WRAP_TEST_VECTOR
0684 FE C8 0686 74 08	C DEC AL C JZ FIND_VECTOR_3
0688 80 3E 0042 R	Č LEA DI,QUERY_STATE_VECTOR
068C FE C8	C DEC AL C JZ FIND_VECTOR_3
068E 74 03	r – –
0690 B0 05	C MOV AL,RC_INVALID_VECTOR ; INVALID INTERRUPT VECTOR CODE
0692 0692 F9	C FIND_VECTOR_2: C STC ; INDICATE INVALID INTERRUPT VECTOR CODE
	C
0693	C FIND_VECTOR_3:
0693 C3	C RET C
0694	C FIND_VECTOR ENDP
	C
0694	NDSP_API ENDP
0694	NDSPBIOS ENDS
	END
	-

APPENDIX D

PAGE D-1

-

The IBM Personal Computer MACRO Assembler 3278/3279 Attachment BIO5 End Label

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Specifications \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Module Name: NDSPBEND Descriptive Name: 3278/3279 Attachment BIOS extension end label Copyright: IBM Corp 1982 Status: Version 1.00 Function: This label marks the physical end of the 3278/3279 attachment BIOS extension ; Notes: Dependencies: None Restrictions: No code or data which is needed by the 3278/3279 interrupt handlers may follow this module when 3270ATT is linked together ; Nodule Type: Program Processor: Assembler ; Entry Point: BIOS\_END Purpose: See function Linkage: None Input Parameters: None Exit Normal: None Exit Error: None ; External References: Procedure Invoked: None Control Blocks: None ; Messages Issued: None : Change Activity: None , ; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Specifications \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NDSPBIOS SEGMENT PUBLIC 'NDSPBIOS' 0000 PUBLIC BICS\_END ; LAST BYTE IN 3278/3279 ATTACHMENT BIOS CODE BIOS\_END LABEL BYTE 0000 NDSPBIOS ENDS 0000 END

,

. .

•• ••

۰.

APPENDIX E

The IBM Personal Computer MACRO Assembler 3278/3279 Attachment Initialization

٠

PAGE	· E-1
------	-------

6		
7	;	**************************************
9 10	;	
11 12	1	; ; Descriptive Name: 3278/3279 attachment initialization
13 14	1	; Copyright: IBM Corp 1982
15 16		; Status: Version 1.00
17 18 19	:	; ; Function: To iniitialize the 3278/3279 Attachment Option and the ; 3278/3279 Attachment BIOS extension (interrupt handlers)
20 21	:	i i Notes:
22 23	1	· · · · · · · · · · · · · · · · · · ·
24 25	i	Restrictions: None
26 27		; Hodule Type: Program
28 29	1	Processor: Assembler
30 31	1	; ; Entry Point: ND5P_INIT
32 33	1	; ; Purpose: See function
34 35		; ; Linkage: Call
36 37		
36 39	i	; ; (DS) = address of DOS Program Segment Prefix
40 41	i	; ; Exit Normal:
42 43 44 45		; Purpose: To fix the 3278/3279 Attachment interrupt handlers in ; storage and return to DOS
45 46 47		; Linkage: INT 27H
48		Output:
49 50		; (DX) = offset to end of interrupt handler
51 52		; Exit Error: None
53 54		; External References:
55 56		; Procedure Invoked: None
57 58		; Control Blocks:
59 60 61 62		; NDSPDCLS - 3278/3279 Attachment Option interfaces ; NDSPUSER - 3278/3279 Attachment BIOS extension interfaces ;
63 64	•	Messages Issued:
65 66 67 68		<ul> <li>1901 - 3278 adapter not operational.</li> <li>1902 - 3278 Keyboard not attached - typewriter assumed.</li> <li>1903 - 3278 Keyboard not supported by the specified language - typewriter assumed.</li> </ul>
69 70		; 1904 - 3278 adapter failure - cannot communicate with host ; system.
71 72		; ; Change Activity: None
73 74		; ; **************************** End of Specifications **************************
75 76		SUBTTL WORK AREAS
77 78		; INCLUDE NDSPDCLS. 450
79 80		INCLUDE NDSPUSER.ASM .LIST
81 82		。我想想我在这些家族在不能多很需要不是在家族的家族在这些不是不是不是不是我们在这个家族的不是是不是不不能不不能不是不是不是不是不是不是不是不是不是不是
83 84		<pre>;*     IBM PERSONAL COMPUTER DOS PROGRAM SEGMENT FREFIX ;* </pre>
85		· 我我我就在少年就是发生的最高级的名字的是这些家家的是这些的是这些不能有不能要有这些不是不是我们不可能不可能不不不不不不不不不不能要不不不可。" \$ \$
87	000	PSP SEGMENT AT 0 INT20 LABEL BYTE ; INT 20H INSTRUCTION (USED TO RETURN
	000	; TO DOS)
	000	PSP ENDS
93		

r.

2       2:2:2:2:2:2:2:2:2:2:2:2:2:2:2:2:2:2:2:	123	4,641,262 <b>124</b>
0         1         DIA AFAS USE D THE END BIOS ROUTINES           0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0		
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0		<pre>;* DATA AREAS USED BY THE ROM BIOS ROUTINES ;*</pre>
2       2722       RESET_FLEBOND       226       : HORD-1234H IF KETBOARD RESET UNDERAUT         0       40 (       27       : HORD-1234H IF KETBOARD RESET UNDERAUT         77       1       27       : CATE PARAMETERS SET BY MODE COMMAND         10       ROMDATA HUS       : CATE PARAMETERS SET BY MODE COMMAND         1       0.011 () : CATE PARAMETERS SET BY MODE COMMAND         1       0.012 () : CATE PARAMETERS SET BY MODE COMMAND         1       0.012 () : CATE PARAMETERS SET BY MODE COMMAND         1       0.013 () : CATE PARAMETERS FOR 3270 DISPLAY         2       36 08 20 05 1A 00       0 : STATUS DIS         1       6(45 CF1 CONTROLLER PARAMETERS FOR 3270 DISPLAY         2       36 08 20 05 1A 00       D3 () : CATE, CAT, CAH, CAH, CAH, CAH, CH, CH, CH, CH, CH, CH, CH, CH, CH, C	0000	ROMDATA SEGMENT AT 40H
0       40 1       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       000       00	0072 0072 ????	ORG 72H
27       1         00       ROMDATA ENDS         1       RAMDATA ENDS         1       RAMDATA ENDS     <	0090 0090 40 [	DRG 90H
Description         Description           0         27         3NITDATA SEGMENT "NOSPINIT" K0_DD 00         28         0         28         5XITDATA SEGMENT "NOSPINIT" K0_DD 00         28         0         5XITDATA SEGMENT "NOSPINIT" K0_DD 00         28         5XITDATA SEGMENT "NOSPINIT" K0	??	
0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0         0	0000	ROHDATA ENDS
0       1*       DATA AREAS         0       27       DATA AREAS         1       00       DD       DEFENT         0       1       DD       DEFENT       NETDATA SECRENT         0       1       DD       DE       1         0       0       1       STATUS       DB       1         1       00       DD       DE       1       STATUS       DB         1       00       DD       DE       1       STATUS       DE       1         2       30       26       22       05       1A       00       DD       DCM.07H.0EH.07H.0EH.0DH.19H.19H 1       40x25       TEXT THOLEE         2       15       57       0A       1A       DD       DD       DCM.07H.0EH.0CH.0CH.0CH.0CH.0CH.0CH       DD         2       15       55       70       1A       DD       DD       DCM.07H.0EH.0CH.0CH.0CH.0CH.0CH       DD         2       15       55       70       1A       DD       DD       DD       DD       DD       DCM.0CH.0CH.0CH.0CH.0CH.0CH.0CH.0CH.0CH.0CH		
0       1       INITIATA SEGMENT "MOSPILIT"         0       1       100       2         0       1       100       2         0       1       1       1         0       1       1       1         0       1       1       1       1         0       1       1       1       1       1         0       1       1       1       1       1       1         0       1       1       1       1       1       1       1         1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1		IN DATA AREAS
0       2:       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 000       1 0000       1 0000       1 000       1 00		
1       00       37ATUS DB 0       ; 37ATUS DTS         000       ST_FB_NOT_OP       EQU BON       ; FEATURE BUS ADAPTER NOT OPERATIONAL         ;       6145 CFT CONTROLLER PARAMETERS FOR 3278 DISPLAY         2       38 28 28 05 1A 00       D0       D0         1       10       00       CH.45 CFT CONTROLLER PARAMETERS FOR 3278 DISPLAY         2       38 28 28 05 1A 00       D0       CH.47.00H.00H.00H.00H.00H.00H.00H.00H.00H.00	0000 0000 77	
2       38       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28       28	0001 00 = 0080	STATUS DE O ; STATUS BITS
2         36         26         26         26         26         26         26         26         26         26         26         26         27         16         16         16         16         16         16         16         16         16         16         16         17         16         16         17         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16 </td <td></td> <td>; 60:45 CFT CONTROLLER PARAMETERS FOR 3278 DISPLAY</td>		; 60:45 CFT CONTROLLER PARAMETERS FOR 3278 DISPLAY
19       19       19         20       00       60       60       00         27       15       57       04       14       00       03       711.501.577.044.14.001.001.001.001         27       15       57       04       14       00       03       711.501.577.044.14.001.001.001.001.001         27       15       57       04       14       00       03       711.501.577.044.14.001.001.001.001.001         27       15       57       04       04       07       00       00       05         28       28       28       05       68       00       15       351.201.001.001.001.001.001.001.001         29       16       07       00       00       05       021.011.061.071.001.001.001.001.001.001         20       00       05       05       05       05       05       00       05         20       00       05       05       05       05       06       07       00       05         21       16       16       16       16       05       04       07       01.01.001.001.001.001.001.001         21       16       05       04       07       00<	0002 0002 38 28 28 05 1A 00	
19       19       19         19       19       02       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00 <t< td=""><td>19 19 0004 02 07 06 07 00 00</td><td></td></t<>	19 19 0004 02 07 06 07 00 00	
A       02       07       06       07       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00	0012 71 50 57 0A 1A 00	D3 71H,50H,57H,0AH,1AH,00H,19H,19H ; 80×25 TEXT MODE
64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64 <td< td=""><td>001A 02 07 06 07 00 00</td><td>CB 02H,07H,06H,07H,00H,00H,00H,00H</td></td<>	001A 02 07 06 07 00 00	CB 02H,07H,06H,07H,00H,00H,00H,00H
A       02       01       05       07       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00	0022 38 28 28 05 68 00	[B 32H,28H,2BH,05H,68H,00H,64H,64H ; GRAFHICS MODE
19 18       02 00 08 0C 00 00       DB       02H,0DH,0CH,00H,0CH,0OH,0DH,0DH         2       58 28 28 05 14 00       12 00       DB       02H,0DH,0CH,0CH,0CH,0OH,0DH,0DH         19 19       02 07 06 07 00 00       DB       02H,0TH,0CH,0TH,0CH,0CH,0CH,0CH,0CH,0CH         2       71 50 56 0A 1A 00       DB       71H,5CH,5CH,0CH,0TH,0CH,0CH,0CH,0CH         2       71 50 56 0A 1A 00       DB       71H,5CH,5CH,0CH,0TH,0CH,0CH,0CH         2       71 50 56 0A 1A 00       DB       71H,5CH,5CH,0CH,0TH,0CH,0CH,0CH         2       71 50 56 0A 1A 00       DB       71H,5CH,5CH,0CH,0TH,0CH,0CH,0CH         2       71 50 56 0A 1A 00       DB       71H,5CH,5CH,0CH,0CH,0CH,0CH,0CH         2       71 50 56 0A 1A 00       DB       71H,5CH,5CH,0CH,0CH,0CH,0CH,0CH         2       71 50 56 0A 1A 00       DB       71H,5CH,5CH,0CH,0CH,0CH,0CH,0CH         2       71 50 56 0A 1A 00       DB       71H,5CH,5CH,0CH,0CH,0CH,0CH,0CH         2       71 50 56 0A 1A 00       DB       71H,5CH,5CH,0CH,0CH,0CH,0CH,0CH         2       71 60 77 06 07 00 00       DB       32H,2CH,2BH,0SH,6EH,0CH,0CH,0CH,0CH         2       71 50 52 0F 19 00       DB       63H,5CH,5CH,0FH,19H,0CH,19H,1BH ; 80x25 HONOCHROME         19 18       A0 20 00 08 0C 00 00       DB       <	002A 02 01 06 07 00 00	DB 02H,01H,06H,07H,00H,00H,00H,00H
A       02       0D       0B       02H,0DH,0BH,0CH,0DH,0DH,0DH         I       6845       CRT CONTROLLER PARAMETERS FOR 3279 DISPLAY         I       6845       CRT CONTROLLER PARAMETERS FOR 3279 DISPLAY         PARME_DI79       L4DEL BYTE         DB       02H,0TH.06H,0TH.00H,0TH.00H,0H,0H         I       19       19         A       02       07       06       07         00       00       DB       02H,0TH.06H,0TH.00H,0TH.00H,0H       0H         2       71       50       56       0A       1A       00         19       19       19       19       19       19       19         2       07       06       07       00       00       DB       71H,50H,56H,0AH,1AH,0OH,0H,0OH,0OH       0H         2       71       50       56       0A       DB       02H,07H,06H,07H,0OH,0OH,0OH,0OH       0DE         2       38       26       2B       05       68       0D       DB       38H,22H,2EH,05H,6EH,0OH,6HH,6HH; GRAPHICS MODE         64       64       64       64       64       64       64       64       64         62       01       64       64       64	0032 63 50 52 OF 19 00 19 18	DB 63H,50H,52H,0FH,19H,00H,19H,16H ; 80×25 MONDCHROME
PARHS_D179       LADEL BYTE         D3       381,28,28       28 05 14 00         19       19         D2       381,28,28,28         D2       381,28,28,28         D2       381,28,28,28         D2       381,28,28         D2       381,28,28         D2       00         D3       00         D4       00         D5       01         D5       02         D6       07         D6       07         D6       07         D6       07         D7       00         D8       02         D9       08         D9       08         D9       08         D9       08         D9       08	003A 02 0D 0B 0C 00 00 00 00	DB 02H, 0DH, 0BH, 0CH, 00H, 00H, 00H, 00H
2       38       28       28       05       14       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00		: 6845 CRT CONTROLLER PARAMETERS FOR 3279 DISPLAY
19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       19       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10 <td< td=""><td></td><td>PARMS_DU79_LABEL BYTE</td></td<>		PARMS_DU79_LABEL BYTE
00       00       00         2       71       50       56       0A       1A       00       DB       71H,50H,56H,0AH,1AH,0OH,19H,19H ; 80x25 TEXT HODE         19       19       19       19       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00 <td>19 19</td> <td></td>	19 19	
19       19       19         4       02       07       06       07       00       00         20       00       00       00       00       00       00         2       36       26       28       05       68       00       00       00         2       36       26       28       05       68       00       00       00       00         2       36       26       28       05       68       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00 <t< td=""><td></td><td>UK 02H,07H,06H,07H,00H,00H,00H,00H</td></t<>		UK 02H,07H,06H,07H,00H,00H,00H,00H
A       C2       07       06       07       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00	0052 71 50 56 0A 1A 00	DB 71H,50H,56H,0AH,1AH,00H,19H,19H ; 80x25 TEXT MODE
64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       64       70       64       65       74       65       72       64       64       74       64       64       74       64       74       64       74       64       74       64       74       64       64       74       64       64       74       64       74       64       74       64       64       74       64       64       74       64       64       74       64       64       74       64       64       64       64       64       64 <td< td=""><td>005A C2 D7 06 D7 C0 00</td><td>DB D2H, 07H, 06H, 07H, 00H, 00H, 00H, 00H</td></td<>	005A C2 D7 06 D7 C0 00	DB D2H, 07H, 06H, 07H, 00H, 00H, 00H, 00H
A       02       01       06       07       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00	0062 38 28 28 05 68 00	DB 38H,28H,2BH,05H,66H,00H,64H,64H ; GRAPHICS MODE
19       18       02       0D       0B       02H,0DH,0BH,0CH,00H,00H,00H,00H,00H         00       00       00       i       i       ERROR MESSAGES         000       00       00       0H       ;       LINE FEED CHARACTER         000       00       0H       i       CARRIAGE RETURN CHARACTER         000       00       0H       ;       CARRIAGE RETURN CHARACTER         000       03       32       37       38       20         20       33       32       37       38       20         61       64       70       74       65         72       20       6E       6F       74       65         72       20       6E       6F       74       20         6F       70       65       72       61       64         60       66       66       62       E       DB       CR.LF.,*\$'         9       31       39       30       32       20       MSG_HO_KEYBOARD       DB       '1902 - 3278 keyboard not attached'         68       65       79       62       66       61       61	006A 02 01 06 07 00 00	DE 02H,01H,06H,07H,00H,00H,00H,00H
A       02       0D       0B       DC       D0       DD       DB       02H,0DH,0BH,0CH,00H,00H,00H,00H         00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00 <td>0072 63 50 52 0F 19 00</td> <td>DB 63H,50H,52H,0FH,19H,00H,19H,18H ; 80×25 MONOCHROME</td>	0072 63 50 52 0F 19 00	DB 63H,50H,52H,0FH,19H,00H,19H,18H ; 80×25 MONOCHROME
; ERROR MESSAGES 200A LF EQU DAH ; LINE FEED CHARACTER 200D CR EQU 0DH ; CARRIAGE RETURN CHARACTER 20 33 32 37 38 20 61 64 61 70 74 65 72 20 6E 6F 74 20 6F 70 65 72 61 74 69 6F 6E 61 6C 2E 5 0D 0A 24 DB CR.LF.'\$' 31 39 30 32 20 2D MSG_NO_KEYBOARD DB '1902 - 3278 keyboard not attached' 20 33 32 37 38 20- 6B 65 79 62 6F 61	007A 02 0D 0B DC D0 00 00 00	DB 02H, 0DH, 0BH, 0CH, 00H, 00H, 00H, 00H
DOD         CR         EQU         ODH         ; CARRIAGE RETURN CHARACTER           2         31 39 30 31 20 2D         MSG_ADAPTER_NOT_OP         DB '1901 - 3278 adapter not operational.'           20 33 32 37 38 20         61 64 61 70 74 65         72 20 6E 6F 74 20         65 72 63 74           67 66 56 72 63 74         69 6F 6E 61 6C 2E         DB CR.LF.'\$'           9 31 39 30 32 20 2D         MSG_NO_KEYBDARD         DB '1902 - 3278 keyboard not attached'           20 33 32 37 38 20*         66 67 79 62 6F 61         67 61		ERROR MESSAGES
20 33 32 37 38 20 61 64 61 70 74 65 72 20 6E 6F 74 20 6F 70 65 72 61 74 69 6F 6E 61 6C 2E DB CR.LF.'\$' 31 39 30 32 20 2D MSG_NO_KEYBOARD DB '1902 - 3278 keyboard not attached' 20 33 32 37 38 20* 6B 65 79 62 6F 61	= 000A = 000D	
72 20 6E 6F 74 20 6F 70 65 72 6J 74 69 6F 6E 6J 6C 2E 00 0A 24 01 31 39 30 32 20 2D MSG_HO_KEYBOARD DB '1902 - 3278 keyboard not attached' 20 33 32 37 38 20* 6B 65 79 62 6F 61		
31 39 30 32 20 2D MSG_NO_KEYBOARD DB '1902 - 3278 keyboard not attached' 20 33 32 37 38 20 6B 65 79 62 6F 61	72 20 6E 6F 74 20 6F 70 65 72 6] 74 69 6F 6E 61 6C 2E	
		MSG_NO_KEYBOARD DB '1902 - 3278 keyboard not attached'

•

		4,641,20			
	125				126
	20 61 74 74 61 63				
0004	68 65 64 20 20 20 74 79 70		DB	• - type	writer assumed.'
	65 77 72 69 74 65				
	72 20 61 73 73 75 6D 65 64 2E				
00E0	0D 0A 24		DB	CR,LF,'	it and have a supported by '
0023	31 39 30 33 20 2D 20 33 32 37 38 20	MSG_INVALID_KEYBOARD	08	1903 -	3278 keyboard not supported by '
	6B 65 79 62 6F 61				
	72 64 20 6E 6F 74				
	20 73 75 70 70 6F 72 74 65 64 20 62				
	79 20		DB	the co	cified language'
109	74 68 65 20 73 70 65 63 69 66 69 65		00	the sp	
	64 20 6C 61 6E 67				
11F	75 61 67 65 20 2D 20 74 79 70		DB	' - typ	ewriter assumed.'
	65 77 72 69 74 65				
	72 20 61 73 73 75 6D 65 64 2E				
	DD 0A 24		DB	CR,LF,'	5' 3278 adapter failure - cannot '
138	31 39 30 34 20 2D 20 33 32 37 38 20	MSG_FBA_NOT_OP	DB	1904 -	3276 adapter failure - damist
	61 64 61 70 74 65				
	72 20 66 61 69 60				
	75 72 65 20 2D 20 63 61 6E 6E 6F 74				
	20		58	100000	icate with host system."
0150	63 6F 6D 6D 75 6E 69 63 61 74 65 20		00	COnnul	
	77 69 74 68 20 68				
	6F 73 74 20 73 79 73 74 65 6D 2E				
017A	0D DA 24		DB	CR,LF,'	5 '
		INITDATA ENDS			
170					
					<b>水水小产业水水就在其些有点就水在就水在就水就是在这大块有有少水水都在于有大的小水</b> 。
		3.演 3.演奏表示,是我又是不是不是不是有些要求。	******	*	
		STACK AREA			LIZATION ROUTINE
		STACK AREA			
		;# STACK AREA ;# ;#################################	*****	*******	<b>这个自主我们在不能没有不能没有不能要不是要要要要要要要要要要要要要要要要</b>
		INITSTCK SEGMENT PA	*****	********* LCK 'STAC	<b>这个自主我们在不能没有不能没有不能要不是要要要要要要要要要要要要要要要要</b>
	20 [ 27??	;# STACK AREA ;# ;#*********************************	******	********* LCK 'STAC	<b>这个自主我们在不能没有不能没有不能要不是要要要要要要要要要要要要要要要要</b>
		INITSTCK SEGMENT PA	******	********* LCK 'STAC	<b>这个自主我们在不能没有不能没有不能要不是要要要要要要要要要要要要要要要要</b>
0000	77??	INITSTCK SEGMENT PA	******	********* LCK 'STAC	<b>这个自主我们在不能没有不能没有不能要不是要要要要要要要要要要要要要要要要</b>
0000	77??	INITSTCK ENDS	RA STJ DUP(?	********* *CK 'STAC	К , <del>х р о «м к и и и и и и и и и и и и и и и и и и </del>
0000	77??	* STACK AREA ** ********************************	RA STJ DUP(?	********* *CK 'STAC	在》《龙安斯·莱尔斯·斯斯·斯斯·斯斯·斯斯·斯斯·斯斯·斯斯·斯斯·斯斯·斯斯·斯斯·斯斯·斯
000	77??	INITSTCK ENDS	RA STJ DUP(?	LCK 'STAC	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
0000	77??	<pre>im STACK AREA 1 imitstck segment pa         DW 32 INITSTCK ENDS imitstck ends imi</pre>	RA STJ DUP(?	LCK 'STAC	**************************************
0000	77??	INITSTCK ENDS	******* RA ST/ DUP(? ******	LCK *STAC	**************************************
000	77??	INITSTCK         SEGMENT         PA           INITSTCK         SEGMENT         PA           DW         32           INITSTCK         ENDS           IMITSTCK         ENDS           IMITSTCK         AREAS           IMITSTCK         STATA	RA ST/ DUP(? (****** WITHI ******	LEK 'STAC	**************************************
0000	77??	INITSTCK SEGMENT PA DW 32 INITSTCK ENDS INITSTCK ENDS INITSTCK AREAS INITSTCK AREAS INITSTCK ENDS INITSTCK ENDS	RA ST/ DUP(? (****** WITHI ******	LEK 'STAC ) N OTHER 1 NOSPBIO: YRIGHT:B' :FAR	K' NODULES 3276/3279 ATTACHMENT BIOS CODE 5 INTERRUPT HANDLER FOR 3276/3279
0000	77??	INITSTCK SEGMENT PA DW 32 INITSTCK ENDS INITSTCK ENDS INITSTCK AREAS INITSTCK ENDS INITSTCK ENDS	RA STJ DUP(? WITHI WITHI BLIC SP_COP SP_INT	LCK 'STAC	**************************************
000	77??	INITSTCK SEGMENT PA DW 32 INITSTCK ENDS INITSTCK ENDS INIT	RA ST/ DUP(? WITHI MITHI JBLIC SP_COP SP_INT SP_API		K' 10DULES 13276/3279 ATTACHMENT BIOS CODE 14 JOURS 15 J COPYRIGHT NOTICE 15 INTERRUPT HANDLER FOR 3276/3279 17 JOAPTER 11 JITERRUPT HANDLER FOR 3278/3279 10 JUTERRUPT S
0000	77??	INITSTCK SEGMENT PA DH 32 INITSTCK ENDS INITSTCK ENDS INIT	RA ST/ DUP(? WITHI MITHI JBLIC SP_COP SP_INT SP_API		**************************************
0000		INITSTCK SEGMENT PA DH 32 INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INDSPBIOS SEGMENT PI EXTRN NDI EXTRN NDI EXTRN NDI EXTRN NDI EXTRN NDI	RA STJ DUP(? WITRI BLIC SP_COP SP_INT SP_API OS_END	LCK 'STAC W OTHER P W OTHER P	K' ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES
0000		INITSTCK SEGMENT PA DH 32 INITSTCK SEGMENT PA DH 32 INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS EXTRN NDI EXTRN NDI	RA SIJ DUP(? WITHI BLIC SP_COP SP_INT SP_API OS_ENE OS_ENE	KEY STAC	K' ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES
0000 0000 0000		INITSTCK SEGHENT PA DH 32 INITSTCK ENDS INITSTCK ENDS	RA SIJ DUP(? HITHI JBLIC SP_COP INT SP_INT SP_API OS_ENC CSDATA	NOTHER P NOTHER P NOTHER P NOTHER P NOTHER P STANDAR STANDAR STANDAR NOTHER P NOTHER P NOTHE	K' MODULES S'; 3276/3279 ATTACHMENT BIOS CODE TTE; COPYRIGHT NOTICE ; INTERRUPT HANDLER FOR 3276/3279 ; ADAPTER ; INTERRUPT HANDLER FOR 3276/3279 ; INTERRUPT HANDLER FOR 3278/3279 ; INTERUPT HANDLER FOR 3278/3279 ; INTERRUPT HANDLER FOR 3278/3279 ; IN
0000 0040 0000		INITSTCK SEGHENT PA DW 32 INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS EXTRN NDS EXTRN NDS	RA STJ DUP(? HITHI BLIC SP_COP SP_INT SP_API OS_ENC CSDATA SEGMEN	N OTHER ? N OTHER ? NOTHER ? YRIGHT:B :FAR :FAR :FAR KEYBOAR T FUBLII	K' S' ; 3276/3279 ATTACHMENT BIOS CODE (TE ; COPYRIGHT NOTICE ; INTERRUPT HANDLER FOR 3276/3279 ; INTERRUPT HANDLER FOR 3276/3279 ; INTERRUPT HANDLER FOR 3276/3279 ; INTERRUPT HANDLER FOR 3278/3279 ; INTERRUPT HANDLER F
0000 0040 0000 0000		INITSTCK SEGMENT PA DW 32 INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INDSPBIOS SEGMENT PI EXTRN NDI EXTRN NDI EXTRN NDI EXTRN NDI EXTRN NDI BIOSDATA GROUP BI BIOSDATA GROUP BI BIOSDATA_KEYDARD <sup>I</sup> EXTRN XL	RA STJ DUP(? WITHI BLIC SP_COP SP_INT SP_API OSS_ENE OSDATJ ASSEME AT_TAL AT_TAL	N OTHER ? N OTHER ? NOTHER ? YRIGHT:B :FAR :FAR :FAR KEYBOAR T FUBLII	K' MODULES S'; 3276/3279 ATTACHMENT BIOS CODE TTE; COPYRIGHT NOTICE ; INTERRUPT HANDLER FOR 3276/3279 ; ADAPTER ; INTERRUPT HANDLER FOR 3276/3279 ; INTERRUPT HANDLER FOR 3278/3279 ; INTERUPT HANDLER FOR 3278/3279 ; INTERRUPT HANDLER FOR 3278/3279 ; IN
0000 0040 0000		INITSTCK SEGHENT PA DW 32 INITSTCK SEGHENT PA DW 32 INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS EXTRN NDS EXTRN NDS EXTRN NDS BIOSDATA GROUP BI BIOSDATA GROUP BI BIOSDATA_KEYBOARD EXTRN XL EXTRN XL BIOSDATA_KEYBOARD	RA STJ DUP(? WITHI BLIC SP_COP SP_INT SP_API OSS_ENC OSDATJ SEGMEH AT_TAL X_TAL ENDS	N OTHER ? N OTHER ? N OTHER ? NOSPBIC: YRIGHT:B' :FAR :FAR :FAR :FAR LES:BYTE SLES:END:	K' MODULES S'; 3276/3279 ATTACHMENT BIOS CODE YTE; COPYRIGHT NOTICE ; INTERRUPT HANDLER FOR 3276/3279 ; ADAPTER ; INTERVPT HANDLER FOR 3278/3279 ; INTERVACE SOFTWARE INTERRUPTS ; END OF 3278/3279 ATTACHMENT BIOS D,BIOSDATA_FBA 'hDSPBIOS'; KEYBOARD BIOS DATA AREAS ; KEYBOARD TRANSLATION TABLE BYTE; END OF KEYBOARD TRANSLATION TAB
0000 0040 0000 0000		INITSTCK SEGHENT PA DW 32 INITSTCK SEGHENT PA DW 32 INITSTCK ENDS INITSTCK ENDS EXTRN NDI EXTRN NDI EXTRN NDI EXTRN NDI EXTRN SI BIOSDATA GROUP BI BIOSDATA_KEYBOARD EXTRN XL EXTRN XL EXTRN XL	RA STJ DUP(? UP) WITHI BLIC SP_COP SP_INT SP_API OS_ENE OSDATJ SEGMEH AT_TAI AT_TAI ENDS	ICK 'STAC	K' ADULLES ADULLES ADULLES ADULES ADULES ADULES ADULES ADUTER INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3276/3
0000 0400 0000 0000 0000 0000		INITSTCK AREA INITSTCK SEGHENT PA DW 32 INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INITSTCK ENDS INDSPBIOS SEGMENT PI EXTRN NDI EXTRN NDI EXTRN NDI EXTRN NDI EXTRN NDI BIOSDATA GROUP BI BIOSDATA GROUP BI BIOSDATA KEYBOARD EXTRN XL EXTRN XL BIOSDATA FBA SEGME EXTRN FE	RA STJ DUP(? UP) WITHI BLIC SP_COP SP_INT SP_API OS_ENE OSDATJ SEGMEH AT_TAL ENDS STAT PUIL	ICK 'STAC	K' MODULES S'; 3276/3279 ATTACHMENT BIOS CODE (TE ; COPYRIGHT NOTICE ; INTERRUPT HANDLER FOR 3276/3279 ; ADAPTER ; INTERRUPT HANDLER FOR 3276/3279 ; INTERRUPT HANDLER FOR 3278/3279 ; INTERRUPT HANDLER FOR 3400 ; FEATURE BUS ADAPTER STATE ; FEATURE BUS ADAPTER STATE
00000 0040 00000 00000 00000	???? 1	INITSTCK SEGHENT PA DW 32 INITSTCK SEGHENT PA DW 32 INITSTCK ENDS INITSTCK ENDS EXTRN NDI EXTRN NDI EXTRN NDI EXTRN SEINOS BIOSDATA GROUP BI BIOSDATA_KEYBOARD EXTRN XL EXTRN SEGME EXTRN FR	RA STJ DUP(? UP) WITHI BLIC SP_COP SP_INT SP_API OS_ENE OSDATJ SEGMEH AT_TAL ENDS STAT PUIL	ICK 'STAC	K' MODULES S'; 3276/3279 ATTACHMENT BIOS CODE (TE ; COPYRIGHT NOTICE ; INTERRUPT HANDLER FOR 3276/3279 ; ADAPTER ; INTERRUPT HANDLER FOR 3276/3279 ; INTERRUPT HANDLER FOR 3278/3279 ; INTERRUPT HANDLER FOR 3400 ; FEATURE BUS ADAPTER STATE ; FEATURE BUS ADAPTER STATE
0000 0040 0000 0000 0000 0000 0000 000	???? ]	INITSTCK AREA INITSTCK SEGMENT PA DW 32 INITSTCK ENDS INITSTCK ENDS INIT	RA STJ DUP(? WITHI BLIC SP_COP SP_INT SP_API OSS_ENT OSS_ENT OSS_ENT AT_TAL AT_TAL AT_TAL SEGMETJ SEGMETJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ OSSATJ O	N OTHER ? N OTHER ? N OTHER ? YRIGHT:B' :FAR :FAR :FAR :FAR :FAR :ESPTE SLES:BYTE SLES_END: SLES 'NDS	K' ADDULES ADDULES ADDULES ADDULES ADDITER INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER
	???? ]	<ul> <li>STACK AREA 1</li> <li>STACK AREA 1</li> <li>STACK AREA 1</li> <li>STACK AREA 1</li> <li>STABLES SEGMENT PA DH 32</li> <li>INITSTCK ENDS</li> <li>INITST</li></ul>	RA SIJ DUP(? WITHI WITHI JBLIC SSP_COP SSP_COP SSP_INT SSP_API OS_ENC SSP_API SSP_COP SSP_INT SSP_API SSP_COP SSP_INT TAL SEGMEF AT_TAL SEGMEF AT_TAL SEGMEF AT_TAL SI SU OII SA_TAT_TAL	IN OTHER I NOTHER I NOTHER I INDSPBIC: YRIGHT:B' IFAR IFAR IFAR IFAR IFAR IFAR IFAR IFAR	K' ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3278/3279 INTERRUPT HANDLER FOR 3278/3279 INTERUPT HANDLER FOR 3278/3279 INTERRUPT HANDLER FOR 3278/327
	???? ]	<ul> <li>STACK AREA 1</li> <li>STACK AREA 1</li> <li>STACK AREA 1</li> <li>STACK AREA 1</li> <li>STABLES SEGMENT PA DH 32</li> <li>INITSTCK ENDS</li> <li>INITST</li></ul>	RA SIJ DUP(? WITHI WITHI JBLIC SSP_COP SSP_COP SSP_INT SSP_API OS_ENC SSP_API SSP_COP SSP_INT SSP_API SSP_COP SSP_INT TAL SEGMEF AT_TAL SEGMEF AT_TAL SEGMEF AT_TAL SI SU OII SA_TAT_TAL	IN OTHER I NOTHER I NOTHER I INDSPBIC: YRIGHT:B' IFAR IFAR IFAR IFAR IFAR IFAR IFAR IFAR	K' ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3278/3279 INTERRUPT HANDLER FOR 3278/3279 INTERUPT HANDLER FOR 3278/3279 INTERRUPT HANDLER FOR 3278/327
	???? ]	<ul> <li>STACK AREA 1</li> <li>STATA AREA 1</li> <li>STATA AREAS</li> <li>STABLES SEGMENT PL</li> <li>EXTRN XL</li> <li>EXTRN XL</li> <li>EXTRN AREAS</li> <li>STABLES SEGMENT PL</li> <li>EXTRN XL</li> <li>EXTRN XL</li> <li>EXTRN XL</li> <li>EXTRN AREAS</li> </ul>	RA SIJ DUP(? WITHI WITHI JBLIC SSP_COP SSP_COP SSP_INT SSP_API OS_ENC SSP_API SSP_COP SSP_INT SSP_API SSP_COP SSP_INT TAL SEGMEF AT_TAL SEGMEF AT_TAL SEGMEF AT_TAL SI SU OII SA_TAT_TAL	IN OTHER I NOTHER I NOTHER I INDSPBIC: YRIGHT:B' IFAR IFAR IFAR IFAR IFAR IFAR IFAR IFAR	K' ADDULES ADDULES ADDULES ADDULES ADDULES ADDITER INTERRUPT HANDLER FOR 3276/3279 ADAPTER INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3278/3279 INTERRUPT BUS ADAPTER DATA AREAS FEATURE BUS ADAPTER STATE FEATURE BUS ADAPTER NOT OPERATIONA KEYBOARD TRANSLATION TABLES KORD ; ARRAY OF POINTERS TO TRANSLATION
	7777 1 0 1	<ul> <li>STACK AREA 1</li> <li>STABLES ENDS</li> </ul>	RA STJ DUP(? WITHI BLIC SP_COP SP_INT SP_API OS_ENC OSDATJ SEGHEH AT_TAL AT_TAL AT_TAL ENDS INT PUU DI BLIC C_AT_TA PQ_TAB	N OTHER P N OTHER P N OTHER P N OTHER P N OTHER P STATE STAR STAR STAR STAR STAR STAR STAR STAR	K' ADDULES ADDULES ADDULES ADDULES ADDULES ADDITER INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR AREAS KEYBOARD TRANSLATION TABLES WORD ; ARRAY OF POINTERS TO TRANSLATION I TABLES I TRANSLATION TABLE UPDATES FOR AN RPG
0000 0040 0000 0000 0000 0000 0000 000	7777 1 0 1	<ul> <li>STACK AREA 1</li> <li>STABLES ENDS</li> </ul>	RA STJ DUP(? WITHI BLIC SP_COP SP_INT SP_API OS_ENC OSDATJ SEGHEH AT_TAL AT_TAL AT_TAL ENDS INT PUU DI BLIC C_AT_TA PQ_TAB	N OTHER P N OTHER P N OTHER P N OTHER P N OTHER P STATE STAR STAR STAR STAR STAR STAR STAR STAR	K' ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES ADDULES INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3276/3279 INTERRUPT HANDLER FOR 3278/3279 INTERRUPT HANDLER FOR 3278/3279 INTERUPT HANDLER FOR 3278/3279 INTERRUPT HANDLER FOR 3278/327

.

### 4,641,262

128

278 279 2e0 NDSPINIT SEGMENT 'NDSPINIT' 0000 281 "我这就要吃这家老家不是你不是我就就在我就不不是你就不要我就是这些我就不能不要不是我不是我不是我就不是我要就有我来要我要不是不是我的吗?" 282 263 ;\* ;\* Procedure Name: NDSP\_INIT 264 285 ;\* Function: To initialize the 3278/3279 Attachment Option and interrupt handlers 286 287 ;\* ;\* ;\* 288 289 290 ;\* Input: ;\* (CS) = address of NDSPINIT segment (DS) = address of DDS Program Segment Prefix 291 14 292 : # 293 **;** # 294 295 18 Output: :# (DX) = offset to end of interrupt handler code 296 ;\* ;\* 298 299 300 0000 NDSP\_INIT PROC FAR 301 CS:NDSPINIT SETUP\_EXIT ; SET UP FOR RETURN TO DOS INIT\_VECTORS ; INITIALIZE INTERRUPT VECTORS INIT\_6845 ; INITIALIZE THE PARAMETERS FOR THE 68 ; DISPLAY PROCESSOR BASIC\_ASSURANCE\_TEST ; TEST 3278/3279 ADAPTER INIT\_1 ; ADAPTER IS NOT INSTALLED INIT\_ADAPTER ; INITIALIZE 3278/3279 ADAPTER INIT\_XLAT ; INITIALIZE XEYBOARD TRANSLATION TABL ENABLE\_INTERRUPT ; ENABLE INTERRUPTS FROM THE 3278 ; ADAPTER CS:NDSPINIT ASSUME 302 303 0000 E8 0019 R CALL 304 0003 E8 0035 R CALL 305 E8 00C6 R CALL 0006 306 CALL 307 0009 E8 0040 R 72 09 E8 010E R E8 016E R JC CALL 308 0000 309 310 000E CALL 0011 311 0014 E8 0166 R CALL 312 313 RETURN TO DOS AND LEAVE THE INTERRUPT HANDLER RESIDENT 314 \$ 315 316 0017 INIT\_1: ; RETRIEVE END OF PROGRAM OFFSET POP DX 317 0017 5A ; RETURN TO DOS 318 0018 CB RET 319 320 0019 NDSP\_INIT\_ENOP 321 SUBITL Set up for the return to DOS 322 323 324 325 3 藏琴旅客摆放冷雨放下去不在中有分子不可有有个!在今,这么个有个个个个个个人的不可能是我就不是不是不是不是不是不是不能不能不能不能不能不能。 326 ;# ;# Procedure Name: SETUP\_EXIT 328 :\* Function: To set up for the exit from the initialization routine :\* 329 330 ;× back to DOS :\* 331 332 . Input: 333 ;× 334 ;\* (DS) = address of Program Segment Prefix 335 ;\* Output: 336 ;\* 337 34 return instruction in the PSP set to INT 27H 338 :# address of return instruction put on the stack offset of end of interrupt handler code put on the stack 339 :# 340 ;× 341 342 ;\* 343 344 SETUP\_EXIT PROC NEAR 0019 345 ASSUME DS:PSP 346 ; SAVE SUBROUTINE RETURN ADDRESS 347 0019 58 POP BX 348 ; CHANGE RETURN INSTRUCTION TO INT 27H 349 001A C6 06 0001 R 27 MOV INT20+1,27H 350 FUSH ; FUT ADDRESS OF INT 27H INSTRUCTION 001F 1E DS 351 352 33 D2 XOR xa,xa ON STACK 0020 ; PUSH 353 0022 52 DX 354 DX, SEG BIDS\_END ; PUT OFFSET TO END OF INTERRUPT HANDLEF 355 0023 BA -- E MOV DH STACK 0026 8C DS HOV AX, DS ; 356 SUB HOV 357 0028 26 D0 DX,AX B1 04 D3 E2 CL,4 358 007A 359 SAL DX.CL 0020 DX, OFFSET BIOS\_END 360 002E 81 C2 0000 E 4DD 0032 52 PUSH DХ 361 362 ; PUT SUBROUTINE RETURN ADDRESS BACK PUSH вх 363 364 0033 53 ; ON STACK ; RETURN TO MAINLINE ROUTINE 365 0034 63 RET ASSUME DS:NOTHING 366 367 SETUP\_EXIT ENDP 368 0035 369 SUBTTL Initialize 3278/3279 Attachment interrupt vectors 370

•

371			
372 373			· · · · · · · · · · · · · · · · · · ·
374			j¥ · · · · · · · · · · · · · · · · · · ·
375 376			;* Procedure Name: INIT_VECTORS ;*
377			:* Function: To initialize the interrupt vectors used by the
378			* 3278/3279 Attachment interrupt handlers
379 380			;*
381			3 · · · · · · · · · · · · · · · · · · ·
382			;* Output: *
383 384			;* interrupt vector GA initialized for NDSP_INT *
385			<pre>interrupt vector 50 initialized for NOSP_API *</pre>
386			。 那些老师就是这家家家就就不完成这些的家族是我家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家家
387 388			
389	D035		INIT_VECTORS PROC NEAR
390 391	0075	BA R	MOV DX,NDSPBIOS ; PICK UP ADDRESS OF BIOS CODE SEGMENT
392	0038	BE DA	HOW DE DY
393		BA DODO E	HOV DX, OFFSET NDSP_INT ; PICK UP ADDRESS OF HARDWARE ; INTERRUPT HANDLER
394		<b>DD D</b>	HOY AL, CAH ; INITIALIZE INTERRUPT VECTOR CA
395 396		B0 0A B4 25	MOV AH,25H
397		CD 21	INT 21H
398 300	0047	BA 0000 E	MOV DX, OFFSET NDSP_API ; PICK UP ADDRESS OF SOFTWARE
399 400	0043	58 0000 E	: INTERRUPT HANULER
401		B0 50	MOV AL, NDSP_INT_CODE ; INITIALIZE INTERRUPT VECTOR
402		B4 25 CD 21	MOV AH,25H INT 21H
403 404	GUNA		
405	004C	C3	RET ; RETURN TO MAINLINE ROUTINE
406	0040		INIT VECTORS ENDP
407 408	0040		
409			
410			* 我我就要你出去了自己有不多的事实的是要是不可能不有的现在分词现在不能能不能是我就能是我的是我就能能能能能能能不能要不不不不不不不不。
411 412			
413			* Procedure Name: BASIC_ASSURANCE_TEST
414			;# ;# Function: To verify that the 3278/3279 Attachment Option is
415			
416			
416 417			;* operational ;*
417 418			;* operational ;* ;* Input: None
417 418 419			;* operational ;* ;* Input: None ;*
417 418			;* operational ;* ;* Input: None ;* ;* Output: ;*
417 418 419 420 421 422			;* operational ;* ;* Input: None ;* ;* Dutput: ;* ;* CF = 0 if the 3278/3279 adapter is installed
417 418 419 420 421 422 423			<pre>;* operational ;* ;* Input: None ;* ;* ;* Output: ;* ;* ;* ;* CF = 0 if the 3278/3279 adapter is installed ;* 1 if the 3278/3279 adapter is not installed ;* FFA NOT OP = 1 if the 3278/3209 adapter is installed ;* FFA NOT OP = 1 if the 3278/3278 adapter is installed</pre>
417 418 419 420 421 422			;* operational ;* ;* Input: None ;* ;* Dutput: ;* ;* CF = 0 if the 3278/3279 adapter is installed
417 418 419 420 421 422 423 424 425 426			<pre>;* operational ;* ;* input: None ;* ;* Output: ;* ;* CF = 0 if the 3278/3279 adapter is installed ;* 1 if the 3278/3279 adapter is not installed ;* F_FBA_NOT_OP = 1 if the 3278/3279 adapter is installed and ;* the feature bus interface is not operational ;*</pre>
417 418 419 420 421 422 423 424 425 426 427			<pre>;* operational ;* i* Input: None ;* i* Output: ;* ;* ;* CF = 0 if the 3278/3279 adapter is installed ;* 1 if the 3278/3279 adapter is not installed ;* F_FBA_NOT_OP = 1 if the 3278/3279 adapter is installed and ;* the feature bus interface is not operational </pre>
417 418 419 420 421 422 423 424 425 426	004D		<pre>;* operational ;* imput: None ;* ;* Input: None ;* ;*    Dutput: ;*    CF = 0 if the 3278/3279 adapter is installed ;*</pre>
417 418 420 422 422 422 422 425 426 425 426 428 429 429 430			<pre>;* operational ;* imput: None ;* * * * * * * * * * * * * * * * * * *</pre>
417 418 419 420 421 422 423 424 425 424 425 426 427 428 429 430 431	004D	BA R BE DA	<pre>;* operational ;* imput: None ;* ;* Input: None ;* ;*    Dutput: ;*    CF = 0 if the 3278/3279 adapter is installed ;*</pre>
417 418 420 422 422 422 422 425 426 425 426 428 429 429 430	004D 0050	8E DA	<pre>if operational if if operational if if operational if if Input: None if if Uutput: if if 0utput: if CF = 0 if the 3278/3279 adapter is installed if 1 if the 3278/3279 adapter is not installed if F_FBA_NOT_OP = 1 if the 3278/3279 adapter is installed if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if the feature bus interface is not operation</pre>
$\begin{array}{c} 417\\ 418\\ 419\\ 420\\ 421\\ 422\\ 422\\ 422\\ 425\\ 425\\ 426\\ 425\\ 426\\ 431\\ 431\\ 432\\ 433\\ 434\\ \end{array}$	004D 0050 0052	8E DA BA 03E5	<pre>if operational if if operational if if Input: None if if Uutput: if if Output: if if CF = 0 if the 3278/3279 adapter is installed if F_FBA_NOT_OP = 1 if the 3278/3279 adapter is installed if the feature bus interface is not operational if if the SITURANCE_TEST PROC NEAR ASSUME DS:INITDATA ; ESTABLISH ADDRESSABILITY TO DATA ARE/ MOV DX,INITDATA HOV DX,START_OP_MOD ; CLEAR THE START OPERATION MODIFIER WOV DX START_OP_MOD ; CLEAR THE START OPERATION MODIFIER WOV DX START_OP_MOD ; CLEAR THE START OPERATION MODIFIER WOV DX START_OP_MOD ; CLEAR THE START OPERATION MODIFIER WOV DX START_OP_MOD ; CLEAR THE START OPERATION HER WOV DX START_OP_MOD ; CLEAR THE START OPERATION</pre>
417 418 419 420 421 422 423 424 425 425 425 425 428 428 431 431 432 433	004D 0050	8E DA BA 03E5	<pre>if operational if if operational if if operational if if Input: None if if Uutput: if if 0utput: if CF = 0 if the 3278/3279 adapter is installed if 1 if the 3278/3279 adapter is not installed if F_FBA_NOT_OP = 1 if the 3278/3279 adapter is installed if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if the feature bus interface is not operation</pre>
$\begin{array}{c} 417\\ 418\\ 420\\ 421\\ 422\\ 422\\ 422\\ 422\\ 422\\ 422\\ 422$	004D 0050 0052 0055	8E DA 8A 03E5 EC	<pre>image: provide the second second</pre>
$\begin{array}{c} 417\\ 418\\ 420\\ 421\\ 422\\ 422\\ 422\\ 422\\ 422\\ 425\\ 426\\ 427\\ 428\\ 431\\ 432\\ 433\\ 435\\ 435\\ 435\\ 436\\ 438\\ \end{array}$	004D 0050 0052 0055 0056	8E DA BA 03E5 EC BA 03E4	<pre>if operational if if operational if if input: None if if uput: if if Output: if if cF = 0 if the 3278/3279 adapter is installed if if the 3278/3279 adapter</pre>
$\begin{array}{c} 417\\ 418\\ 420\\ 421\\ 4221\\ 4224\\ 4224\\ 4225\\ 4225\\ 4226\\ 431\\ 433\\ 4335\\ 4335\\ 4335\\ 4336\\ 738\\ 439\\ \end{array}$	004D 0050 0052 0055 0056 0056	8E DA BA 03E5 EC BA 03E4 EC	<pre>image: provide the second second</pre>
$\begin{array}{c} 417\\ 418\\ 420\\ 421\\ 422\\ 422\\ 422\\ 422\\ 422\\ 425\\ 426\\ 427\\ 428\\ 431\\ 432\\ 433\\ 435\\ 435\\ 435\\ 436\\ 438\\ \end{array}$	004D 0050 0052 0055 0056 0059 005A	8E DA BA 03E5 EC BA 03E4	<pre>image: provide the second second</pre>
$\begin{array}{c} 417\\ 418\\ 420\\ 4221\\ 4224\\ 4224\\ 4224\\ 4225\\ 4225\\ 4226\\ 4226\\ 4331\\ 4334\\ 4335\\ 4336\\ 78\\ 9\\ 442\\ 442\\ 4334\\ 5356\\ 78\\ 4339\\ 9\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\$	004D 0050 0052 0055 0056 0059 005A	BE DA BA 03E5 EC BA 03E4 EC 3C FF	<pre>image: provide the second second</pre>
$\begin{array}{c} 417\\ 418\\ 420\\ 4221\\ 4224\\ 4224\\ 4224\\ 4225\\ 44225\\ 44226\\ 44227\\ 4429\\ 4331\\ 233\\ 4335\\ 556\\ 789\\ 4433\\ 4335\\ 4336\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 44$	004D 0050 0052 0055 0056 0059 005A	BE DA BA 03E5 EC BA 03E4 EC 3C FF	<pre>if operational if if input: None if if</pre>
$\begin{array}{c} 417\\ 418\\ 420\\ 4221\\ 4224\\ 4224\\ 4224\\ 4225\\ 4225\\ 4226\\ 4226\\ 4331\\ 4334\\ 4335\\ 4336\\ 78\\ 9\\ 442\\ 442\\ 4334\\ 5356\\ 78\\ 4339\\ 9\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\$	004D 0050 0052 0055 0056 0059 005A 005C	8E DA BA 03E5 EC BA 03E4 EC 3C FF 74 2E	<pre>if operational if if input: None if if Input: None if if Output: if if CF = 0 if the 3278/3279 adapter is installed if I if the 3278/3279 adapter is not installed if F_FBA_NOT_OP = 1 if the 3278/3279 adapter is installed and if the feature bus interface is not operational if if the feature bus interface is not operational if if the SSURE DS:INITDATA is ESTABLISH ADDRESSABILITY TO DATA ARE/ HOV DX,START_OP_MOD is CLEAR THE START OPERATION MODIFIER IN AL.DX is REGISTER TO INSURE THAT THE ADAPTEF if AL.DX is THAT THE ADAPTER STATUS - FF INDICATE IN AL.DX is THAT THE ADAPTER STATUS - FF INDICATE IN AL.OFFH JE BAT_2 if TEST THE MEMORY ON THE 3278/3279 ADAPTER ASSUME FS:FBA BUFFER is ESTABLISH ADDRESSABILITY TO FEATURE </pre>
$\begin{array}{c} 417\\ 441\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 443\\ 33\\ 44\\ 43\\ 33\\ 44\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444$	004D 0050 0052 0055 0056 0059 005A 005C	8E DA BA 03E5 EC BA 03E4 EC 3C FF 74 2E BA R	<pre>if operational if if input: None if if Input: None if if Output: if if CF = 0 if the 3278/3279 adapter is installed if f=FBA_NOT_OP = 1 if the 3278/3279 adapter is installed if f=FBA_NOT_OP = 1 if the 3278/3279 adapter is installed and if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if if the feature bus interface is not operational if the feature bus feature is interface is not operational if the feature bus interface is not operational if the feature bus interface is not operational if the feature bus feature is interface is not operational if the feature bus feature is interface is not operational if the feature bus feature is interface is not operational if the feature bus feature bus feature is interface is not operational if the feature bus feature is interface is not operational if the feature bus feature is interface is not installed if the feature bus feature bus feature is interface is not installed if the feature bus f</pre>
$\begin{array}{c} 417\\ 4412\\ 44221\\ 44221\\ 44225\\ 44225\\ 44225\\ 44225\\ 44225\\ 4444444444444444444$	004D 0050 0052 0055 0056 0059 005A 005C	8E DA BA 03E5 EC BA 03E4 EC 3C FF 74 2E	<pre>if operational if input: None if if</pre>
$\begin{array}{c} 417\\ 441\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 442\\ 443\\ 33\\ 44\\ 43\\ 33\\ 44\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444\\ 444$	004D 0050 0052 0055 0056 0059 005A 005C	8E DA BA 03E5 EC BA 03E4 EC 3C FF 74 2E BA R	<pre>if operational if operational if Input: None if Output: if Output: if CF = 0 if the 3278/3279 adapter is installed if 1 if the 3278/3279 adapter is installed if 1 if the 3278/3279 adapter is installed if F_FBA_NOT_OP = 1 if the 3278/3279 adapter is installed and if the feature bus interface is not operational if the feature bus interface is not installed if the feature bus interface</pre>
$\begin{array}{c} 417\\ 4444444444444444444$	004D 0050 0052 0055 0056 0059 005A 005C 005E 0061 0063 0066	8E DA BA 03E5 EC BA 03E4 EC 3C FF 74 2E BA R 8E C2 BA 03E4 BO DC	<pre>if operational if if input: None if if input: None if if output: if if if output: if if if the 3278/3279 adapter is installed if if the 3278/3279 adapter is not installed if if the 3278/3279 adapter is installed and if if the 3278/3279 adapter is installed and if if the istrate bus interface is not operational if if if the istrate istrate istrate is installed and if if it is installed is installed if is installed and if if istrate istrate is installed is installed and if if istrate istrate is installed is installed and if if istrate istrate is installed and if if istrate istrate is installed and if istrate istrate is installed and if istrate istrate is installed and if istrate istrate istrate is installed and if istrate istrate istrate is installed and if istrate istrate istrate istrate is installed and if istrate is installed istrate istrate istrate is installed istrate istrate istrate. If is istrate istrate istrate istrate is istrate is istrate is istrate istrate istrate istrate. If is istrate istrate istrate istrate istrate is istrate. If is istrate istrate istrate istrate istrate is istrate. If is istrate istrate istrate istrate istrate istrate. If is istrate istrate istrate istrate istrate istrate. If is istrate istrate istrate istrate istrate. If is istrate istrate istra</pre>
$\begin{array}{c} 417\\ 44221\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2222\\ 2$	004D 0050 0052 0055 0056 0054 0054 0055 0055	8E DA BA 03E5 EC BA 03E4 EC 3C FF 74 2E BA R 8E C2 BA 03E4 BO DC	<pre>image: provide a second s</pre>
$\begin{array}{c} 417\\ 4444444444444444444$	004D 0050 0052 0055 0056 0058 0058 0058 0056 0068	8E DA BA 03E5 EC BA 03E4 EC 3C FF 74 2E BA R 8E C2 BA 03E4 BO DC	<pre>image: provide the second second</pre>
$\begin{array}{c} 4  4  4  4  4  4  4  4  4  4 $	004D 0050 0052 0055 0056 0054 0054 0054 0056 0061 0063 0066 0068 0069	8E DA BA 03E5 EC BA 03E4 EC 3C FF 74 2E BA R 8E C2 BA 03E4 BO DC EE 33 C0 8B F8	<pre>image: provide the image: p</pre>
4 4 14 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	004D 0050 0052 0055 0056 0058 0058 0058 0061 0063 0066 0068 0068	8E DA BA 03E5 EC BA 03E4 EC 3C FF 74 2E BA R 8E C2 BA 03E4 B0 DC EE 33 C0 8B F8 B9 0080	<pre>image: provide the image: p</pre>
4 4 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	0040 0050 0052 0055 0056 0058 0058 0056 0061 0063 0066 0068 0068 0069 0068	8E DA BA 03E5 EC BA 03E4 EC 3C FF 74 2E BA R 8E C2 BA 03E4 B0 DC EE 33 C0 8B F8 B9 0080 FC	<pre>image: provide the image: p</pre>
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	004D 0050 0052 0055 0056 0058 0058 0058 0061 0063 0066 0068 0068 0068 0069	8E DA         BA 03E5         EC         BA 03E4         EC         3C FF         74 2E         BA R         8E C2         BA 03E4         BO 0C         EE         33 C0         8B F8         B9 0080         FC         F3/ AB	<pre>i* operational i* imput: None i* i* i* Dutput: i* CF = 0 if the 3278/3279 adapter is installed i 1 if the 3278/3279 adapter is not installed i* 1 if the 3278/3279 adapter is not operational i* F_EBA_NOT_OP = 1 if the 3278/3279 adapter is installed and i* EASIC_ASSURANCE_TEST PROC NEAR ASSUME DS:INITDATA HOV DX,INITDATA HOV DX,INITDATA HOV DX,START_OP_MOD : CLEAR THE START OPERATION MODIFIER IN AL,DX : REGISTER TO INSURE THAT THE ADAPTEF ; STATUS CANNOT BE FF MOV DX,STATUS_PORT ; READ THE ADAPTER STATUS - FF INDICATE IN AL,DX : THAT THE ADAPTER IS NOT INSTALLED CHP AL.OFFH JE BAT_2 ; TEST THE MEMORY ON THE 3278/3279 ADAPTER ASSUME ES:FBA_BUFFER ; ESTABLISH ADDRESSABILITY TO FEATURE NOV DX.FCHAND_PORT ; INDICATE THE ADAPTER IS NOT NOV AL,RESE_TIPL_COMPLETE ; INITIALIZED SO WE CAN ACCESS TH OUT DX.AL ; INITIALIZE THE FBA BUFFER TO ALL ZERC NOV DI.AX NOV DI.AX NOV CX.(SIZE_FBA_BUFFER)/2 CLD REP STOSW</pre>
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	004D 0050 0052 0055 005A 005A 005A 005A 0061 0063 0066 0068 0068 0069 0068 0069 0068	BE DA         BA 03E5         EC         BA 03E4         EC         SC FF         74 2E         BA R         BE C2         BA 03E4         BO DC         EE         33 C0         8B F8         B9 0080         FC         F3/ AB         B8 0DFF	<pre>if operational if input: None if input: None if output: if Output: if CF = 0 if the 3278/3279 adapter is installed if 1 if the 3278/3279 adapter is not installed if 1 if the 3278/3279 adapter is not installed if if the 3278/3279 adapter is not installed if if the 3278/3279 adapter is not installed if if the 3278/3279 adapter is not operational if the feature bus interface is not operational if the feature bus interface is not operational if if the isometation interface is not operational if if the isometation interface is not operational if if the feature bus interface is not operational if if the isometation isometation interface is not operational if if the isometation isometation interface is not insure that an interface is not insure that if the adapter is not installed if if the isometation is interface is not insure that the isometation is not installed if if the isometation is not insure that the adapter is not installed interface is not installed interfa</pre>
4 4 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	0040 0050 0052 0055 0054 0054 0054 0054	8E DA 8A 03E5 EC BA 03E4 EC 3C FF 74 2E BA 03E4 BA 03E4 BO 0C EE 33 C0 8B F8 B9 0080 FC F3/ AB B8 00FF E8 0098 R	<pre>if operational if input: None if input: None if output: if Output: if CF = 0 if the 3278/3279 adapter is installed if 1 if the 3278/3279 adapter is not installed if 1 if the 3278/3279 adapter is not installed if 1 if the 3278/3279 adapter is not installed if 1 if the 3278/3279 adapter is not installed if 1 if the 3278/3279 adapter is not operational if the feature bus interface is not operational if the feature bus interface is not operational if 1 if the 3278/3279 adapter is installed and if 1 if the feature bus interface is not operational if 1 if the 3278/3279 adapter is installed and if 1 if the 3278/3279 adapter is installed and if 1 if the 3278/3279 adapter is not operational if 1 if the 3278/3279 adapter is not operational if 1 if the dot dot dot dot dot dot dot dot dot dot</pre>
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	0040 0050 0052 0055 0054 0054 0054 0054	BE DA         BA 03E5         EC         BA 03E4         EC         SC FF         74 2E         BA R         BE C2         BA 03E4         BO DC         EE         33 C0         8B F8         B9 0080         FC         F3/ AB         B8 0DFF	<pre>if cperational if input: None if input: None if output: if if the 3278/3279 adapter is installed and if if the 3278/3279 adapter is installed if if the 3278/3279 adapter is installed if if the 3278/3279 adapter is installed and if if it is installed is installed if if it is installed and if if it is installed is installed is installed if it is installed and if if is installed is installed is installed is installed if it is installed is installed if it is installed if it is installed is installed if it is</pre>
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	0040 0050 0052 0055 0056 0054 0054 0056 0061 0063 0066 0068 0068 0068 0069 0068 0071 0071 0071	8E DA 8A 03E5 EC BA 03E4 EC 3C FF 74 2E BA 03E4 BA 03E4 BO 0C EE 33 C0 8B F8 B9 0080 FC F3/ AB B8 00FF E8 0098 R	<pre>if operational if input: None if input: None if output: if Output: if CF = 0 if the 3278/3279 adapter is installed if 1 if the 3278/3279 adapter is not installed if 1 if the 3278/3279 adapter is not installed if if the 3278/3279 adapter is not installed if if the 3278/3279 adapter is not installed if if the 3278/3279 adapter is not operational if the feature bus interface is not operational if the feature bus interface is not operational if if the isometation interface is not operational if if the isometation interface is not operational if if the feature bus interface is not operational if if the isometation isometation interface is not operational if if the isometation isometation interface is not insure that an interface is not insure that if the adapter is not installed if if the isometation is interface is not insure that the isometation is not installed if if the isometation is not insure that the adapter is not installed interface is not installed interfa</pre>

۰

		121		4,04	1,202	
		131				132
466						102
467	0083	33 C0		> OR	AX,AX	; TEST EACH BYTE FOR DO AND LEAVE THE
468	0085	E8 0098 R		CALL	TEST_RAM	HEMORY CLEARED TO ZEROS
469	0088	72 00		.'C	EAT_Ĩ	; MEMORY TEST FAILED
470						
471	ACOO	5.0	BAT_1:			
472 473	008A 008B			CLC		; INDICATE ADAPTER IS INSTALLED
474	0000	C3		RET		; RETURN TO MAINLINE ROUTINE
475	008C		BAT_2:			
476	0080	80 16 0082 R	DCC.	LEA	DX.MSG ADAPTER	NOT_OP ; INFORM THE USER THAT THE
477	0090	E8 022F R		CALL		ADAPTER IS NOT INSTALLED
478					-	
479	0093	E8 00B3 R		CALL	FEA_NOT_OP	; INDICATE THE FEATURE BUS ADAPTER
450						; IS NOT OPERATIONAL
461						
482 483	0096 0097	F9 C3		STC		INDICATE ADAPTER IS NOT INSTALLED
484	0041	5		RET		; FETURN TO MAINLINE FOUTINE
485	0098		TEST RA	M PROC N	EAR	
486	0098	33 FF		XOR	DI,01	START AT BEGINNING OF BUFFER
487	009A	89 0100		MOV	CX,SIZE_FBA_BUF	FER ; SCAN ALL BYTES IN THE BUFFER
488						
489	009D		TEST_RA			
490	009D	26: 3A 25		CMP		DI ; IS THIS BYTE CORRECT?
491 492	00A0 00A2	75 05 AA		JNE STOSB	TEST_RAM_2	; NO, MEMORY TEST FAILED ; YES, REPLACE IT WITH THE NEW VALUE
493		E2 F8		LOOP	TEST_RAM_1	; LCOP UNTIL ENTIRE EUFFER IS TESTED
494	0045			CLC		INDICATE MEMORY TEST SUCCEEDED
495	00A6	C3		RET		FETURN TO CALLER
496						
497	00A7		TEST_RA			
498 499	00A7			LEA		DP ; INFORM USER THAT THE FEATURE
500	UUAB	EB 022F R		CALL	INFORM_OPERATOR	; BUS ADAPTER IS NOT OPERATIONAL
501	ODAF	E8 00B3 R		CALL	FBA_NOT_OP	; INDICATE FEATURE BUS ADAPTER IS
502					102-401-01	; NOT OPERATIONAL
503						
504	0081	F9		STC		; INDICATE MEMORY TEST FAILED
505	0082	C3		RET		; RETURN TO CALLER
506	0083		TEST_RA	m endp		
507 508	0083				NEAD	
509		BA R	LDA_UOI	_OP PROC HOV	DX,BIOSDATA	I ESTABLISH ADDRESSABILITY TO FBA
510	0086	BE C2		MOV	ES,DX	STATE FLAGS
511	0088	8D 3E 0000 E		LEA	DI,BIOSOATA:FBA	
512					•	-
513	OOBC	26: 80 00 01		OR	BYTE PTR ES: IDI	),F_FBA_NOT_OP ; INDICATE THAT THE
514 515	0000	80 DE 0001 R 80		OR	STATUS, ST_FBA_NO	DT_OP ; FEATURE BUS ADAPTER IS NOT
516						; WORKING
517	0005	C3		RET		; RETURN TO CALLER
518	0006		FBA_NOT	OP- ENDP		
519						
520 521				ASSUME I	DS:NOTHING,ES:NOT	THING
522	0006		BASTC A	SSURANCE	TEST ENDP	
523						
524		c			NDSPINI2.ASM	
525		C		SUBTTL	Initialize 6845	Parameters
526						
527 528		C		******	*************************************	<b>我我我爹爹我们们努力才不不可不可能把你们把你们把你会给你我做你没做?你我要求我做了</b>
529		c c				
530		c	;# Prod	cedure Na	ame: INIT_6845	
531		C				1) <b>n</b>
532		C				display processor parameters so the
533 534		C C			ersonal Computer 278 and 3279 disp	screen image is correct on both the
535					Lie and Sciff disp	tey stations
536						
537						
538		C		278/3279	adapter status f	lag indicating whether the display
539		c			station is a 327	'8 or a 3279
540		C				
541		C		put:		
542		C				·
543 544						d to point to the parameter area d, which is initialized with the
545						s for this display station
546		č	; ;*	• •	-	
547		c	; ;******	*****	****	<b>这家长我就没能容然没有不能不能没有不能没有我不能没有这些我的不能不</b> 能不能
548		c	:			
549	00C6	C		45 PROC	NEAR	
550		C				
551					DS:INITDATA,ES:R	UNUALA
552 553	0006			MOV	DX,INITDATA	; PICK UP ADDRESS OF WORK AREA SEGMENT
553 554	0009	BE DA C		MOV	DS.DX	
555	00CB	BA R C		MOV	DX,RCMDATA	; PICK UP ADDRESS OF ROM WORK AREA
556						; SEGMENT
557	OOCE	BE C2 C		MOV	ES,DX	
558		c	:			

•

				4,641	,262	
		133				134
					DV CTATUS DODT	; READ ADAPTER STATUS
559		BA 03E4	c c	MOV IN	AL,DX	
560 561	0003	EC	c	10		
562	0004	8D 36 0002 R	č	LEA	SI, PARMS_3278	; POINT TO PROPER SET OF 6845
563			č			; PARAMETERS
564	0008	A8 20	С	TEST	AL, ST_3278	
565		75 04	C	JNZ	MOV_6845	
566	0 ODC	8D 36 0042 R		LEA	SI, PARMS_3279	
567 568	0020		ç	MOV_6845:		
569	00E0	26: 80 3E 0090 R	č	LEA	DI, PARMS_6845	; MOVE PARAMETERS TO BIOS DATA AREA
570		B9 0020	c	MOV	CX,32	
571	00E8		C	CLD		
572	00E9	F3/ A5	c	REP	MOVSW	
573	0050	AC C2	c c	MOV	DX,ES	; FOINT INTERRUPT VECTOR 1D TO THE
574 575	OOED	8C C2 8E DA	č	MOV		NEW VIDEO PARAMETERS
576		26: 80 16 0090 R	č	LEA	DX, PARMS_6845	
577		60 1D	С	MOV	AL, IDH	
578	00F6	B4 25	С	MOV	AH,25H	
579	00F8	CD 21	ç	INT	21H	
580 581	0054	B4 OF	с с	HOV	AH,15	I GET CURRENT CRTC MODE
582		CD 10	č	INT	108	
583	ODFE	32 E4	C	XOR	AH, AH	; REINITIALIZE THE 6845 WITH THE NEW
584	0100	CD 10	С	INT	1 O H	; PARAMETERS WITHOUT CHANGING MODES
585			C			RIGHT ; DISPLAY THE COPYRIGHT MESSAGE
586		BB E	с с	MOV MOV	DS.AX	
587 588	0105 0107	8E D8 BA 0000 E	č	HOV	DX, DFFSET HDSP_C	OPYRIGHT
589		EB 022F R	č	CALL	INFORM_OPERATOR	
590			C			
591	010D	C3	C	RET		; RETURN TO MAINLINE ROUTINE
592			C	ASSUME	DS:NOTHING,ES:N	1171/2
593	0105		C C	INIT_6845 ENDP		
594 595	010E		č	_		
596		•	Ċ	SUBTTL	Initialize 3278.	/3279 Attachment Option
597			С			
598			С			
599			С		******	· 卡努市产效率扩展中水学型素能激素的实验的生物的关系,因为其水学及全国学家的常常。
600			c	;# .v. December N		
601 602			C C	;* Procedure N	ame: INIT_ADAPTER	s de la constante de
603			č	;* Function: T	o initialize the	3276/3279 Attachment Option
604			Ċ	; *		
			-			
605			С	<pre>;* Input: None</pre>		
606			C	5 <b>%</b>		
606 607			C C	;* ;* Dutput:		
606 607 608			с с с	;* }* Output: ;*		d identifier
606 607			C C	;* ;* Output: ;* ;* KB_ID = 3	278/3279 keyboard adapter initiali	
606 607 608 609				;* ;* Output: ;* ;* KB_ID = 3 ;* 3278/3279	278/3279 keyboard adapter initial	ized
606 607 608 609 610 611 612				;* ;* Output: ;* ;* KB_ID = 3 ;* 3278/3279	278/3279 keyboard adapter initial	
606 607 608 609 610 611 612 613	<b>A</b> 107			;* ;* Dutput: ;* KB_ID = 3 ;* 3278/3279 ;* ;******	278/3279 keyboard adapter initial *******************	zed
606 607 608 609 610 611 612 613 614	010E			;* ;* Output: ;* ;* KB_ID = 3 ;* 3278/3279	278/3279 keyboard adapter initial *******************	zed
606 607 608 609 610 611 612 613 614 615	DIGE			;* ;* ;* ;* ;* KB_ID = 3 ;* 3278/3279 ;* ;*****************	278/3279 keyboard adapter initial ************************************	**************************************
606 607 608 609 610 611 612 613 614		BA R		;* ;* ;* ;* ;* KB_ID = 3 ;* 3278/3279 ;* ;*****************	278/3279 keyboard adapter initial ************************************	zed
606 607 608 609 610 611 612 613 614 615 614 615 616 617 618	010E	ΒΑ R 8ε da		;* ;* ;* ;* KB_ID = 3 ;* 3278/3279 ;* ;***************** INIT_ADAPTER PR ASSUME	278/3279 keyboard adapter initial ************************************	**************************************
606 607 608 609 610 611 612 613 614 615 616 616 617 618 619	010E 0111	8E DA		;* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;	278/3279 keyboard adapter initial ************************************	; PICK UP ADDRESS OF DATA SEGMENT
606 607 608 609 610 611 612 613 614 615 616 616 616 617 618 619 620	010E 0111 0113	8E DA Ba 03E4		;* ;* Dutput: ;* KB_ID = 3 ;* 3278/3279 ;* ;*********************************	278/3279 keyboard adapter initial ************************************	; PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER
606 607 608 609 610 611 612 613 614 615 614 615 616 617 618 619 620 621	010E 0111 0113 0116	8E DA Ba 03E4 B0 0E		;* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;* ;	278/3279 keyboard adapter initiali www.www.www. OC NEAR DS:INITDATA DX,INITDATA DS.DX DS.DX DX,COMMAND_PORT AL.RESET_ADAPTEI	; PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER
606 607 608 609 610 611 612 613 614 615 616 616 616 617 618 619 620	010E 0111 0113	8E DA Ba 03E4 B0 0E		;* ;* Dutput: ;* KB_ID = 3 ;* 3278/3279 ;* ;*********************************	278/3279 keyboard adapter initiali www.www.www. CC NEAR DS:INITDATA DS,DX DS,DX DS,DX DX,COMMAND_PORT AL,RESET_ADAPTEI DX,AL	; PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER
606 607 608 609 610 611 612 613 614 615 614 615 616 616 617 618 619 621 622 623 624	010E 0111 0113 0116 0118 0119	8E DA BA 03E4 BO 0E EE BO 0A		;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initiali www.www.www.www. OC NEAR DS:INITDATA DS,DX DS,DX DX,COMMAND_PORT AL,RESET_ADAPTEI DX,AL AL,DISABLE_NDS	; PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY
606 607 608 609 610 612 612 613 614 615 616 615 616 617 620 621 622 623 624 625	010E 0111 0113 0116 0118	8E DA BA 03E4 BO 0E EE BO 0A		;* ;* Output: ;* KB_ID = 3 ;* 3276/3279 ;* ;********************************	278/3279 keyboard adapter initiali www.www.www. OC NEAR DS:INITDATA DS,DX DS,DX DS,DX DX,COMMAND_PORT AL,RESET_ADAPTEI DX,AL	; PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER
606 607 608 609 610 611 612 613 614 615 616 616 616 617 618 619 620 621 622 623 624 625 626	010E 0111 0113 0116 0118 0119 0118	8E DA BA 03E4 BO 0E EE BO 0A EE		;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV OUT HOV OUT	278/3279 keyboard adapter initial ************************************	; PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE
606 607 608 609 610 611 612 613 614 615 616 616 617 618 616 620 621 622 623 624 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 627 625 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 7 7 7 7 7 7 7	010E 0111 0113 0116 0118 0119 0118 0119	8E DA BA 03E4 BO 0E EE BO 0A EE BA 03E3		;* ;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV OUT MDV OUT MDV	278/3279 keyboard adapter initiali www.www.www.www. CO NEAR DS:INITDATA DS,INITDATA DS,DX DX,COMMAND_PORT AL,RESET_ADAPTEI DX,AL AL,DISABLE_NDS DX,AL DX,CMD_8255	<pre>ized ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT</pre>
606 607 609 610 612 612 613 614 615 616 615 616 617 620 621 622 623 622 623 625 626 625 626 628	010E 0111 0113 0116 0118 0119 0118 0117 0118	8E DA BA 03E4 BO 0E EE BO 0A EE BA 03E3 BO 94		;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV OUT HOV OUT	278/3279 keyboard adapter initial ************************************	<pre>ized ; PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT B = MODE 1 OUTPUT</pre>
606 607 608 609 610 611 612 613 614 615 616 616 617 618 616 620 621 622 623 624 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 625 627 625 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 627 7 7 7 7 7 7 7	010E 0111 0113 0116 0118 0119 0118 0119	8E DA BA 03E4 BO 0E EE BO 0A EE BA 03E3 BO 94		;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV HOV MOV OUT MOV OUT MOV MOV MOV	278/3279 keyboard adapter initial www.www.www.www. DS:INITDATA DS.INITDATA DS.DX DX.COMMAND_PORT AL,RESET_ADAPTEI DX.AL AL,DISABLE_NDS DX.AL DX.CKD_8255 AL.94H	<pre>ized ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT</pre>
606 607 609 610 611 612 613 614 616 617 618 619 620 621 622 622 623 624 625 624 626 627 628 622 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 626 627 628 629	010E 0111 0113 0116 0118 0119 0118 0117 0117 0121	8E DA BA 03E4 BO 0E EE BO 0A EE BA 03E3 BO 94 EE		;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV HOV MOV OUT MOV OUT MOV OUT	278/3279 keyboard adapter initial WWWWWWWWWWWWWWWWWWWWWWWW DS:INITDATA DS:INITDATA DS.DX DX.COMMAND_PORT AL,RESET_ADAPTED DX,AL AL,DISABLE_NDS DX,AL DX,CMD_8255 AL,94H DX,AL	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT B = MODE 1 OUTPUT ; PORT C = MODE 0 OUTPUT</pre>
606 607 609 610 611 612 613 614 615 616 617 616 617 618 620 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 623 622 623 622 623 625 626 627 630 632 632	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122	8E DA BA 03E4 BO 0E EE BO 0A EE BA 03E3 BO 94 EE BA 03E2		;* ;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV OUT MOV OUT MOV OUT MOV OUT	278/3279 keyboard adapter initiali ***********************************	<pre>ized ; PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT B = MODE 1 OUTPUT</pre>
606 607 609 610 611 612 613 614 616 617 618 621 6223 6224 6224 6224 6224 6226 6224 6226 6224 6226 6224 6226 630 6312 633 633	010E 0111 0113 0116 0118 0119 0118 0117 0121 0121 0122 0125	8E         DA           BA         03E4           BO         0E           EE         0           BA         03E3           BO         94           EE         03E2           BA         03E2		;* i* Output: i* i* KB_ID = 3 i* 3278/3279 i* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV OUT MOV OUT MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initiali www.www.www.www. CO NEAR DS:INITDATA DX,INITDATA DS,DX DX,COMMAND_PORT AL,RESET_ADAPTEN DX,AL AL,DISABLE_NDS DX,AL DX,CMD_8255 AL,94H DX,AL DX,KB_CTL AL,SELECT_ID	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT B = MODE 1 OUTPUT ; PORT C = MODE 0 OUTPUT</pre>
606 607 609 610 611 612 613 614 615 616 617 616 617 620 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 623 632 632 633 633 634 633 634 633 634 634 634 635 634 635 634 635 634 632 633 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 7 7 7 7 7 7 7 7	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122	8E         DA           BA         03E4           BO         0E           EE         0           BA         03E3           BO         94           EE         03E2           BA         03E2		;* ;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV OUT MOV OUT MOV OUT MOV OUT	278/3279 keyboard adapter initiali ***********************************	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT A = MODE 1 CUTPUT ; PORT B = MODE 1 CUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF READ .</pre>
606 607 609 610 611 612 613 614 615 616 617 616 617 616 617 620 622 622 622 622 622 622 622 625 627 629 630 632 632 633 633 633 635	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127	8E         DA           BA         03E4           BO         0E           EE         0           BA         03E3           BO         94           EE         03E2           BA         03E2		;* i* Output: i* i* KB_ID = 3 i* 3278/3279 i* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV OUT MOV OUT MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initiali ***********************************	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT B = MODE 1 OUTPUT ; PORT C = MODE 0 OUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF RFAD</pre>
606 607 609 610 611 612 613 614 615 616 617 616 617 620 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 623 632 632 633 633 634 633 634 633 634 634 634 635 634 635 634 635 634 632 633 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 634 7 7 7 7 7 7 7 7	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0122 0122 0122 0127 0128	8E         DA           BA         03E4           BO         OE           EE         0           BA         03E3           BO         94           EE         0           BA         03E2           BO         04		;* i* Output: i* KB_ID = 3 ;* 3278/3279 ;* INIT_ADAPTER PR ASSUME MOV HOV HOV OUT HOV OUT HOV OUT HOV OUT HOV OUT HOV OUT HOV OUT HOV OUT HOV OUT	278/3279 keyboard adapter initiali WHENER DS:INITDATA DS.JNITDATA DS.JNITDATA DS.JX DX.COHMAND_PORT AL.RESET_ADAPTED DX.AL DX.CHD_8255 AL.94H DX.AL DX.KB_CTL AL.SELECT_ID DX.AL DX.NDS_KB_IN AL.DX	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT A = MODE 1 CUTPUT ; PORT B = MODE 1 CUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF READ .</pre>
606 607 609 610 612 613 614 615 616 617 616 617 619 620 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 623 625 625 6331 6333 6335 635 636 635 635 636 738 635 635 635 635 636 738 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 635 636 738 635 636 637 638 635 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 638 7 638 7 638 7 7 7 7 7 7 7 7	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127 0128 0128 0126	8E         DA           BA         03E4           BO         0E           EE         0           BA         03E3           BO         94           EE         0           BA         03E2           BA         03E2           BA         03E2           BA         03E2           BA         03E0           EC         24		;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV MOV MOV OUT MOV OUT MOV OUT MOV MOV OUT MOV MOV OUT MOV MOV MOV OUT MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initiali ***********************************	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT A = MODE 1 CUTPUT ; PORT B = MODE 1 CUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF READ .</pre>
606 607 609 610 611 612 613 614 616 617 616 617 616 620 622 622 622 622 622 623 622 622 623 622 623 622 623 622 623 622 623 622 623 622 623 622 623 623 625 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 623 6332 6332 6335 6335 6336 6337 636 536 537 636 537 638 637 638 639 637 638 637 638 637 638 637 638 637 638 637 638 637 638 637 638 639 637 638 639 637 638 637 639 637 636 537 636 536 536 537 638 639 637 638 539 639 637 638 539 639 639 639 639 639 639 639 639 639 639 639 639 639 639 639 639 639 639 7 7 7 7 7 7 7 7	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127 0128 0128 0126	8E         DA           BA         03E4           BO         OE           EE         0           BA         03E3           BO         94           EE         0           BA         03E2           BO         04		;* i* Output: i* KB_ID = 3 ;* 3278/3279 ;* INIT_ADAPTER PR ASSUME MOV HOV HOV OUT HOV OUT HOV OUT HOV OUT HOV OUT HOV OUT HOV OUT HOV OUT HOV OUT	278/3279 keyboard adapter initiali WHENER DS:INITDATA DS.JNITDATA DS.JNITDATA DS.JX DX.COHMAND_PORT AL.RESET_ADAPTED DX.AL DX.CHD_8255 AL.94H DX.AL DX.KB_CTL AL.SELECT_ID DX.AL DX.NDS_KB_IN AL.DX	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT A = MODE 1 CUTPUT ; PORT B = MODE 1 CUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF READ .</pre>
606 607 609 610 611 612 613 614 615 616 617 616 617 621 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 623 632 632 632 633 632 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 7 7 7 7 7 7 7 7	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127 0128 0126 0126 0126	BE         DA           BA         03E4           BO         OE           EE         0           BA         03E3           BO         94           EE         94           BA         03E2           BO         0.3E2           BO         0.3E2           EC         24           OF         A2		;* i* Output: i* KB_ID = 3 ;* 3278/3279 ;* i*********************************	278/3279 keyboard adapter initiali WHENER DS:INITDATA DS.JNITDATA DS.JNITDATA DS.JX DX.COMMAND_PORT AL.RESET_ADAPTEL DX.AL AL.DISABLE_NDS DX.AL DX.KB_CTL AL.SELECT_ID DX.AL DX.KB_CTL AL.SELECT_ID DX.AL DX.NDS_KB_IN AL.OX AL.OFN KB_ID.AL	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT A = MODE 1 CUTPUT ; PORT B = MODE 1 CUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF READ .</pre>
606 607 609 610 612 613 614 615 616 617 615 616 622 6223 6223 6223 6222 6223 6225 6226 6229 6331 6334 5334 5354 6359 6334 6334 6335 6339 6334 6339 6334 6339 6334 6339 6334 6339 6334 6334 6334 6336 6339 6334 6336 6339 641	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127 0127 0127 0128 0128 0122 0122	8E       DA         BA       03E4         BO       0E         EE       0         BA       03E3         BO       94         EE       0         BA       03E2         BA       03E2         BA       03E0         EC       24         DA       03E3		;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV MOV MOV OUT MOV OUT MOV OUT MOV MOV OUT MOV MOV OUT MOV MOV MOV OUT MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initiali ***********************************	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT B = MODE 1 OUTPUT ; PORT C = MODE 0 OUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF RFAD ; READ KEYBOARD IDENTIFIER ; INITIALIZE THE 8255: ; PORT A = MODE 1 INPUT</pre>
606 607 609 610 611 612 613 614 615 616 617 616 617 621 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 622 623 632 632 632 633 632 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 633 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 630 7 7 7 7 7 7 7 7	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127 0127 0127 0128 0128 0122 0122	8E         DA           BA         03E4           BO         0E           EE         0           BA         03E3           BO         94           EE         0           BA         03E3           BO         94           EE         0           BA         03E2           BO         40           EE         0           BA         03E0           EC         24           BA         03E0           EA         0000 R           BA         03E3           BO         B4		;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV MOV MOV OUT MOV OUT MOV OUT MOV OUT MOV MOV OUT MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initial WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT C = MODE 0 OUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF READ ; READ KEYBOARD IDENTIFIER ; INITIALIZE THE 8255: ; PORT A = MODE 1 INPUT ; PORT B = MODE 1 OUTPUT ; PORT B = MODE 1</pre>
606 607 609 610 611 612 613 614 616 617 616 617 616 620 622 6223 6222 6223 6222 6223 6222 6224 620 6332 6334 6335 6335 6337 6339 6401 642	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127 0128 0126 0126 0122 0128	8E         DA           BA         03E4           BO         0E           EE         0           BA         03E3           BO         94           EE         0           BA         03E3           BO         94           EE         0           BA         03E2           BO         40           EE         0           BA         03E0           EC         24           BA         03E0           EA         0000 R           BA         03E3           BO         B4		;* i* Dutput: i* KB_ID = 3 i* 3278/3279 i* 3278/3279 i* 3278/3279 i* AND MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initiali ***********************************	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255:     PORT A = MODE 0 INPUT ; PORT B = MODE 1 OUTPUT ; PORT C = MODE 0 OUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF READ ; READ KEYBOARD IDENTIFIER ; INITIALIZE THE 8255: ; PORT A = MODE 1 INPUT </pre>
606 607 609 610 6112 613 6145 6145 616 617 616 617 619 622 6223 6222 6223 6222 6223 6222 6223 6222 6225 6229 6332 6335 63378 6339 6442 6442 645	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127 0128 0127 0128 0127 0128 0127 0128 0127 0128 0127	8E         DA           BA         03E4           BO         0E           EE         0           BA         03E3           BO         94           EE         0           BA         03E3           BO         94           EE         0           BA         03E2           BO         40           EE         0           BA         03E0           EC         24           BA         03E0           EA         0000 R           BA         03E3           BO         B4		;* i* Output: i* i* KB_ID = 3 i* 3276/3279 i* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV OUT MOV OUT MOV HOV OUT MOV MOV OUT MOV MOV OUT MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initiali ***********************************	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT C = MODE 0 OUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF READ ; READ KEYBOARD IDENTIFIER ; INITIALIZE THE 8255: ; PORT A = MODE 1 INPUT ; PORT B = MODE 1 OUTPUT ; PORT B = MODE 1</pre>
606 607 609 610 612 613 615 616 615 616 615 622 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6224 6225 6227 62334 63334 6337 6339 64412 6443 6444 6445 6446	010E 0111 0113 0116 0118 0117 0121 0122 0125 0127 0128 0126 0127 0128 0122 0128 0127 0128 0126 0127	8E       DA         BA       03E4         BO       OE         EE       0         BA       03E3         BO       94         EE       94         BA       03E2         BO       40         EE       40         BA       03E2         BA       03E2         BA       03E0         EC       24         PA       03E0         BA       03E3         BO       B4         EE       94		;* i* Output: i* KB_ID = 3 i* 3276/3279 i* i*INIT_ADAPTER PR ASSUME MOV MOV MOV OUT MOV OUT MOV OUT MOV OUT MOV OUT MOV MOV OUT MOV MOV OUT MOV MOV MOV OUT MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initiali WHENHERMENT OC NEAR DS:INITDATA DS,DX DX,COMMAND_PORT AL,RESET_ADAPTEN DX,AL AL,DISABLE_NDS DX,AL DX,CMD_6255 AL,94H DX,AL DX,KB_CTL AL,SELECT_ID DX.AL DX,KB_CTL AL,OS_KB_IN AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF A	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT C = MODE 0 OUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF RFAD ; INITIALIZE THE 8255: ; PORT A = MODE 1 INPUT ; PORT A = MODE 1 INPUT ; PORT A = MODE 1 OUTPUT ; PORT C = MODE 1 OUTPUT</pre>
606 607 609 6112 612 613 6145 6156 6156 61676 612234 622234 622234 63334 633345 633345 633345 6444234 644456 645647	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127 0128 0126 0126 0126 0126 0126 0126 0126 0126	8E       DA         BA       03E4         BO       OE         EE       0         BA       03E3         BO       94         EE       94         BA       03E2         BO       40         EE       0         BA       03E0         EC       24         OF       A2         BA       03E3         BO       B4         EE       94		<pre>i* Output: i* Output: i* KB_ID = 3 j* 3278/3279 j* i**********************************</pre>	278/3279 keyboard adapter initial WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW	<pre>ized i PICK UP ADDRESS OF DATA SEGMENT ; RESET THE 3278/3279 ADAPTER ; DISABLE THE 3278/3279 DISPLAY ; INTERFACE ; INITIALIZE THE 8255: ; PORT A = MODE 0 INPUT ; PORT C = MODE 0 OUTPUT ; SELECT KEYBOARD IDENTIFIER TO BF READ ; READ KEYBOARD IDENTIFIER ; INITIALIZE THE 8255: ; PORT A = MODE 1 INPUT ; PORT B = MODE 1 OUTPUT ; PORT B = MODE 1</pre>
606 607 609 611 612 612 613 6145 6145 6145 616 6222 6222 6222 6222 6222 6222 6222 63333 63356 63378 6445 6445 6445 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6467 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6478 6	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127 0128 0126 0127 0128 0126 0127 0128 0126 0127 0134	8E       DA         BA       03E4         BO       0E         EE       0A         BA       03E3         BO       94         EE       0A         BA       03E3         BO       94         EE       0A         BA       03E2         BA       03E0         EC       0A         BA       03E0         EE       BA         BA       03E0         EE       BA         BA       03E3         BO       B4         BA       03E3         BO       B4		;* i* Output: i* KB_ID = 3 i* 3276/3279 i* i*INIT_ADAPTER PR ASSUME MOV MOV MOV OUT MOV OUT MOV OUT MOV OUT MOV OUT MOV MOV OUT MOV MOV OUT MOV MOV MOV OUT MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initiali WHENHER OC NEAR DS:INITDATA DS,DX DX,COMMAND_PORT AL,RESET_ADAPTEN DX,AL AL,DISABLE_NDS DX,AL DX,CMD_6255 AL,94H DX,AL DX,KB_CTL AL,SELECT_ID DX.AL DX,KB_CTL AL,OS_KB_IN AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF AL,OF	<pre>ized ized i PICK UP ADDRESS OF DATA SEGMENT i RESET THE 3278/3279 ADAPTER i DISABLE THE 3278/3279 DISPLAY i INTERFACE i INITIALIZE THE 8255:     PORT A = MODE 0 OUTPUT i SELECT KEYBOARD IDENTIFIER TO BF RFAD i READ KEYBOARD IDENTIFIER i INITIALIZE THE 8255:     PORT A = MODE 1 OUTPUT i PORT C = MODE 1 OUTPUT i FORT A = MODE 1 OUTPUT i FORT A = MODE 1 OUTPUT i FORT C = MODE 1 OUTPUT i FORT</pre>
606 607 609 6112 612 613 6145 6156 6156 61676 612234 622234 622234 63334 633345 633345 633345 6444234 644456 645647	010E 0111 0113 0116 0118 0119 0118 0117 0121 0122 0125 0127 0128 0126 0127 0128 0126 0127 0128 0126 0127 0134	8E       DA         BA       03E4         BO       OE         EE       0         BA       03E3         BO       94         EE       0         BA       03E2         BO       40         EE       0         BA       03E2         BA       03E0         EC       03E0         EC       BA         BA       03E0         EC       BA         BA       03E0         EC       BA         BA       03E0         EC       BA		;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initial WHENNER DS:INITDATA DS,INITDATA DS,DX DX,COMMAND_PORT AL,RESET_ADAPTEN DX,AL AL,DISABLE_NDS DX,AL DX,CHD_8255 AL,94H DX,AL DX,KB_CTL AL,SELECT_ID DX.AL DX.NDS_KB_IN AL,DX AL,OFH KB_ID,AL DX,CHD_8255 AL,054H DX,AL DX,NDS_KB_IN AL,DX AL,054H DX,AL	<pre>ized ized i PICK UP ADDRESS OF DATA SEGMENT i RESET THE 3278/3279 ADAPTER i DISABLE THE 3278/3279 DISPLAY i INTERFACE i INITIALIZE THE 8255:     PORT A = MODE 0 OUTPUT i SELECT KEYBOARD IDENTIFIER TO BF RFAD i READ KEYBOARD IDENTIFIER i INITIALIZE THE 8255:     PORT A = MODE 1 OUTPUT i PORT C = MODE 1 OUTPUT i FORT A = MODE 1 OUTPUT i FORT A = MODE 1 OUTPUT i FORT C = MODE 1 OUTPUT i FORT</pre>
606 607 609 610 612 613 615 615 616 615 6223 6223 6223 6223 6223 6223 6223 6223 63333 5334 5334 5334 5444 5446 6449 6449	010E 0111 0113 0116 0118 0119 0118 0117 0122 0125 0127 0128 0126 0126 0126 0126 0126 0126 0126 0127 0134 0136 0137 0134 0136 0136	8E       DA         BA       03E4         BO       OE         EE       0         BA       03E3         BO       94         EE       0         BA       03E2         BO       40         EE       0         BA       03E2         BA       03E0         EC       03E0         EC       BA         BA       03E0         EC       BA         BA       03E0         EC       BA         BA       03E0         EC       BA		;* ;* ;* ;* ;* ;* ;* INIT_ADAPTER PR ASSUME MOV MOV MOV MOV MOV MOV MOV MOV	278/3279 keyboard adapter initial WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW	<pre>ized ized i PICK UP ADDRESS OF DATA SEGMENT i RESET THE 3278/3279 ADAPTER i DISABLE THE 3278/3279 DISPLAY i INTERFACE i INITIALIZE THE 8255:     PORT A = MODE 0 OUTPUT i SELECT KEYBOARD IDENTIFIER TO BF RFAD i READ KEYBOARD IDENTIFIER i INITIALIZE THE 8255:     PORT A = MODE 1 OUTPUT i PORT C = MODE 1 OUTPUT i FORT A = MODE 1 OUTPUT i FORT A = MODE 1 OUTPUT i FORT C = MODE 1 OUTPUT i FORT</pre>
606 607 609 6112 6123 6145 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 6156 61578 6156 6156 61578 6156 6156 61578 6156 61578 6156 61578 6156 61578 6156 61578 6156 61578 6156 6222345 6222345 625789 03123345 65333789 04123345 65664445 644789 656664789 0123445 6566644789 6566664789 0123445 6566644789 6566664789 6566666666666666666666666666666666666	010E 0111 0113 0116 0118 0117 0121 0122 0125 0125 0125 0125 0125 0125	8E       DA         BA       03E4         BO       OE         EE       0         BA       03E3         BO       94         EE       94         BA       03E2         BO       0         BA       03E2         BA       03E0         EC       24         OF       A2         BA       03E3         BO       B4         EE       94		<pre>;*</pre>	278/3279 keyboard adapter initial WHENNER DS:INITDATA DS,INITDATA DS,DX DX,COMMAND_PORT AL,RESET_ADAPTEN DX,AL AL,DISABLE_NDS DX,AL DX,CHD_8255 AL,94H DX,AL DX,KB_CTL AL,SELECT_ID DX.AL DX.NDS_KB_IN AL,DX AL,OFH KB_ID,AL DX,CHD_8255 AL,054H DX,AL DX,NDS_KB_IN AL,DX AL,054H DX,AL	<pre>ized ized i PICK UP ADDRESS OF DATA SEGMENT i RESET THE 3278/3279 ADAPTER i DISABLE THE 3278/3279 DISPLAY i INTERFACE i INITIALIZE THE 8255:     PORT A = MODE 0 OUTPUT i SELECT KEYBOARD IDENTIFIER TO BF RFAD i READ KEYBOARD IDENTIFIER i INITIALIZE THE 8255:     PORT A = MODE 1 OUTPUT i PORT C = MODE 1 OUTPUT i FORT A = MODE 1 OUTPUT i FORT A = MODE 1 OUTPUT i FORT C = MODE 1 OUTPUT i FORT</pre>

.

135

## 136

ARACTER AVAILABL
ſ
YBOARD CLICKER
INTERRUPT FLAG
IER
DISPLAY INTERFAC
O FC NCDE
UTINE
ttachment Option
***********
•
hy the
by the * *
•
*
nterrupt +
*
*****
TERRUPT MASK
ERRUPT LEVEL 2
DUTINE
nslation tables
on table for tine
ion tables
***
DATA SEGMENT
ENTIFIER   ATTACHED? DARD
THAT THERE ITACHED

-

### 138

~

					4,641	,262	
		137			•		138
747	0181	BO OF	ç	м	IOV	AL,15	; USE THE DEFAULT KEYBOARD
748 749	0183		ç				
750	A102		C C	INIT_XLAT	SSUME	DS:TABLES,ES:BI	ATARCO
751	0183	BA R	č			DX, TABLES	; PICK UP ADDRESS OF TRANSLATION TABLES
752	0166	BE DA	č		101	DS,DX	SEGMENT
753			С				
754	0188	BA R	C		IOV	DX, BIOSDATA	I PICK UP ADDRESS OF BIDS DATA SEGMENT
755	0166	8E C2	C	۲	107	ES,DX	
756 757	018D	DO EO	C C		AL	AL,1	; AX = 8*KEYBOARD IDENTIFIER
758	018F	DO EO	č		SAL	AL,1	, HA = 0.112,001.00 -00-00
759	0191	DO ED	č		AL	AL,1	
760	0193	32 E4	С	>	OR	LH,AH	
761			C				TABLE TOENTIFIER
762 763	0195 0197	88 FO 88 84 0000 E	C C		10V	SI,AX	; PICK UP TRANSLATION TABLE IDENTIFIER
764	019B	0B C0	c		10V )R	AX,XLAT_TABLES_ AX,AX	; IS THERE A TRANSLATION TABLE FOR THIS
765			č	-			; KEYBOARD TYPE?
766	01 7D	75 15	C	_	MZ	INIT_XLAT_2	; YES, USE THIS KEYBOARD TYPE
767			C				THE WEY DO THE THE VEYDOARD
768	019F	16	C		PUSH	DS	; INFORM THE USER THAT THIS KEYBOARD ; IS NOT SUPPORTED
769 770	01A0 01A3	BA R 8E DA	C C		10V 10V	DX, INITDATA DS, DX	S IS NOT SUPPORTED
771	VINJ	OL BA	č		SSUME	DS:INITDATA	
772	01A5	8D 16 00E3 R	č		EA	DX, HSG_INVALID_	KEYBDARD
773	01A9	E8 022F R	C	C	ALL	INFORM_OPERATOR	
774	DAID	1F	c		POP	DS	
775 776			C C	1	SSUME	DS:TABLES	
777	DALO	BE 0078	C C	۲	10V	SI,15×8	; USE THE DEFAULT KEYBOARD TRANSLATION
778	01B0	88 84 0000 E	č		10V		FTRISII ; TABLE
779		-	C				-
780	01B4		C	INIT_XLA			
761	0184	26: A3 D000 E	C	-	10V	HORD PIR XLAI_I	ABLES,AX ; SAVE KEYBOARD IDENTIFIER
782 783	C188	56 .	C C		าบรห	51	; SAVE TRANSLATION TABLE INDEX ADDRESS
784	(100	50	č	•	00		
785	0167	E5 54 0002 E	Ē	r	101		PTR+21SI) ; MOVE THE TRANSLATION TABLE
786	01E0	26: 80 3E 0002 E	C		.EA		2 ; INTO THE BIDS DATA AREA
787	01C2	25: ED DE DODD E	C		EA	CX,XLAT_TABLES_	END
789	01C7		C		UB	CX,DI	
789		D1 F9	C C		10 10	CX,1	
790 791	01CB 01CC	FC F3/ A5	č		EP	MOVSW	
792	0100		č				
793	01CE	5E	С	7	907	SI	A RETRIEVE TRANSLATION TABLE INDEX
794	01CF	56	C		USH	51	THE ADDRESS OF FIRST
795	01D0	88 B4 0004 E	c	m	IOV	SI .XLAI_IABLES_	PTR+4[SI] ; PICK UP ADDRESS OF FIRST HODIFICATION TABLE
796 797	0104	0B F6	C C	0	R	SI,SI	IS THERE & MODIFICATION TABLE?
798	0106	74 03	č		IZ	INIT_XLAT_3	NO, CONTINUE
799			С				
800	01D8	E8 0203 R	C	c	ALL	INIT_XLAT_MOD	; YES, APPLY THE MODIFICATIONS
801			ç				
802 803	01DB 01DB	5E	C C	INIT_XLAT	-2: 909	SI	RETRIEVE TRANSLATION TABLE INDEX
803	01DC	8B B4 0006 E	č		107	SI,XLAT TABLES_	PTR+6151) ; PICK UP ADDRESS OF SECOND
805		··	č				MODIFICATION TABLE
806		0B F6	C		R	SI,SI	IS THERE A MODIFICATION TABLE?
807	01E2	74 03	c	4	IZ	INIT_XLAT_4	I NO, CONTINUE
808 809	0154	EB 0203 R	C C	ſ	ALL	INIT XLAT MOD	1 YES, APPLY THE MODIFICATIONS
810	0164	10 VEV8 N	č	-			
811	01E7		С				AND WE HAVE DES TABLE LODITES TO THE
812	01E7	80 36 0000 E	C	- ı	.EA	SI, RPQ_TABLE	SEE IF THE RPQ TABLE APPLIES TO THIS
813		26: 88 1E 0000 E	c		10V		T_TABLES ; KEYBDARD
814 815		3B 1C 75 0E	C C		:MP INE	BX, WORD PTR [SI INIT_XLAT_5	I NO, SKIP THE REQ TABLE
815	0165		c	•			
817	01F4	8B 5C 02	C		10V	BX, WORD PTR [SI	+21; YES, CHANGE THE TRANSLATION TABLE
618		26: 89 1E 0000 E	C	+	10V	NORD PTR XLAT_T	ABLES, BX ; IDENTIFIER TO THE REQ TABLE
819			C				; IDENTIFIER
820 821	0150	83 C6 04	с с		DD	51,4	; POINT TO START OF RPQ MODIFICATIONS
821 822		83 C6 04 E8 0203 R	C C		CALL	INIT_XLAT_MOD	
823			č	. `			
824	0202		С	INIT_XLAT			PETITIN TO MATHE DOUTTINE
825	0202	C3	С	ŗ	739		; RETURN TO MAINLINE ROUTINE
826			C C		CCIME	DS:NOTHING,ES:N	DTHING
827 828			с С		COOUNE	53.NUINING)13.N	
829	0203		c	INIT_XLAT	ENDP		
830			c	·			
831			Ċ				

•

4,64	41,262	
		140
*******		\$P\$
	ame: INIT_XLAT	_MOD
Ermations 7	a apply a party	es of modifications to the keyboard
¥ Function:⊺ ;¥ 1	translation tab	les
* * Input:		
; <b>H</b>		
;# (SI)=ac ;#	ddress of modif	ication table
;= ;= Output:		
translati	ion tables upda	ted
• #	-	水煤建水水和水水水水水水水水水水水水水水水水水水水水水水水水水水
,	*************	********
INIT_XLAT_MOD I	PROC NEAR	
INIT_XLAT_MOD_	1:	MODIFY INBOUND TRANSLATION TABLE
110V OR	CX, HORD PTR I CX, CX	: IS THIS THE END OF THE TABLE?
JZ	THET WHAT MOD	A . YEE END OF THROUGH XIAT TABLE UP
MOV	DX,WORD PTR [	SI+2] ; NO, MAKE THE SPECIFIED CHANGE
HOV	AH, FUNC_MOD_X	LAT_IN ; THE INBOUND RETBUARD
INT	NDSP_INT_CODE	POINT TO THE NEXT TABLE ENTRY
ADD JHP	SI,4 INIT XLAT_MOD	1 ; PROCESS ALL INBOUND TABLE CHANGES
INIT_XLAT_MOD_		
ADD	SI,4	; POINT TO FIRST OUTBOUND TABLE ENT
INIT_XLAT_MOD_	3:	; MODIFY OUTBOUND TRANSLATION TABLE
MOV	CH,BYTE PTR [	
MOV	DX, WORD PTRIS	51+1)
OR	СК,СН	; IS THIS THE END OF THE TABLE?
JNZ		)_4 ; NO, CONTINUE
RO JZ	DX,DX INIT XLAT HOD	5 ; YES, END OF TRANSLATION TABLE UID
INIT_XLAT_HOD_	6 ·	
- mov -	AH, FUNC_HOD_X	LAT_OUT ; MAKE THE SPECIFIED CHANGE IN
INT	NDSP_INT_CODE	FOUTBOUND TRANSLATION TABLE
ADD JHP	SI,3 INIT_XLAT_MOD	3 ; PROCESS ALL OUTBOUND TABLE CHANNEL
INIT_XLAT_MOD_		
RET		; RETURN TO CALLER
INIT_XLAT_MOD	ENDP	
	Utility Subro	outines
	•	
⋽₩ ⋽₩₩₩₩¥¥₩¥₩₩₩₩₩₩	**************	·圣学家主要子》》王大王是我不成百年不禁非常能能能要要要要要要要要要。
<pre>}* Procedure }*</pre>	Name: INFORM_DF	PERATOR
	To display a me	essage to the system operator
;* ;* Input:		
1*		1. 1. Second
;* (DX) = a ;*	ddress of messa	age to be issued
;* Output:		
;∺ AHisun	predictable	
. #		
;*** <u>*</u> ********	****	紧紧紧接触不在意地推动掌握的掌握的掌握的掌握的人名英法法尔英英法尔斯
INFORM_OPERATO		A TACHT THE NECCOS
HOV INT	AH,9 21H	; ISSUE THE MESSAGE
-		
RET INFORM_OPERATO		; RETURN TO CALLER
INFURN_UPERATO		
NDSPINIT ENDS		
P 1 P3	NOSE TNIT	

.

0203 0203

022E 022E C3 022F

022F 022F 0231

0233 C3 0234

0234

B4 09 CD 21

 8B
 0C

 0B
 C9

 74
 0C

 8B
 54
 02

 B4
 07

 CD
 50

 83
 C6
 04

 EB
 EE

8A 2C 6B 54 01 0A ED 75 04 0B D2 74 09

B4 08 CD 50 &3 C6 03 EB EA

0215 0215 83 C6 04

END NDSP\_INIT

.

APPENDIX F

The IBM Personal Computer MACRO Assembler PAGE F-1 United States EBCDIC Keyboard Translation Tables

3										
4 5								*******	******** Start of Specifica	tions **************
6 7							1 Module	e Name: X	LATUSE	
8							•		me: United States EBCDIC ke	yboard translation tables
10 11							;		Corp 1982	
12 13							3	s: Versio		
14										
15 16							; Funct ;		poard translation tables for poards	the 3278/3279 0.3. Ebulic
17 18							; Notes	:		
19 20							•	endencies	: None	
21 22								trictions	: None	
23							; Modul	e Type: P	Program	
25 26								cessor: A	lssembler	
27 28							-	Point: X	LAT_TABLES_PTR	
29 30								pose: Arr	ay of pointers to keyboard	translation tables
31 32							; ; Extern	nal Refer	ences:	
33 34							i i Con	trol Bloc	:ks:	
35 36							;	NDSPXLAT	- translation table values	
37 38							; ; Messa	ges Issue	d: None	
39 40							;	e Activit		
41							1 -		-	1005 *****
42 43							,		Keyboard Selection Table	
44 45							_		NDSPXLAT.ASH	
46 47							1	LIST	RUSPALKI KSII	
48 49	0000						TABLES	SEGMENT	PUBLIC 'TABLES'	
50 51									<b>4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.</b>	客乐成学家客业本家家家家家家家家家家家家家家家家家家家
52 53							3*			
54 55							;* ;*		SELECTION TABLE	<b>长我子们的长期的的时候,我们的</b> 是有些的人的人,我们还是有些有些的
56 57							;******			
58							VINT TA	PUBLIC	XLAT_TABLES_PTR LABEL WORD	
59 60		0000					APA1210	DW DW	00008.0.0.0	; RESERVED ; APL, NUMERIC LOCK
61 62		0140							0150H, TYPEWRITER, APL, TEXT	; TEXT, NUMERIC LOCK
63 64		0150 06DC	R					DH	0114H, TYPEWRITER, R8K1038,0	
65 66	0018	D114 0000						DH	0160H, TYPEWRITER, AS, 0	; TYPEWRITER, ATTRIB SELECT
67 68	0020	0160 0000						DW	0140H, TYPEWRITER, APL, 0	; APL
69 70	0028	0140 0000								; TEXT
71 72	0030	0150 06DC		R 05F	FA R			DW	0170H, TYPEWRITER, APL, AS	; APL, ATTRIBUTE SELECTT
73 74	0038	0170 0741		R 05	FAR			DM		; DATA ENTRY 2, NUMERIC LOC
75 76	0040	0130	0908	R DE	82 R			DW	0130H, DATA_ENTRY, KP, 0	. DATA ENTRY 1, NUMERIC LOC
77		0120						DW DW	0120H,DATA_ENTRY,0,0 0110H,TYPEWRITER,0,0	; TYPEWRITER, NUMERIC LUCK
78 79		0110 0000						DW	0000K,0,0,0	; RESERVED ; DATA ENTRY 2
80		0130						DW	0130H,DATA_ENTRY,KP,0	
81		0000				00		DW	0120H,DATA_ENTRY,0,0	3 DATA ENTRY 1
82 83	0068	0120 0110	0080	R 00	00 00	00		DW	0110H, TYPEWRITER, 0, 0	; TYPEWRITER ; NO KEYBOARD - DEFAULT TO
84	0078	0110	0080	R 00	00 00	00		DW	0110H, TYPEWRITER, 0,0	; TYPEWRITER
85										
86 87	•					C		INCLUDE	USETYP.ASH	; TYPEWRITER XLAT TABLE
88						C		SUBTTL	Typewriter Keyboard	

# 4,641,262

1	A	4
	-	

Norm         Typewitter Keybaard Translation Table           Typewitter Lable byte         Utable CASE           0000         000000000000000000000000000000000000	89	1	43	~		144	
0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         00000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000 <t< th=""><th>90 91</th><th></th><th></th><th></th><th>****</th><th>""你这个老老家就不能有你这么个个家父子你有你你的你的你的你?"我们不是你们的我们都不是你们不能不能不能</th><th>******</th></t<>	90 91				****	""你这个老老家就不能有你这么个个家父子你有你你的你的你的你?"我们不是你们的我们都不是你们不能不能不能	******
1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         1000         10000         1000         1000 <t< th=""><th>93</th><th></th><th></th><th>C 1#</th><th></th><th></th><th></th></t<>	93			C 1#			
0000         TYPERITE LABL BYTE           1         DHER CASE           0         0         0         0         0         0         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1         1	95			C 1¥			
C         I         CLEAR         CLEAR </td <td></td> <td>0030</td> <td></td> <td>С</td> <td></td> <td></td> <td>*****</td>		0030		С			*****
0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000         0000 <th< td=""><td></td><td></td><td></td><td>C</td><td></td><td></td><td></td></th<>				C			
100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100 <td></td> <td>0080 00 29 00 1</td> <td>N2 00 03</td> <td>C</td> <td></td> <td></td> <td></td>		0080 00 29 00 1	N2 00 03	C			
International and the set of the	102	00 04 00 0	05	С			; 1-5
166         177         167         177         177         177         177           177         0000         0000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         10000         100000         100000         100000         1000000	104	00 09 00 (	DA	C		L, 6,L, 7,L, 8,L, 9,L,10	; 6-10
100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100         100 <td>106</td> <td>00 DE 00 0</td> <td>DF</td> <td>С</td> <td>DB</td> <td>L,11,L,12,L,13,L,14,L,15</td> <td>; 11-15</td>	106	00 DE 00 0	DF	С	DB	L,11,L,12,L,13,L,14,L,15	; 11-15
111       0000       000       14       0000       14       0000       14       0000       14       0000       14       0000       14       0000       14       0000       14       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       0000       15       05       15       15       15       15       15       15       15       15       15       15       15       15       15       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16<	108	00 13 00 1	14		DB	L,16,L,17,L,18,L,19,L,20	; 16-20
112         112         113         00E         00         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120         120	110	00 18 00 1	19		DB	L,21,L,22,L,23,L,24,L,25	; 21-25
113       0000       00       10       10       0000       00       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10	112	00 FF 00 1	28 80 OF 1E		DB	L,26,L,43,U,15,0,SHIFT_LOCK,L,30	1 26-30
115       00CC       00 24 00 25 00 26 c       C       DB       L.36.L.37.L.36.L.37.L.40       ; 30-40         110       0000       00 12 00 05 00 27       C       DB       U.26.L.28.6.L.STLUUSIL.44       ; 41-45         110       0000       00 12 00 05 00 27       C       DB       L.45.L.46.L.47.L.48.L.49       ; 46-50         110       0000       00 12 00 05 00 27       C       DB       L.45.L.46.L.47.L.48.L.49       ; 46-50         112       0015       00 32 00 33 00 34       C       DB       L.45.L.46.L.47.L.48.L.49       ; 61-65         112       0016       00 76 00 10 40 47       C       DB       0.6.0.0.79.U.55.0.CLICK       ; 61-65         113       0110       00 55 80 04 50 01 52       DB       D8       0.6.0.0.0.79.U.55.0.CLICK       ; 61-65         113       0110       00 32 00 32 00 32 00 32       C       DB       U.61.U.62.U.63.U.63.U.64.U.65       ; 76-60         133       0120       00 32 00 32 00 32 00 70       C       DB       U.61.U.62.U.63.U.64.U.65       ; 76-60         134       0120       00 32 00 32 00 70       C       DB       U.61.U.62.U.63.U.64.U.65       ; 76-60         135       0124       00 32 00 32 00 70       C       DB			20 00 21	С	DB	1,31,1,32,1,33,1,34,1,35	; 31-35
117       0000       00       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10       10		0006 00 24 00 2	5 00 26	С	DB	L,36,L,37,L,38,L,39,L,40	; 36-40
110       000 Å       00 20 00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 30 00 31       00 40 47       00 00 00 00 00 00 00 44       00 00 00 00 00 00 44       00 00 00 00 00 00 00 44       00 00 00 00 00 00 00 00 00 00 00 00 00		00D0 80 1A 00 1	C 00 FE	с	OB	U,26,L,28,0,L_SHIFT,U,51,L,44	; 41-45
121       0074       00 \$ 2 00 \$ 3 00 34       C       DB       L150.L151.L152.L153.0.R_SMIFT       : 51-55         123       00FE       00 \$ 6 0 0 10 0 0 4 4 4 7       C       DB       0.RESET.L.28.C.70.L 0R B.70.0.0       : 56-60         124       00FE       00 \$ 70 0 0 0 4 4 4 7       C       DB       0.RESET.L.28.C.70.L 0R B.70.0.0       : 56-60         125       0102       00 \$ 50 0 4 6 0 1 52       DB       0.0.0.0.C.79.U.55.0.CLCK       : 61-65         126       0102       00 \$ 50 0 4 6 0 1 52       DB       U.61.U.62.U.63.U.65.0.CLTK       : 61-65         126       0102       00 \$ 50 0 4 8 00 40       DB       U.61.U.62.U.63.U.63.U.65       : 77-60         133       0120       60 2 60 0 3 80 0 40       C       DB       U.61.U.62.U.63.U.63.U.65       : 76-60         133       0122       40 3D 40 3E 00 00       C       DB       U.61.U.62.U.63.U.63.U.65       : 76-60         133       0124       40 3D 40 3E 00 00       C       DB       U.61.U.62.U.63.U.63.U.65       : 76-60         134       0126       60 6 6 00 7 60 68       DB       U.61.U.62.U.63.U.63.U.64       : 81-65         135       0126       60 6 6 00 7 60 68       DB       U.61.U.7U.64.U.9.U.64       : 81-65 <td>119</td> <td>DODA OD 2D 00 2</td> <td>E 00 2F</td> <td>C</td> <td>DB</td> <td>L,45,L,46,L,47,L,48,L,49</td> <td>; 46-50</td>	119	DODA OD 2D 00 2	E 00 2F	C	DB	L,45,L,46,L,47,L,48,L,49	; 46-50
123 124 125 126 126 126 126 126 126 126 126 126 126	121	00E4 D0 32 00 3	3 00 34	с	DB	L,50,L,51,L,52,L,53,0,R_SHIFT	; 51-55
125       0076       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00	123	00EE DO F8 00 1	C 40 46	С	DB	0,RESET,L,28,C,70,L OR B,70,0,0	; 56-60
1120       012       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       00       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7	125	COF8 CD CO DO O	0 40 4F		DB		
120       0.0       53       0.0       48       0       0       0       0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0	127						
130       00       69       00       39       C       DE       DE <td< td=""><td>129</td><td></td><td></td><td></td><td></td><td></td><td></td></td<>	129						
133       010       40       80       41       C       DB       Understandsstations       1 / 0 - 80         133       012       60       64       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       80       40       40       40       40       40       40       40       40       40       40       40       40       40       40       40       40       40       40       40       40       40       40 <td></td> <td>00 F9 00 3</td> <td>9</td> <td>С</td> <td></td> <td></td> <td></td>		00 F9 00 3	9	С			
134       40 35 40 3C       C       DB       DB       DB       C, 61, C, 62, 0, 0, 0, 0       ; 61-63         135       0124       40 35 40 3E 00 00       C       DB       C, 61, C, 62, 0, 0, 0, 0       ; 86-89         136       0122       60 29 80 28 80 03       C       DB       UPPER CASE          141       50 04 80 05       C       DB       U, 41, U, 43, U, 3, U, 4, U, 5       ; 1-5         142       60 07 80 68       C       DB       U, 41, U, 43, U, 3, U, 4, U, 5       ; 1-5         142       0126       60 08 80 07 80 68       C       DB       U, 41, U, 43, U, 3, U, 4, U, 15       ; 11-15         144       0146 80 08 80 07 80 68       C       DB       U, 11, U, 12, U, 13, U, 14, L, 15       ; 11-5         145       60 10 80 11 80 12       C       DB       U, 11, U, 12, U, 13, U, 14, L, 15       ; 11-5         146       015 80 16 80 17       C       DB       U, 21, U, 22, U, 23, U, 24, U, 25       ; 21-25         150       016 16 80 12       C       DB       U, 21, U, 23, U, 33, U, 34, U, 35       ; 31-35         151       00 FF 80 28 26       C       DB       U, 21, U, 23, U, 33, U, 34, U, 35       ; 31-35         151       0176 80 32 00 33 00 34		80 40 80 4	1	С			
136       137       136       137       138       140       141       152       15       15       15       15       15       15       15       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16		40 3B 40 30	C	C			
138       139       0       0       132       80       29       80       28       0       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14       14 <td< td=""><td>136</td><td></td><td></td><td>С</td><td>08</td><td>C,61,C,62,0, 0,0, 0</td><td>; 86-89</td></td<>	136			С	08	C,61,C,62,0, 0,0, 0	; 86-89
140       0132       80       29       00       28       00       1       5       5       1       5       5       1       5       5       1       5       5       5       1       5       5       5       1       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5       5 <td< td=""><td>138</td><td></td><td></td><td>C ;</td><td>UPPER</td><td>CASE</td><td></td></td<>	138			C ;	UPPER	CASE	
142       013C       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60       60	140			С	DB	0,41,0,43,0, 3,0, 4,0, 5	; 1-5
144       0146       00 00 00 00 00       00 00 00       00 00 00       00 00       00 00       00 00       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       111111       1111111       1111111       1111111       1111111       1111111       1111111       1111111       1111111       1111111       1111111       1111111       111111       111111       111111       111111       111111       111111       1111111       1111111       1111111       1111111       11111111	142	0130 80 06 80 07	7 80 08	С	DB	U, 6,U, 7,U, 8,U, 9,U,10	; 6-10
166       0150       80       10       00       00       00       00       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16       16	144	0146 80 0B 80 00	C 80 0D	С	DB	U,11,U,12,U,13,U,14,L,15	; 11-15
143       015 & 00 14       0       015 & 00 14       0       015 & 00 14       0       015 & 00 14       0       015 & 00 14       015 & 00 14       015 & 00 14       015 & 00 15       016 & 00 17       0       0       016 & 00 17       0       016 & 00 17       0       016 & 00 17       0       016 & 00 17       0       016 & 00 17       0       016 & 00 17       0       016 & 00 17       0       0       016 & 00 17       0       0       016 & 00 17       0       0       016 & 00 16       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0 <td>146</td> <td>0150 80 10 80 11</td> <td>1 80 12</td> <td></td> <td>DB</td> <td></td> <td></td>	146	0150 80 10 80 11	1 80 12		DB		
149       80       18       80       19       C       Definition of the second of the	148				DB		
151       00       FF 80       1E       C       DB       U,31,U,32,U,33,U,34,U,35       ; 31-35         152       016       80       1F 80       20       80       23       ;       DB       U,31,U,32,U,33,U,34,U,35       ; 31-35         154       0176       80       22       80       23       C       DB       U,36,U,37,U,38,U,39,U,40       ; 36-40         155       60       27       80       26       C       DB       U,27,U,28,0,L_SHIFT,U,52,U,44       ; 41-45         157       60       32       00       31       C       DB       U,45,U,46,U,47,U,48,U,49       ; 46-50         158       018C       60       20       80       31       C       DB       U,50,L,51,L,52,U,53,0,R_SHIFT       ; 51-55         160       0196       80       32       00       30       03       C       DB       U,50,L,51,L,52,U,53,0,R_SHIFT       ; 51-55         161       80       35       00       F7       C       DB       0, 00, 0,C,75,U,55,0,CLICK       ; 61-65         163       6146       01AD       00       F8       80       3F       C       DB       0, 0,0,0,C,75,U,55,0,CLICK       ; 61-65	149 150	80 18 80 19 0164 80 02 00 18	80 0F				
153       60       22       80       23       010       013140,22,0,33,0,34,0,35       ; 31-35         154       0176       80       24       80       25       80       25       010       013140,22,0,33,0,34,0,35       ; 31-35         154       0176       80       24       80       25       80       26       C       DB       U,36,U,37,U,38,U,39,U,40       ; 36-40         155       60       27       80       24       80       16       80       16       60       164       144       1445         157       60       34       80       16       0176       80       31       6       141       1445         158       0186       80       20       33       00       34       C       DB       U,45,U,46,U,47,U,48,U,49       ; 41-45         159       80       33       80       31       C       DB       U,50,L,51,L,52,U,53,0,R_SHIFT       ; 51-55         161       0140       00       78       80       16       C       DB       U,60,0,0,C,75,U,55,0,CLICK       ; 61-65         163       81       46       00       70       67       C       DB <t< td=""><td>151 152</td><td>DO FF 80 16</td><td></td><td>C</td><td></td><td></td><td></td></t<>	151 152	DO FF 80 16		C			
155       60 27 80 26       C       DB       U,27,U,28,0,L_SHIFT,U,52,U,44       ; 41-45         156       0162       80 18 80 1C 00 FE       C       DB       U,27,U,28,0,L_SHIFT,U,52,U,44       ; 41-45         157       80 34 60 2C       C       DB       U,45,U,46,U,47,U,48,U,49       ; 46-50         158       016C       80 32 00 33 00 34       C       DB       U,50,L,51,L,52,U,53,0,R_SHIFT       ; 51-55         161       80 35 00 FD       C       DB       U,50,L,51,L,52,U,53,0,R_SHIFT       ; 56-60         163       81 46 02 00       C       DB       D,RESET,U,26,C,70,U OR B,70,0,0       ; 56-60         164       01A0 00 F8 80 1C 40 46       C       DB       0, 0.0, 0,C,75,U,55,0,CLICK       ; 61-65         164       01A4 00 00 00 00 40 4F       C       DB       0, PAUSE_KEY,0,DEC_INP,L OR B,82,L,63,L,72       ; 66-70         165       80 37 00 F7       C       DB       L,80,L,75,L,77,0,ALT_SHIFT,U,57       ; 71-75         166       01B4 00 F2 00 F6 01 52       C       DB       L,80,L,75,L,77,0,ALT_SHIFT,U,57       ; 71-75         167       00 53 00 48       C       DB       U,61,U,62,U,63,U,64,U,65       ; 76-80         171       80 42 80 43 80 3F       C       DB <td>153</td> <td>80 22 80 23</td> <td>5</td> <td>С</td> <td></td> <td></td> <td></td>	153	80 22 80 23	5	С			
157       80       34       80       20       0       0       0       0       0       177,0,22,0,14,44       ;       41-45         158       018C       80       22       80       22       80       22       80       22       80       22       80       22       80       22       80       22       80       22       80       22       80       22       80       22       80       22       80       33       00       30       80       33       00       34       C       DB       U,45,U,46,U,47,U,48,U,49       i       46-50         160       0196       80       32       00       33       00       34       C       DB       U,50,L,51,L,52,U,53,O,R_SHIFT       i       51-55         161       80       37       00       F8       60       164       0144       00       00       00       44       F       C       DB       U,50,L,51,L,52,U,53,O,R_SHIFT       i       56-60         163       80       37       00       F7       C       DB       0,0,0,0,C,75,U,0R       B,70,0,0       i       66-70         164       0184       00       F2	155	80 27 80 28	6	C			; 36-40
159       80 30 60 31       C       DB       019510145004700480049       ; 46-50         160       0196       80 32 00 33 00 34       C       DB       U,50,L,51,L,52,U,53,0,R_SHIFT       ; 51-55         161       80 35 00 FD       C       DB       0,80,0,0,0,0,0,0,0,0       i 56-60         162       01AD       00 F8 80 1C 40 46       C       DB       0,80,0,0,0,0,0,0,0,0       i 56-60         163       81 46 DC 00       C       DB       0,0,0,0,0,75,0,0,0,0       i 56-60         164       01AA       00 00 00 40 4F       C       DB       0,0,0,0,0,75,0,0,0,0,0       i 56-60         165       80 37 00 F7       C       DB       0,0,0,0,0,75,0,0,0,0,0       i 56-60         166       01B4       00 F2 00 F6 01 52       C       DB       0,0,0,0,0,75,0,0,0,0       i 66-70         166       01B2       05 30 0 48       C       DB       1,80,1,75,1,77,0,ALT_SHIFT,0,57       i 71-75         167       05 00 0 48 00 32       C       DB       U,61,U,62,U,63,U,64,U,65       i 76-80         168       01B2 40 30 40 3E       D 00       C       DB       U,61,U,62,U,63,U,64,U,65       i 81-85         173       40 3B 40 3C       C       DB       U,	157	80 34 80 20	:	с			; 41-45
161       80 35 00 FD       C       DB       D,RESET,U,26,C,70,U DR B,70,0,0       I 56-60         163       61 46 0C 00       C       DB       D,RESET,U,26,C,70,U DR B,70,0,0       I 56-60         164       01AA       00 00 00 40 4F       C       DB       0,00,0,C,75,U,55,0,CLICK       I 61-65         165       80 37 00 F7       C       DB       0,00,0,C,75,U,55,0,CLICK       I 61-65         166       01B4       00 F2 00 F6 01 52       C       DB       0,PAUSE_KEY.0,DEC_INP.L OR B,82,L,83,L,72       I 66-70         166       01BE       00 50 00 4B 00 4D       C       DB       L,80,L,75,L,77,0,ALT_SHIFT,U,57       I 71-75         167       00 F9 80 39       C       DB       U,61,U,62,U,63,U,64,U,65       I 81-85         170       01C6       60 3D 60 3E 80 3F       C       DB       U,61,U,62,U,63,U,64,U,65       I 81-85         171       80 40 80 41       C       DB       U,66,U,67,U,68,C,59,C,60       I 81-85         174       01DC 40 3D 40 3E 00 00       C       DB       C,61,C,62,0,0,0,0       I 86-89         175       00 00       C       C       DB       C,61,C,62,0,0,0,0       I 86-89         175       00 00       C       C       D	159	80 30 80 31		с		U,45,U,46,U,47,U,48,U,49	; 46-50
162       01AD       00       F8       80       1C       40       46       C       C       DB       D,RESET,U,26,C,70,U DR B,70,0,0       ; 56-60         163       01AA       00       00       00       00       46       C       C       DB       D,RESET,U,26,C,70,U DR B,70,0,0       ; 56-60         164       01AA       00       00       00       46       C       DB       0,0,0,0,C,75,U,55,0,CLICK       ; 61-65         165       01B4       00       F6       01       52       C       DB       0,PAUSE_KEY,0,DEC_INP,L OR B,82,L,83,L,72       ; 66-70         166       01B4       00       F6       01       52       C       DB       0,PAUSE_KEY,0,DEC_INP,L OR B,82,L,83,L,72       ; 66-70         166       01B4       00       F6       01       52       C       DB       L,80,L,75,L,77,0,ALT_SHIFT,U,57       ; 71-75         166       01B2       80       80       81       81       6       5       76-80         170       01C6       80       92       C       DB       U,61,U,62,U,63,U,64,U,65       ; 76-80       ; 81-85         173       40       38       40       20       C	160 161				DB	U,50,L,51,L,52,U,53,0,R_SHIFT	\$ 51-55
164       01AA       00       00       00       00       44       0       0       00       64       5       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-65       5       61-70       5       5       7       61-70       5       5       7       61-70       5       5       7       61-70       5       7       5       7       5       7       5       7       5       7       5       5       7       5       5 <t< td=""><td>162 163</td><td>01A0 00 F8 80 1C</td><td>48 46</td><td>с</td><td>DB</td><td>0,RESET,U,26,C,70,U OR 8,70,0,0</td><td>\$ 56-60</td></t<>	162 163	01A0 00 F8 80 1C	48 46	с	DB	0,RESET,U,26,C,70,U OR 8,70,0,0	\$ 56-60
166       01B4       00       F2       00       F6       01       52       C       DB       0,PAUSE_KEY,0,DEC_INP,L OR B,82,L,83,L,72       ; 66-70         167       00       53       00       48       C       DB       L,80,L,75,L,77,0,ALT_SHIFT,U,57       ; 71-75         168       01EE       00       59       00       48       00       F7       ; 76-80         170       01C6       60       3D       80       3F       C       DB       U,61,U,62,U,63,U,64,U,65       ; 76-80         171       80       40       80       43       80       44       C       DB       U,66,U,67,U,68,C,59,C,60       ; 81-85         172       01D2       80       42       80       43       80       40       81-85         174       01DC       40       3D       40       3E       00       00       C       DB       C,61,C,62,C,0,0,0       ; 86-89         175       00       00       00       00       00       00       ; 86-89       ;         175       00       00       00       00       00       00       ; 1-5         175       00       00       00 <td>164</td> <td>01AA 00 00 00 00</td> <td>40 4F</td> <td>С</td> <td>DB</td> <td>0, 0.0, 0,C,79,U,55,0,CLICK</td> <td>; 61-65</td>	164	01AA 00 00 00 00	40 4F	С	DB	0, 0.0, 0,C,79,U,55,0,CLICK	; 61-65
168       01EE       C0       50       00       48       00       40       C       DB       L,80,L,75,L,77,0,ALT_SHIFT,U,57       ; 71-75         169       00       F9       80       39       C       DB       U,61,U,62,U,63,U,64,U,65       ; 76-80         171       60       40       80       41       C       DB       U,61,U,62,U,63,U,64,U,65       ; 76-80         172       01D2       80       42       80       44       C       DB       U,66,U,67,U,68,C,59,C,60       ; 81-85         173       40       3B       40       3E       00       C       DB       C,61,C,62,0,0,0,0       ; 86-89         175       00       00       C       DB       C,61,C,62,0,0,0,0       ; 86-89       ; 86-89         176       C       FERSONAL COMPUTER CONTROL CASE       C       ; 9       9       ; 1-5       ; 1-5         176       C       C       DB       0,0,0,0,0,0,0,0,0       ; 1-5       ; 5         177       0100       00       00       00       00       ; 5       ; 5         177       C       FERSONAL COMPUTER CONTROL CASE       ; 1-5       ; 5       ; 6       ; 6 <t< td=""><td>166</td><td>0184 00 F2 00 F6</td><td>01 52</td><td>С</td><td>DB</td><td>0,PAUSE_KEY.0,DEC_INP.L OR B,82,L,83,L,72</td><td>; 66-70</td></t<>	166	0184 00 F2 00 F6	01 52	С	DB	0,PAUSE_KEY.0,DEC_INP.L OR B,82,L,83,L,72	; 66-70
169       00       F9       80       39       C         170       01C6       60       30       80       35       C       DB       U,61,U,62,U,63,U,64,U,65       ; 76-80         171       80       40       80       41       C       DB       U,61,U,62,U,63,U,64,U,65       ; 76-80         172       01D2       80       42       80       44       C       DB       U,66,U,67,U,68,C,59,C,60       ; 81-85         173       40       3B       40       3E       00       00       C       DB       C,61,C,62,0,0,0,0       ; 86-89         174       01DC       40       3E       00       00       C       DB       C,61,C,62,0,0,0,0       ; 86-89         175       00       00       C       DB       C,61,C,62,0,0,0,0       ; 86-89         176       C       C       C       C       C       C         176       C       ;       PERSONAL COMPUTER CONTROL CASE       C         178       C       ;       PERSONAL COMPUTER CONTROL CASE       ;       1 - 5         179       01E4       00       00       00       00       ;       1 - 5         16	168	01BE 00 50 00 4B	00 4D	С	DB	L,80,L,75,L,77,0,ALT_SHIFT,U,57	; 71-75
171       60       40       80       41       C         172       0102       80       42       80       43       80       44       C       DB       U,66,U,67,U,68,C,59,C,60       7       81-85         173       40       36       40       36       0       36       0       36       68       7         174       01DC       40       30       40       3E       00       00       7       7       66       7       86-89       7       7       60       7       7       60       7       7       7       60       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7       7 <t< td=""><td>170</td><td>01C8 80 3D 80 3E</td><td>80 3F</td><td>c</td><td>ΟВ .</td><td></td><td>; 76-80</td></t<>	170	01C8 80 3D 80 3E	80 3F	c	ΟВ .		; 76-80
173       40       3B       40       3C       C         174       01DC       40       3D       00       00       C       DB       C,61,C,62,0,0,0       ;       86-89         175       00       00       C       DB       C,61,C,62,0,0,0       ;       86-89         176       C       C       FERSONAL COMPUTER CONTROL CASE       ;       1       1       5         177       C       ;       PERSONAL COMPUTER CONTROL CASE       ;       1       5         178       C       DB       0,0,0,0,0,0,0,0,0,0       ;       1       5         178       C       DB       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	171 172	0102 80 42 80 43	80 44				
175       00 00       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C       C </td <td>173 174</td> <td>40 3B 40 3C</td> <td></td> <td>C</td> <td></td> <td></td> <td></td>	173 174	40 3B 40 3C		C			
77         C         ;         PERSONAL COMPUTER CONTROL CASE           78         C         C         DB         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	175 176			с			, 00-07
79       01E4       00       00       00       03       C       DB       0, 0, 0, 0, C, 3, 0, 0, 0, 0       ; 1-5         80       00       00       00       00       C       DB       0, 0, 0, C, 7, 0, 0, 0, 0, 0       ; 1-5         81       D1EE       00       00       00       C       DB       0, 0, C, 7, 0, 0, 0, 0, 0, 0, 0       ; 6-10	177			С ;	PERSONA	L COMPUTER CONTROL CASE	
81 DIEE 00 00 40 07 00 00 C DB 0, 0,C, 7,0, 0,0, 0,0, 0 ; 6-10	179		40 03	С	08	0, 0,0, 0,C, 3,0, 0,0, 0	; 1-5
62 00 00 00 00 C	181	DIEE 00 00 40 07	00 00	c	08	0, 0,C, 7,0, 0,0, 0,0, 0	; 6-10
	182	00 00 00 00		с	•		

		4,	641,262	
	145		146	
01F8		C DB	0, 0,C,12,0, 0,C,14,0, 0	; 11-15
0202	40 DE 00 DD 40 10 40 11 40 12	C C DB	C, 16, C, 17, C, 18, C, 19, C, 20	; 16-20
0200	40 13 40 14 40 15 40 16 40 17	C C D8	C, 21, C, 22, C, 23, C, 24, C, 25	; 21-25
	40 18 40 19 40 18 40 28 40 47	C C DB	C,26,C,43,C,71,0,SHIFT_LOCK,C,30	; 26-30
	00 FF 40 1E	C DB	C, 31, C, 32, C, 33, C, 34, C, 35	; 31-35
	40 1F 40 20 40 21 40 22 40 23	C	C,36,C,37,C,38,O, 0,D, 0	; 36-40
	40 24 40 25 40 26 00 00 00 00	C DB C		; 41-45
0234	40 1B 40 1C 00 FE 00 00 40 2C	C DB C	C,27,C,28,D,L_SHIFT,D, 0,C,44	; 46-50
023E	40 2D 40 2E 40 2F 40 30 40 31	C DB C	C,45,C,46,C,47,C,48,C,49	; 51-55
0248	40 32 00 00 00 00 00 00 00 FD	C DB C	C,50,0, 0,0, 0,0, 0,0,R_SHIFT	
0252	00 F8 40 1C 00 00 00 00 00 00	C DB	0,RESET,C,28,0, 0,0, 0,0, 0	; 56-60
025C	00 00 00 00 00 00 40 37 00 00	C DB	0, 0,0, 0,0, 0,0,55,0, 0	; 61-65
0266	00 00 00 00 00 00	C DB	0, 0,0, 0,0, 0,C OR A,63,C,73	; 66-70
0270	60 53 40 49 40 51 00 00 00 00	C DB	C,81,0, 0,0, D,0,ALT_SHIFT,C,57	1 71-75
027Å	00 F9 40 39 00 00 00 00 00 00	C DB	0, 0,0, 0,0, 0,0, 0,0, D	; 76-80
0284	00 00 00 00 00 00 00	C C DB	0, 0,0, 0,0, 0,0, 0,0, 0	; 81-85
028E	00 00 00 00 00 00 00 00	C C DB	0, 0,0, 0,0, 0,0, D	; 86-89
	00 00	C C		
		C 3 PEP.	SONAL COMPUTER ALT CASE	. 1 5
0296	00 00 20 02 20 03 20 04 20 05	C DB C	0, 0,4, 2,4, 3,4, 4,4, 5	; 1-5
DASO	20 06 20 07 20 08 20 09 20 0A	C DB	A, 6,A, 7,A, 8,A, 9,A,10	; 6-10
02AA	20 0B 20 CC 20 DD	C DB	A,11,A,12,A,13,0, 0,0, 0	; 11-15
0284	00 00 00 00 20 10 20 11 20 12	C DB	A, 16, A, 17, A, 18, A, 19, A, 20	; 16-20
02BE	20 13 20 14 20 15 20 16 20 17	C DB	A,21,A,22,A,23,A,24,A,25	; 21-25
02C8	20 18 20 19 00 00 60 00 00 00	C DB	0, 0,0, 0,0, 0,0,SHIFT_LCCK,A,30	; 26-30
02D2	00 FF 20 IE 20 1F 20 20 20 21	С С DB	A, 31, A, 32, A, 33, A, 34, A, 35	; 31-35
02DC	20 22 20 23 20 24 20 25 20 26	C DB	A,36,A,37,A,38,D, D,D, D	; 36-40
0266	00 00 00 00 00 00 00 00 00 FE	C DB	0, 0,0, 0,0,L_SHIFT,0, 0,4,44	; 41-45
	00 00 20 2C 20 2D 20 2E 2C 2F	C DB	A, 45, A, 65, 2, 47, A, 46, A, 49	: 46-50
	20 30 20 31 20 32 00 00 00 00	C DB	A,50,0, 0,0, 0,0, 0,0,R_SHIFT	1 51-55
	00 00 00 FD	C DB	0,RESET,0, 0,0, 0,0, 0,0, 0	; 56-60
0304	00 F8 00 00 00 00 00 00 00 00	C		; 61-65
030E	00 00 00 00	C DB C	0, 0,0, 0,0, 0,0, 0,0, 0	; 66-70
0318	00 00 00 00	C DB C		; 71-75
0322	00 00 00 00 00 00 00 00 F9 20 39	C DB C	0, 0,0, 0,0, 0,0,ALT_SHIFT,4,57	; 76-80
032C	00 00 00 00 00 00 00 00 00 00 00 00	C DB C	0, 0,0, 0,0, 0,0, 0,0, 0	1 81-85
0336	00 00 00 00 00 00 00 00 00 00 00 00	C DB	0, 0,0, 0,0, 0,0, 0,0, 0	1 86-89
0340	00 00 00 00 00 00 00 00	C DB	0, 0,0, 0,0, 0,0, 0	,,
			CASE	
0149	00 01 00 3B 00 3C	C DB	L, 1,L,59,L,60,L,61,L,62	; 1-5
	00 3D 00 3E	C DB	L,63,L,64,L,65,L,66,L,67	; 6-10
	00 3F 00 40 00 41 00 42 00 43	C	L,68,U,59,U,60,0, 0,L,79	; 11-15
0350	00 00 00 4F	C	0,59,0,60,0,61,0,62,0,63	; 16-20
0366	80 3E 80 3F	C DB C		; 21-25
0370	80 40 80 41 80 42 80 43 80 44	C DB C	U,64,U,65,U,66,U,67,U,68	; 26-30
037A		C DB C		; 31-35
0384		C DB		; 36-40
038E		C DB	C,65,C,66,C,67,C,68,D, D	
0398	00 00 40 1C 00 FE	C DB	0, 0,C,28,0,L_SHIFT,0, 0,A,59	; 41-45
03A2	00 00 20 3B 20 3C 20 3D 20 3E	C DB	A,60,A,61,4,62,A,63,A,64	; 46-50
	20 3F 20 40	<b>.</b> .		

-

# 4,641,262

	148	
DB	A,65,4,66,4,67,A,68,0,R_SHIFT	; 51-55
DB	0,RESET,C.28,0,0,C,71,0, 0	; 56-60
DB	C,81,0, D,0, 0,0, 0,0, D	; 61-65
DB	0,CTL_KEY,0,ALT_KEY,0,SWITCH_MODE,0, 0,L,73	; 66-70
DB	L,31,C,75,C,77,0,ALT_SHIFT,L,57	; 71-75
DB	0, 0,0, 0,0, 0,0, 0,0, 0	; 76-80
DB	0, 0,0, 0,0, 0,0, 0,0, 0	; 81-85
DB	0, 0,0, 0,0, 0,0, 0	; 86-89
ENCODED	KLYSTROKE TRANSLATION TABLE	
DB	0. 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0	; 00-07
DB	L.14.L.15.0, 0.0, 0.0, 0.L.42.D, 0.0, 0	; 08-0F
DB	D, D,D, D,D, D, <b>D, D,O, D,O, D,O, O,O, O</b>	3 10-17
DB	D, D,O, O,O, D.A,63.0, D,O, D,O, D,O, D	; 18-1F
BG	L,75,U,26,U,40,U, 4,U, 5,U, 6,U, 8,L,40	; 20-27
DB	U,10,U,11,U, 9,U,13,L,52,L,12,L,53,L.54	; 28-2F
DB	L,11,L, 2,L, 3,L, 4,L, 5,L, 6,L, 7,L, 8	; 30-37
DB	L, 9,L,10,U,39,L,39,L,44,L,13,U,44,U,54	; 36-3F
DB	U, 3,U,30,U,49,U,47,U,32,U,18,U,33,U,34	; 40-47
DB	U, 35, U, 23, U, 36, U, 37, U, 38, U, 51, U, 50, U, 24	; 48-4F ; 50-57
DB	U,25,U,16,U,19,U,31,U,20,U,22,U,48,U,17	, 50-57
DB	U,46,U,21,U,45,D, 0,L,27,0, 0,0, 0,U,12	; 58-5F
DB	L, 1,L,30,L,49,L,47,L,32,L,18,L,33,L,34	; 60-67
DB	L,35,L,23,L,36,L,37,L,38,L,51,L,50,L,24	; 68-6F
DB	L,25,L,16,L,19,L,31,L,20,L,22,L,48,L,17	; 70-77
DB	L,46,L,21,L,45,L,41,U.27,U,41,U, 1,0, 0	; 78-7F
DB	U, 2,U, 7,L,26,L,14,L,15,L,28,L,42,A,28	; 50-67
DB	L,56,A,56,L,57,A, 0,BA,0,0, 0,D, 0,C, 0	; 68-8F
DB	L,58,A,58,L,59,A,59,L,60,A,60,L,61,A,61	; 90-97
08	L,62,A,62,L,63,A,63,L,64,A,64,L,65,A,65	; 98-9F 1 AD-A7
DB	L,66,A,66,L,67,A,67,L,68,A,68,L,69,A,69	; 48-AF
DB	L,70,A,70,L,71,A,71,L,72,A,72,L,73,A,73	; 80-87
DB	A, 2,A, 3,A, 4,A, 5,A, 6,A, 7,A, 8,A, 9	; BE-BF
DB	A, 10, A, 11, A, 12, A, 13, L, 76, L, 77, L, 78, L, 79	; C(-C7
DB	L,80,L,81,L,82,L,83,L,64,L,85,L,86,L,87	; C8-CF
DB	D, 0,0, 0,0, <b>0,0, 0,</b> 0, 0,0, 0,0, 0,0, 0	,

$ \begin{array}{cccccccccccccccccccccccccccccccccccc$						14	7		
279       0366       00       64       40       47       00       00         281       03C0       40       67       00       00       00         283       03C0       60       60       60       67       00       75       00         285       03D4       00       51       00       60       64       40       40         286       03D4       00       51       40       48       40       40         286       03E8       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00		03AC					20	43	
281         0 3C0         40         51         00         00         00           282         03CA         00         64         05         30         64         64         40           283         03CA         00         64         64         40         64         40           286         03C0         05         44         64         40         00         73         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00	279	03B6	0 D	F8	40	10	00	00	
283         03CA         00         F4         00         F3         D0         F5           284         03         00         00         00         00         00         00           285         0304         00         00         00         00         00         00         00           286         00         00         00         00         00         00         00         00           290         03F2         00         00         00         00         00         00         00           293         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00	281	03C0	40	51	80	00	00	00	
285       0304       00       51       40       48       40       40         286       030       00       00       00       00       00       00         287       031E       00       00       00       00       00       00       00         289       032E       00       00       00       00       00       00       00         291       0372       00       00       00       00       00       00       00       00         293       00       00       00       00       00       00       00       00       00       00         294       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00	283	03CA	00	F4	00	F3	00	FS	
287       03DE       00       00       00       00       00       00       00         289       03F2       00       00       00       00       00       00         291       03F2       00       00       00       00       00       00       00         293       00       00       00       00       00       00       00       00         294       00       00       00       00       00       00       00       00         296       03FA       00       00       00       00       00       00       00       00         296       03FA       00       00       00       00       00       00       00       00         297       040A       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00	285	0304	00	51	40	4B	40	40	
289         03E8         00         00         00         00         00         00         00           291         03F2         00         00         00         00         00         00           292         00         00         00         00         00         00         00           294         294         294         294         294         294         294           295         297         040A         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00	287	03DE	00	00	00	80	00	00	
291         03F2         00         00         00         00         00         00           293         00         00         00         00         00         00         00         00           296         03FA         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         <	289	03E8	00	00	00	00	00	00	
293         294         295         296       03FA       00       00       00       00       00         297       00       00       00       00       00       00       00         299       040A       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       0	291	03F2	00	00			00	00	
295         296       03 $FA$ 00       00       00       00       00         297       040A       00       00       00       00       00         300       00       00       00       00       00       00       00         300       00       00       00       00       00       00       00       00         301       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00 <t< td=""><td>293</td><td></td><td>00</td><td>00</td><td></td><td></td><td></td><td></td><td></td></t<>	293		00	00					
297         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         0	295								
299         040A         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00 <th< td=""><td>297</td><td>03FA</td><td>00</td><td>00</td><td>00</td><td>00</td><td></td><td></td><td></td></th<>	297	03FA	00	00	00	00			
301         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         0	299	840A	00	DE	00	OF			
303         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         0	301		00	00	00	00			
305 $042A$ 00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         <	303	041A	00	00	00	00			
307         00.00         00         00           308 $043A$ 00         48         60         1A         80         28           310 $043A$ 00         48         00         1A         80         28           311 $044A$ 80         0.4         80         0.0         34         00         C           313         00         0.5         0.0         36         0.0         0.0         33           314         045A         0.0         08         0.0         20         0.0         33           315         0.0         0.7         0.0         80         27         0.0         31           316         0.0         27         0.0         2.0         0.0         80         27           318         0.0         2.6         0.3         60         1.8         80         2.8         31           321         80         2.7         80         2.8         80         2.8         80         2.8         33           322         04AA         80         2.6         80         31         80         1.7         80	305	042A	00	00	00	00			
309 $\varepsilon D$ 60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60         60 <t< td=""><td></td><td></td><td>00.</td><td>DD</td><td>OD.</td><td>nn</td><td></td><td></td><td></td></t<>			00.	DD	OD.	nn			
311 $044A$ 80       0A       80       0B       80       00       03       40       0C         312       00       05       00       34       00       0C       313       00       34       00       0C       03       50       00       03       50       00       03       50       00       02       00       03       313       00       04       00       05       00       04       00       07       00       08       02       00       03       03       03       03       03       03       03       04       00       07       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00       00		043A	eD			05			
313       00       35       00       36       00       36         314       045A       00       08       00       02       00       36         315       00       04       00       05       00       06         317       046A       00       09       00       04       80       27         318       00       27       00       22       00       00       12         320       047A       80       360       16       80       31         321       80       27       60       22       32         323       048A       80       25       80       16       80       13         322       80       17       80       17       80       12       33         325       80       17       80       18       80       18       80       18         326       049A       80       19       80       10       80       13         327       80       17       80       16       80       16       80       13         327       80       16       80       16 <t< td=""><td>311</td><td>0444</td><td></td><td>0Å</td><td>80</td><td>0B</td><td></td><td>÷ ·</td><td></td></t<>	311	0444		0Å	80	0B		÷ ·	
315         00         04         00         05         00         06           316         00         07         00         08         00         07         00         08           317         046A         00         07         00         0A         00         0D           319         60         07         00         0A         00         1A         0D           320         047A         80         03         60         1E         80         31           321         80         2F         60         20         80         12           322         80         21         80         22         80         18         24           322         80         18         80         25         80         18         33           325         80         17         80         17         80         14         80         16           326         04AA         80         2E         80         18         80         20         00         13           333         00         01         00         16         00         12         33         33	313		00	35	00	36			
317 $046A$ $00$ $07$ $00$ $27$ $00$ $27$ $00$ $27$ $00$ $27$ $00$ $27$ $00$ $27$ $00$ $27$ $00$ $27$ $00$ $27$ $00$ $27$ $00$ $27$ $00$ $20$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $20$ $00$ $22$ $323$ $048A$ $80$ $23$ $80$ $12$ $80$ $13$ $322$ $80$ $12$ $80$ $17$ $80$ $24$ $333$ $225$ $80$ $16$ $80$ $10$ $80$ $11$ $322$ $80$ $13$ $80$ $11$ $322$ $333$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $0$	315	045A	00	04	00	05			
319         80         2C         80         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         37         80         30         80         30         30         30         30         30         31         30         31         31         31         31         31         31         31         31         31         31         3	317	046A	00	09	00	AO			
321       80       2F       60       20       80       12         322       048A       80       23       80       12       80       21         323       048A       80       23       80       17       80       24         324       80       32       80       18       80       33         325       80       17       80       14       80       13         326       049A       80       18       80       17       80       13         326       049A       80       26       80       18       80       20       80       13         329       04AA       80       26       80       18       80       20       80       11         329       04AA       60       21       00       21       00       21       00       21       31       33       33       00       10       10       11       31         330       04CA       00       21       00       22       00       13       33       33       33       33       33       33       33       33       33       33       3	319		80	2C	80	36			
323       048A       80       23       60       17       80       24         324       80       25       80       33       33         325       80       32       80       18       33         326       049A       80       19       80       10       80       13         327       80       16       80       16       80       13         327       80       16       80       16       80       13         327       94AA       80       26       80       11         329       04AA       80       26       80       11         329       04AA       00       10       01       80       12         330       00       21       00       22       33       31       00       21       00       22         333       04CA       00       10       00       10       01       13         336       04DA       01       10       16       00       13         336       04CA       01       10       11       14       10       16         344       04FA	321	047A	80	2F	80	20			
325 $80$ $32$ $80$ $18$ $326$ $049A$ $80$ $19$ $80$ $16$ $80$ $13$ $327$ $80$ $17$ $80$ $16$ $80$ $11$ $329$ $04AA$ $80$ $2E$ $80$ $15$ $80$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $31$ $320$ $04AA$ $80$ $2E$ $80$ $15$ $80$ $00$ $33$ $30$ $00$ $00$ $00$ $10$ $1E$ $00$ $13$ $333$ $00$ $2F$ $00$ $22$ $333$ $00$ $21$ $00$ $21$ $00$ $23$ $00$ $12$ $333$ $334$ $00$ $12$ $00$ $21$ $00$ $12$ $333$ $337$ $00$ $17$ $00$ $16$ $00$ $11$ $00$ $14$ $00$ <	323	048A	80	23	80	17			
327 $80$ $1F$ $80$ $14$ $80$ $16$ $328$ $60$ $30$ $80$ $11$ $329$ $04AA$ $80$ $2E$ $80$ $15$ $80$ $20$ $330$ $00$ $00$ $00$ $00$ $00$ $00$ $00$ $331$ $00$ $00$ $00$ $10$ $00$ $10$ $00$ $11$ $333$ $00$ $21$ $00$ $22$ $00$ $12$ $334$ $00$ $21$ $00$ $22$ $00$ $12$ $334$ $00$ $21$ $00$ $22$ $00$ $12$ $335$ $04CA$ $00$ $25$ $00$ $10$ $00$ $13$ $336$ $04DA$ $00$ $17$ $00$ $14$ $00$ $16$ $340$ $00$ $16$ $00$ $16$ $00$ $11$ $016$ $00$ $11$ $344$ $04FA$ $80$ $28$ $00$ $00$	325		80	32	80	18			
329 $04AA$ $80$ $2E$ $80$ $15$ $80$ $20$ $330$ 00       00       00       18       00       00 $331$ 00       00       00       18       00       01 $332$ 04BA       00       01       00       22       03       31 $333$ 00       27       00       20       01       12 $334$ 00       21       00       22       03       33 $335$ 04CA       00       23       00       17       00       24 $336$ 04DA       01       90       16       00       13 $337$ 00       17       00       24       00       11 $340$ 00       16       00       11       341       04EA       00       10       01       11 $341$ 04EA       00       28       01       10       00       14       00       14 $344$ 04FA       80       02       80       10       00       00       00       00	327	049A	80	1F	80	14		-	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	329	04AA	80	2E	80	15			
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	331	04.84	00	00	80	00			
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	333	UNDA	00	2F	00	20			
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	335	04CA	00	23	00	17			
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	337	0404	00	32	00	18			
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	339	0.00	00	1 F	00	14			
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	341	04EA	00	2 E	00	15			
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	343	04FA	80	01	00	00			
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	345	•	00	0E	00	OF			
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	347	050A	00	38	20	38			
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	349	051A	00	00	00	00			
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	351		20	38	00	30			
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	353	0524	00					÷.	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	355	0534					00	43	
360         20         47         00         48         20         48           361         00         49         20         49         362         055A         20         02         20         03         20         04           363         20         05         20         02         20         03         20         04           363         20         05         20         06         20         07           364         20         08         20         09         365         056A         20         0A         20         0B         20         0C           366         056A         20         0A         20         0B         20         0C           366         057A         00         4E         00         4F         368         057A         00         50         05         10         52           369         00         53         00         54         00         57         00         56         00         57								44	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	359	0544			00	48	20		
364         20         08         20         09           365         056A         20         0A         20         0B         20         0C           366         056A         20         0A         20         0B         20         0C           366         056A         20         0D         4C         00         4D           367         00         4E         00         4F         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36         36	361	055A	20	02	20	03	20		
365         056A         20         0A         20         0B         20         0C           366         20         00         04         20         04         00         4D           367         00         4E         00         4F         368         057A         00         50         00         51         00         52           369         00         53         00         54         00         55           370         00         56         00         57         00         57			20	08	20	09	1		
368         057A         00         50         00         51         00         52           369         00         53         00         54         00         55           370         00         56         00         57	365	056A	20	00	00	40	00		
370 00 56 00 57	368	057A	00	50	00	51	00		
371 058A 00 00 00 00 00 00	370		00	56	00	57			
	371	058A	00	00	00	00	00	00	

	140	4,04	+1,202		
	149			150	
372	00 00 00 00 00 00	C			
373	00 00 00 00	c			; DD-D7
374	059A DD 00 00 00 00 00	C DB	0, 0,0, 0,0, 0,	,0, 0,0, 0,0, 0,0, 0,0, 0	, 55-57
375	00 00 00 00 00 00	C .			
376 377	00 00 00 00 00 00 0588 00 00 00 00 00 00	с рв	0. 0.0. 0.0. 0	,0, 0,0, 0,0, 0,0, 0,0, 0	; D8-DF
378	00 00 00 00 00 00	c 55	•••••••		
379	00 00 00 00	Ċ			
380	058A 00 00 00 00 00 00	C DB	0, 0,0, 0,0, 0	,0, D,0, 0,D, 0,0, 0,D, D	; E0-E7
381	00 00 00 00 00 00	c			
382	00 00 00 00	C		.C, 0.D, D,D, 0,0, 0,0, 0	; E6-EF
363	05CA 00 00 00 00 00 00	C DB	<b>0</b> , 0, <b>0</b> , 0, 0, 0,	,0, 0,0, 0,0, 0,0, 0,0,	
384	00 00 00 00 00 00	c c			
385	00 00 00 00 05da 00 00 00 00 00 00	C DB	0. 0.0. 0.0. 0	,0, 0,0, 0,0, 0,0, 0,0, 0	; FD-F7
386 387		c Lo			
388	00 00 00 00 00	č			; F8-FF
389	05EA 00 00 00 00 00 00	C DB	0, 0,0, 0,0, 0	.0. 0.0, 0,0, 0,0, 0,0, 0	,
390	00 00 00 00 00 00	C			
391	00 00 00 00	C C			
392		C C			
393 394		C TNC1UDE	USEAPL.ASH	APL MODIFICA	TIONS
395		C SUBTTL	Modifications	For APL Keyboard	
396		C			
397		C		" 我关系我不能没有我们的 <u>你</u> 就是我们不能没有你就能能能能是我们的。	<b>2.11 12 12 13 14 14 14 14 14 14 14 14 14 14 14 14 14 </b>
398			******	***************************************	*
399				to change Typewriter keyboar	d into 🕷
400			keyboard	to change Typen the may	*
401 402					*
403		C ;****************	*******	***	****
404		с			
405	OSFA	C APL LABEL >	IORD		
406		C	·	ble changes	
407			d translation ta		
408	05FA 05 02 00 00	C DB	NA , 2, 0, 0	; Remove PF1-12 from keys 2	-13
409 410	05FA 05 02 00 00 05FE 05 03 00 00	C DB	NA , 3, 0, 0	•	
411	0602 05 04 00 00	C DB	NA , 4, 0, 0		
412	0606 05 05 00 00	C DB	NA , 5, 0, 0		
413	060A 05 06 00 00	C DB	NA , 6, 0, 0		
414	060E 05 07 00 00	C DB	NA , 7, 0, 0		
415	0612 05 08 00 00	C DB	NA , 8, 0, 0 NA , 9, 0, 0		
416	0616 05 09 00 00	C DB C DB	NA ,10, 0, 0		
417 418	061A 05 0A 00 00 061E 05 0B 00 00	C DB	NA ,11, 0, 0		
419	0622 05 0C 00 00	C DB	NA ,12, 0, 0		
420	0626 05 0D 00 00	C DB	NA ,13, 0, 0		
421		С		1 Put PF1-12 on outboard ke	evnad
422	062A 01 4C 00 3B	C DB	NL ,76,PL,59		.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
423	062E 02 4C 00 3B	C DB C DB	NU ,76,PL,59 NL ,77,PL,60	·	
424	0632 01 40 00 3C 0636 02 40 00 3C	C DB	NU ,77,PL,60		
425 426	0636 02 4D 00 3C 063A 01 4E 00 3D	C DB	NL ,78,PL,61		
427	063E 02 4E 00 3D	C DB	NU ,78,PL,61		
428	0642 01 4F 00 3E	C DB	NL ,79,PL,62		
429	0646 02 4F 00 3E	C DB	NU ,79,PL,62		
430	064A 01 50 00 3F	C DB	NL ,80,PL,63 NU ,80,PL,63	-	
431	064E 02 50 00 3F 0652 01 51 00 40	C DB C DB	NL .81,PL,64		
432 433	0652 01 51 00 40 0656 02 51 00 40	C DB	NU ,81,PL,64		
434	065A 01 52 00 41	C DB	NL ,82,PL,65		
435	065E 02 52 00 41	C DB	NU ,82,PL,65		
436	0662 01 53 00 42	C DB	NL ,83,PL,66		
437	0666 02 53 00 42	C DB C DB	NU ,83,PL,66 NL ,84,PL,67		
438	0668° 01 54 00 43 0668° 02 54 00 43	C DB C DB	NU ,84,PL,67		
439 440	066E 02 54 00 43 0672 01 55 00 44	C DB	NL ,85,PL,68		
441	0676 02 55 00 44	C DB	NU ,65,PL,68	•	
442	067A 01 56 80 3B	C DB	NL ,86,PU,59		
443	067E 02 56 80 3B	C DB	NU ,86,PU,59		
444	0682 01 57 80 3C	C DB	NL ,87,PU,60 NU ,87,PU,60		
445	0686 02 57 80 3C	C DB C	NU 10/11/0100		
446	06 84 00 00 00 00	с рв	0, 0, 0, 0	; End of inbound table cha	nges
447 448	058A 00 00 00 00	с			
449		C ; Outbou	md translation	table changes	
450		с		A NEW ARL ON COTE to ALT &	ev 14
451	068E 8F 20 0E	C DB	08FH, A, 14	; Hove APL ON/OFF to ALT k	
452		С С DB	0B0H,L,76	Move PF1-12 to outboard	keypad
453	0691 B0 D0 4C	C DB C DB	0B1H,L,77	,	
454	0694 B1 00 4D 0697 B2 0D 4E	C DB	0B2H,L,78		
455 456	0694 B3 00 4F	C DB	0B3H, L, 79		
457	069D B4 00 50	с ов	054H, L, 80		
458	06AD B5 00 51	C DB	085H,L,81		
459	06A3 B6 D0 52	C DB	086H,1,82		
460	06A6 B7 00 53	C DB	087H, L, 83		
461	06A9 BB 00 54	C DB C DB	088H,L,84 089H,L,85		
462	06AC B9 00 55 06AF BA D0 56	C DB	OBAH, L, 86		
° 463 464	06AF BA 00 56 06B2 BB 00 57	C DB	0BBH,L,87		
465	0685 BC 00 00	C DB	0BCH,0, 0	; Remove PF13-F24	
466	06B8 BD 00 00	C DB	OBDH,0, D		
467	06BB BE 00 00	C DB		•	
468	06BE BF 00 00	C DB	0BFH,0, 0		

4,641,262

1	5	1
_	~	-

.

469 470									102
					-	0B	OCOH.0, 0		
470		CO 00							
470	06C4	C1 00			2	DB	0C1H,0, 0		
471	D6C7	CZ 00	00		C	DB	OC2H,0, O		
472	DECA	C3 D0	00	1	C	DB	0C3H,0, 0		
473		C4 00			C	DB	0C4H,0, 0		
		C5 00			č	DB	0C5H,0, D		
474					с С	DB	0C6H,0, 0		
475		C6 00							
476	06D6	C7 00	00		C	DB	0C7H,0, 0		
477					С				ind of outbound table changes
478	0609	00 D0	00		С	DB	000H,C. 0	; E	nd of outbound table changes
479					C				
					c				
480						THEFT	USETXT.ASH		I TEXT MODIFICATIONS
481					C	INCLODE	USEINI ASI	F	
482					C	SUBTTL	Modifications	FOR I	lext Keyboard
483					С				
							•		
484					c				· 张聪能能能能够多少能能力,并有多有效要求完全就要要要要要要做做这些做法的。
485						*****	******	*****	***************************************
486					C 1*				
487					C ;*	Hodific	ations required	to c	change Typewriter keyboard into
488					C ;*		keyboard		
489									and the transformed for
490					C ;*	NOTE: T	ne changes for	The A	APL keyboard are also required for
491					C ;*	t	he Text keyboard	d	
492					C 1*				
493					C ;*****	****	*****	****	不在就要就就来来就我这些我就是我们都能能能不要要要要要我们能能能能能。
494					c				
							000		
495	06DC				C TEXT	LABEL W			
496					С				
497					С;	Inbound	translation ta	ble (	changes
498					C				
499	06DC	01 1A	An 4		č	DB	NL ,26,PU, 2	: 1	Put ! on lower case key 26
							NU ,26,PL,26		Put I on upper case key 26
500	06EO	02 1A	00 1		c	DB	110 1601FL120	• •	a un apper and may an
501					С				
502	06E4	05 IC	00 0	00	С	DB	NA ,28, 0, 0	; 1	Move HOME to ALT/DELETE
503	06E8				с	0B	NA ,69,PL,71		
504	0010	•5 45			č				
							WI // D CTI	VEV	; Move PA1 to lower case of DUP key
505	OGEC	01 42	00 1		C	DB	NL ,00, 0,011_	NET :	I HOVE PAT to Your sare a s
506					C				
507	06F0	03 10	00 0	00	C	DB	NPC,28, 0, 0	; ł	Move MODE SWITCH to ALT/PA1
538	06F4	05 42	00 F	°5	C	DB	NA ,66, 0,SWIT	ICH_MO	ODE
509					c			-	
-					č	OB	NI 47. 0.41T	KEY	; Move PA2 to lower case of FM key
510	06F8	01 43							,
511	06FC	05 43	00 0		С	DB	NA ,67, 0, 0		
512					С				
513	0700	05 44	00 1	IC	С	DB	NA ,68,PL,28	; I	Move RETURN to ALT/INSERT
514					С				
515	0704	01 3B	40 4	17	с	DB	NL ,59,PC,71	; ;	Swap CURSR SEL and CLEAR
516	0708	02 3B			c	DB	NU ,59,PC,71		
					č	DB	NA ,59,PL,70		
517	070C	05 3B				60	12/11/11		
518					C				End of inbound table changes
519	0710	00 00	00 0		c	DB	0, 0, 0, 0		End of Indound table changes
520					С				
521					C ;	Outbour	d translation t	table	changes
F 2 2					С				
266			1.4		С	DB	021H, L, 26	; 1	Hove ! to lower case key 26
522 523	0714	21 00			С	DB	0814,1,26	;	Move Not Sign to upper case 26
523	0714		1.4		č	DB	082H,A,7		Move Cent Sign to ALT/key 7
523 524	0717	81 80				00			hove dent organice mentally a
523 524 525	0717						00203437		
523 524	0717	81 80			C				
523 524 525	0717	81 80	07		C	DÐ	008H,0, 0		Remove Backspace
523 524 525 526 527	0717 071A 071D	81 80 82 20 08 00	07			DB DB			Remove Backspace
523 524 525 526 527 528	0717 071A 071D	81 80 82 20	07		c c		008H,0, 0	;	
523 524 525 526 527 528 529	0717 071A 071D 0720	61 80 62 20 08 00 63 00	07 00 00		с с с	DB	008H,0, 0 083H,0, 0	;	
523 524 525 526 527 528 529 529 530	0717 071A 071D 0720	81 80 82 20 08 00	07 00 00		с с с с		008H,0, 0	;	Remove Backspace Move TEXT DN/DFF to lower case key 14
523 524 525 526 527 528 529 530 531	0717 071A 071D 0720 0723	81 80 82 20 08 00 83 00 8F 00	07 00 00 00		с с с с с	DB DB	008H,0, 0 083H,0, 0 08FH,L,14	;	Move TEXT ON/DFF to lower case key 14
523 524 525 526 527 528 529 529 530	0717 071A 071D 0720	61 80 62 20 08 00 63 00 8F 00 0D 20	07 00 00 00 00 0E		с с с с с с с с с с	DB DB DB	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,f8	;	Move TEXT ON/OFF to lower case key 14 Move RETURN to ALT/INSERT
523 524 525 526 527 528 529 530 531	0717 071A 071D 0720 0723	81 80 82 20 08 00 83 00 8F 00 0D 20	07 00 00 00 00 0E		с с с с с с с с с с с с с	DB DB	008H,0, 0 083H,0, 0 08FH,L,14	;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE
523 524 525 526 527 528 529 530 531 532 533	0717 071A 071D 0720 0723 0726	81 80 82 20 08 00 83 00 8F 00 0D 20	07 00 00 00 00 0E 04 44 045		с с с с с с с с с с	DB DB DB	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,f8	;   ;   ;   ;	Move TEXT ON/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/Key 59
523 524 525 526 527 528 529 530 531 533 533 533	0717 071A 071D 0720 0723 0726 0729 0720	61 80 62 20 63 00 63 00 8F 00 6D 20 87 20 92 20	07 00 00 00 00 00 00 00 00 00 00 00 00 0			DB DB DB DB	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,68 067H,A,69	; ; ; ;	Move TEXT ON/OFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 59
523 524 525 526 527 528 529 530 531 532 533 533 533 533	0717 071A 071D 0720 0723 0726 0729 072C 072F	61 80 62 20 63 00 63 00 8F 00 6D 20 87 20 92 20 93 00	07 00 00 00 00 00 00 00 00 00 00 00 00 0			DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,68 087H,A,69 092H,A,59 093H,L,59	; ; ; ;	Move TEXT ON/OFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 59
523 524 525 527 528 529 530 531 532 533 533 533 535 536	0717 071A 071D 0720 0723 0726 0729 0726 0727 0727	61 80 62 20 68 00 63 00 8F 00 8F 00 87 20 92 20 93 00 A0 80	07 00 00 00 00 00 00 00 00 00 00 00 00 0			DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,(8 087H,A,(9 092H,A,19 093H,L,19 0A0H,U,16	;   ;   ; : ; : ; :	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 59 Move DUP to upper case key 66
523 524 526 527 528 530 531 533 533 533 533 533 535 535 535	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0727 0732 0735	81 80 82 20 08 00 83 00 8F 00 0D 20 87 20 92 20 93 00 A0 80 A1 00	07 00 00 00 00 00 00 00 00 00 00 00 00 0			DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,(8 067H,A,(9 092H,A,19 092H,L,19 0A0H,U,16 0A1H,L,16	;	Move TEXT ON/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CLERSR SEL to ALT/key 59 Move CLEAR to lower case key 59 Move DUP to upper case key 66 Move PAI to lower case key 66
523 524 525 527 528 529 530 531 532 533 533 533 535 536	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0727 0732 0735	61 80 62 20 68 00 63 00 8F 00 8F 00 87 20 92 20 93 00 A0 80	07 00 00 00 00 00 00 00 00 00 00 00 00 0			DB DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,68 087H,A,69 092H,A,59 092H,A,59 093H,L,59 040H,U,66 042H,U,67	;   ; ; ; ; ; ; ;	Move TEXT ON/DFF to lower case key 14 Move RETURN to ALT/INSERT Move CURSR SEL to ALT/Key 59 Move CLEAR to lower case key 59 Move DUP to upper case key 66 Move PA1 to lower case key 66 Move Field Mark to upper case key 67
523 524 525 526 527 528 530 531 532 533 534 535 536 536 538	0717 071A 071D 0720 0723 0726 0727 0727 0727 0727 0735 0738	81 80 82 20 08 00 83 00 8F 00 0D 20 87 20 92 20 93 00 A0 80 A1 00	<ul> <li>07</li> <li>00</li> &lt;</ul>			DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,(8 067H,A,(9 092H,A,19 092H,L,19 0A0H,U,16 0A1H,L,16	;   ; ; ; ; ; ; ;	Move TEXT ON/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CLERSR SEL to ALT/key 59 Move CLEAR to lower case key 59 Move DUP to upper case key 66 Move PAI to lower case key 66
523 524 525 527 528 529 530 531 532 533 534 533 536 537 538 539	0717 071A 071D 0720 0723 0726 0727 0727 0727 0727 0735 0738	81       80         82       20         08       00         83       00         8F       00         87       20         93       00         A0       80         A1       00         A2       80	<ul> <li>07</li> <li>00</li> &lt;</ul>			DB DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,68 087H,A,69 092H,A,59 092H,A,59 093H,L,59 040H,U,66 042H,U,67	;   ;   ; - ; - ; - ; - ; - ; - ; - ; - ; - ; -	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 59 Hove DUP to upper case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67
523 524 526 527 528 530 531 533 533 533 533 533 533 536 538 538 538 538 538	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0725 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 042 042 042 042 042			DB DB DB DB DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,(8 067H,A,(9 092H,A,59 049H,U,9 040H,U,9 040H,U,9 040H,U,6 042H,U,67 043H,L,67	;   ;   ; - ; - ; - ; - ; - ; - ; - ; - ; - ; -	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 59 Hove DUP to upper case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67
523 524 525 526 529 530 531 532 533 533 535 536 538 538 538 538 539 541	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0725 0732 0735 0738 0738	81       80         82       20         08       00         83       00         8F       00         87       20         93       00         A0       80         A1       00         A2       80	07 000 000 005 044 045 038 042 042 042 042 042			DB DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,68 087H,A,69 092H,A,59 092H,A,59 093H,L,59 040H,U,66 042H,U,67	;   ;   ; - ; - ; - ; - ; - ; - ; - ; - ; - ; -	Move TEXT ON/DFF to lower case key 14 Move RETURN to ALT/INSERT Move CURSR SEL to ALT/Key 59 Move CLEAR to lower case key 59 Move DUP to upper case key 66 Move PA1 to lower case key 66 Move Field Mark to upper case key 67
523 524 525 527 529 530 531 533 533 533 533 536 537 539 540 542	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0725 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 042 042 042 042 042			DB DB DB DB DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,(8 067H,A,(9 092H,A,59 049H,U,9 040H,U,9 040H,U,9 040H,U,6 042H,U,67 043H,L,67	;   ;   ; - ; - ; - ; - ; - ; - ; - ; - ; - ; -	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 59 Hove DUP to upper case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67
523 524 525 526 529 530 531 532 533 533 535 536 538 538 538 538 539 541	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0725 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 042 042 042 042 042			DB DB DB DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,(8 087H,A,69 092H,A,19 093H,L,19 0A0H,U,16 0A1H,L,06 0A2H,U,67 0A3H,L,07 0A3H,L,07	;   ;   ; - ; - ; - ; - ; - ; - ; - ; - ; - ; -	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSE SEL to ALT/key 59 Move CLEAR to lower case key 59 Move DP to upper case key 66 Move FA1 to lower case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes
523 524 526 527 528 530 531 5334 5336 5337 538 537 538 537 538 540 541 543	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0725 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 042 042 042 042 042			DB DB DB DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,f8 087H,A,f9 092H,A,59 093H,L,59 040H,U,76 042H,U,67 043H,L,67 043H,L,67 043H,L,67	; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT ON/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move OUP to upper case key 66 Move Field Mark to upper case key 67 Move Field Mark to upper case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS
523 524 525 526 527 529 530 532 533 533 533 533 534 538 538 538 538 538 538 538 538 538 538	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0725 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 042 042 042 042 042			DB DB DB DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,f8 087H,A,f9 092H,A,59 093H,L,59 040H,U,76 042H,U,67 043H,L,67 043H,L,67 043H,L,67	; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT ON/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move OUP to upper case key 66 Move Field Mark to upper case key 67 Move Field Mark to upper case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS
523 524 5225 529 530 532 533 5335 5335 5339 540 542 544 544 544 545	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0725 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 042 042 042 042 042			DB DB DB DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 08FH,L,14 00DH,A,f8 087H,A,f9 092H,A,59 093H,L,59 040H,U,76 042H,U,67 043H,L,67 043H,L,67 043H,L,67	; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSE SEL to ALT/key 59 Move CLEAR to lower case key 59 Move DP to upper case key 66 Move FA1 to lower case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes
523 524 526 527 529 530 531 533 533 533 533 533 533 533 533 534 542 543 544 544 544 544 544	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 042 042 042 042 042			DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0,0 083H,0,0 08FH,L,14 00DH,A,(8 057H,A,(9 092H,A,19 092H,L,19 040H,U,10 040H,U,10 042H,U,07 043H,L,07 043H,L,07 043H,L,07 043H,L,07	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move UMER to ALT/DELETE Move CLEAR to lower case key 59 Move DUP to upper case key 66 Move PA1 to lower case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard
523 524 5225 529 530 532 533 5335 5335 5339 540 542 544 544 544 545	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 042 042 042 042 042			DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0,0 083H,0,0 08FH,L,14 00DH,A,(8 057H,A,(9 092H,A,19 092H,L,19 040H,U,10 040H,U,10 042H,U,07 043H,L,07 043H,L,07 043H,L,07 043H,L,07	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move UNER to ALT/INSERT Move CLEAR to lower case key 59 Move DUP to upper case key 66 Move PA1 to lower case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard
523 524 526 526 528 531 5334 5336 538 538 541 243 5445 5445 5445 5445 5447	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043			DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0,0 083H,0,0 08FH,L,14 00DH,A,(8 057H,A,(9 092H,A,19 092H,L,19 040H,U,10 040H,U,10 042H,U,07 043H,L,07 043H,L,07 043H,L,07 043H,L,07	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT ON/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move OUP to upper case key 66 Move Field Mark to upper case key 67 Move Field Mark to upper case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS
523 5245 5267529 53015334553789 5334553789 5412344554345545445 544554544554544554544554545545545545545	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043			DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0,0 083H,0,0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 043H,L,59 0A1H,L,56 0A1H,L,66 0A1H,L,66 0A3H,L,57 000H,0,0	; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move DUP to upper case key 69 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard
523 524 5226 5226 5229 5332 5334 5339 540 1 5445 5445 5445 5445 5445 5445 544	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0,0 083H,0,0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 043H,L,59 0A1H,L,56 0A1H,L,66 0A1H,L,66 0A3H,L,57 000H,0,0	; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move DUP to upper case key 69 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard
523 5245 5267529 53015334553789 5334553789 5412344554345545445 544554544554544554544554545545545545545	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0,0 083H,0,0 087H,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 093H,L,59 004H,U,66 0A1H,L,66 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,68 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7 0A3H,C7	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move DUP to upper case key 59 Move Field Mark to upper case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
523 524 5226 5226 5229 5332 5334 5339 540 1 5445 5445 5445 5445 5445 5445 544	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0,0 083H,0,0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 043H,L,59 0A1H,L,56 0A1H,L,66 0A1H,L,66 0A3H,L,57 000H,0,0	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move DUP to upper case key 59 Move Field Mark to upper case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
523 5245 5226 5227 5290 5312 53355378 53355378 5412 5445 5446 5449 5449 5449 551	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 093H,L,59 0A1H,L,66 0A1H,L,66 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move DUP to upper case key 66 Move PA1 to lower case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
523 524 5226 5226 5229 5332 5334 5339 5339 5401 5445 5445 5445 5445 5445 5445 5445	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 093H,L,59 0A1H,L,66 0A1H,L,66 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move DUP to upper case key 66 Move PA1 to lower case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
524 5226 5226 5226 5228 5332 5334 5336 5339 5412 5444 5448 5444 5448 551 5553 5553 5553 5553 5553 5553 5553	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 093H,L,59 0A1H,L,66 0A1H,L,66 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L,68 0A3H,L	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move DUP to upper case key 59 Move Field Mark to upper case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
524 5225 5226 5226 5229 5333 5335 5339 5441 5445 5449 551 552 5554	0717 071A 071D 0720 0723 0726 0727 0725 0735 0738 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB TNCLUDE SUBTTL	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 0A0H,U,66 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move DUP to upper case key 66 Move PA1 to lower case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
524 5226 5226 5226 5228 5332 5334 5336 5339 5412 5444 5448 5444 5448 551 5553 5553 5553 5553 5553 5553 5553	0717 071A 071D 0720 0723 0726 0729 0726 0729 0726 0732 0735 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 0A0H,U,66 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move HOME to ALT/DELETE Move CURSR SEL to ALT/key 59 Move DUP to upper case key 66 Move PA1 to lower case key 66 Move Field Mark to upper case key 67 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
524 5225 5226 5226 5229 5333 5335 5339 5441 5445 5449 551 552 5554	0717 071A 071D 0720 0723 0726 0727 0725 0735 0738 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 043H,L,66 0A1H,L,66 0A1H,L,66 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,68 0A2H,0,67 0A3H,L,68 0A2H,0,67 0A3H,L,68 0A2H,0,67 0A3H,L,68 0A2H,0,70 0A3H,L,68 0A2H,0,70 0A3H,L,78 0A3H,0,70 0A3H,L,78 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move RURSR SEL to ALT/key 59 Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 66 Move PA1 to lower case key 66 Move Field Mark to upper case key 67 Nove PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
524 5226 5226 5226 5228 5333 5336 5339 544 5445 554 555 555 555 555 555 555 5	0717 071A 071D 0720 0723 0726 0727 0725 0735 0738 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 0A0H,U,66 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L	;   ;   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move RURSR SEL to ALT/key 59 Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 66 Move PA1 to lower case key 66 Move Field Mark to upper case key 67 Nove PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
5245 5225 5226 5226 5333 5333 55338 5339 5442 5445 5512 555 555 555 555 555 555 555 555 5	0717 071A 071D 0720 0723 0726 0727 0725 0735 0738 0738 0738	81       80         82       20         08       00         83       00         87       20         93       00         40       80         A0       80         A1       00         A3       00	07 000 000 005 044 045 038 038 042 042 042 042 043		C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0, 0 083H,0, 0 08FH,L,14 00DH,A,68 057H,A,69 092H,A,59 093H,L,59 043H,L,66 0A1H,L,66 0A1H,L,66 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,68 0A2H,0,67 0A3H,L,68 0A2H,0,67 0A3H,L,68 0A2H,0,67 0A3H,L,68 0A2H,0,70 0A3H,L,68 0A2H,0,70 0A3H,L,78 0A3H,0,70 0A3H,L,78 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0,70 0A3H,0	<pre>;   ;   ;   ;   ;   ;   ;   ;   ;   ;  </pre>	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 59 Move DP1 to upper case key 66 Move PA1 to lower case key 66 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
52555555555555555555555555555555555555	0717 071A 071D 0720 0723 0726 0727 0725 0732 0735 0738 0738 0738	61 80 82 20 08 00 83 00 87 00 92 20 93 00 87 20 93 00 87 20 93 00 87 20 93 00 87 20 93 00 80 80 A1 00 A3 00 00 00	0 07 0 00 0 02 0 44 0 38 0 45 0 38 0 42 1 43 1 43 1 00		CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	DB DB DB DB DB DB DB DB DB DB DB DB DB D	008H,0,0 083H,0,0 085H,L,14 00DH,A,18 085H,A,19 092H,A,19 093H,L,19 0A0H,U,16 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,77 0A3H,L,7	<pre>;   ;   ;   ;   ;   ;   ;   ;   ;   ;  </pre>	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 59 Move DP1 to upper case key 66 Move PA1 to lower case key 66 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
52555267 52257 52267 52267 52267 5333455 53390 1233455 53445 555555555555555555555555555	0717 071A 071D 0720 0723 0726 0727 0725 0732 0735 0738 0738 0738 0738 0738	61 80 82 20 08 00 83 00 87 00 92 20 93 00 A0 80 A1 00 A2 80 A3 00 00 00	C C0	00	C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB TNCLUDE SUBTTL ***********************************	008H,0,0 083H,0,0 085H,L,14 00DH,A,(8 057H,A,(9 093H,L,59 093H,L,59 040H,U,76 042H,U,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,67 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 043H,L,77 047 047 04704 047 047 047 047 047 047 047 047 047 0	<pre>;   ;   ;   ;   ;   ;   ;   ;   ;   ;  </pre>	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move RETURN to ALT/INSERT Move CURSR SEL to ALT/Key 59 Move CUEAR to lower case key 59 Move DP1 to upper case key 66 Move PA1 to lower case key 66 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard eyboard ************************************
5245678901234567890123456789012345575555555555555555555555555555555555	0717 071A 071D 0720 0723 0726 0729 0720 0727 0735 0738 0738 0738 0738 0738 0738	61 80 62 20 63 00 63 00 67 20 92 20 92 20 92 20 92 20 93 00 A1 00 A2 80 A3 00 00	C 000	00 00	C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB TNCLUDE SUBTTL ***********************************	008H,0,0 083H,0,0 085H,1,0 085H,1,14 00DH,4,68 057H,4,69 092H,4,59 093H,1,59 093H,1,59 093H,1,59 003H,1,59 004H,0,67 003H,1,57 000H,0,0 CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications	<pre>;   ;   ;   ;   ;   ;   ;   ;   ;   ;  </pre>	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move CURSR SEL to ALT/key 59 Move CLEAR to lower case key 59 Move DP1 to upper case key 66 Move PA1 to lower case key 66 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard change Typewriter or APL keyboard eyboard transformer to APL keyboard transformer to
52555267 52257 52267 52267 52267 5333455 53390 1233455 53445 555555555555555555555555555	0717 071A 071D 0720 0723 0726 0727 0725 0732 0735 0738 0738 0738 0738 0738	61 80 62 20 63 00 63 00 67 20 92 20 92 20 92 20 92 20 93 00 A1 00 A2 80 A3 00 00	C C0	00 00	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	DB DB DB DB DB DB DB DB DB DB	008H,0,0 083H,0,0 085H,L,14 00DH,A,68 087H,A,69 092H,A,19 093H,L,19 0A0H,U,16 0A2H,U,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,67 0A3H,L,6	<pre>;   ;   ;   ;   ;   ;   ;   ;   ;   ;  </pre>	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move RETURN to ALT/INSERT Move CURSR SEL to ALT/Key 59 Move CLEAR to lower case key 59 Move PA1 to lower case key 66 Move PA1 to lower case key 66 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************
5245678901234567890123456789012345575555555555555555555555555555555555	0717 071A 071D 0720 0723 0726 0729 0720 0727 0735 0738 0738 0738 0738 0738 0738	61 80 82 20 08 00 83 00 87 20 92 20 93 00 87 20 93 00 80 80 80 80 80 80 90 80 90 80 90 90 00 80 90 90 00 80 90 90 00 80 90 90 00 80 80 80 80 80 80 80 80 80 80 80 80 8	C 000	00 00 00	C C C C C C C C C C C C C C C C C C C	DB DB DB DB DB DB DB DB DB DB DB DB DB TNCLUDE SUBTTL ***********************************	008H,0,0 083H,0,0 085H,1,0 085H,1,14 00DH,4,68 057H,4,69 092H,4,59 093H,1,59 093H,1,59 093H,1,59 003H,1,59 004H,0,67 003H,1,57 000H,0,0 CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications CUSEAS.45M Modifications	<pre>;   ;   ;   ;   ;   ;   ;   ;   ;   ;  </pre>	Move TEXT DN/DFF to lower case key 14 Move RETURN to ALT/INSERT Move RETURN to ALT/INSERT Move CURSR SEL to ALT/Key 59 Move CLEAR to lower case key 59 Move PA1 to lower case key 66 Move PA1 to lower case key 66 Move PA2 to lower case key 67 End of outbound table changes ; ATTRIBUTE SELECT MODS Attribute Select keyboard ************************************

4,641,262

154

			10.	134	
563	0751	03 E0	~~ ~~	C DB NU,80,0,0	
564	0755	02 50   02 51		C DB NU ,81, 0, 0	
565	0759	02 52		C DB NU,82,0,0	
566	0750	02 53		C DB NU, 83, 0, 0	
567	0761	02 54		C DB NU ,84, 0, 0	
568	0765	02 55		C DB NU ,85, 0, 0	
569	0769	02 56		C DB NU ,86, 0, 0	
570	076D	02 57		C DB NU,87,0,0	
571				C	
572	0771	00 00	00 00	C DB 0,0,0,0 ; End of inbound table change	es
573				C	
574				C ; Outbound translation table changes	
575				~	1
576	0775	D0 80	4C	C DB 0D0H,U,76 ; Put attributes on outboard	key pao
577	0778	D1 80	4F	C DB 0D1H,U,79	
578	077B	D2 80	52	C DB 0D2H,U,82	
579	077E	D3 80	55	C DB 003H,U,85	
580	0781	D4 20	4C	C DB 004H, A, 76	
581	0784	D5 20		C DB 005H,A,79	
582	0787	D6 20		C DB 0D6H,A,82	
583	078A	D7 20	55	C DB OD7H,A,85	
584				C C DB DE0H,U,77 ; Put PS selection on outboa	rd key pad
585	078D	ED 80		- · · · · · · · · · · · · · · · · · · ·	•••
586		E1 80		C DB 0E1H,U,80 C DB 0E2H,U,83	
587		E2 80		C DB 0E2H,U,B3 C DB 0E3H,U,86	
588	.0796	E3 80 E4 20		C DB 0E4H,A,77	
589 590	0799 079C	E4 20 E5 20		C DB 0E5H, A, 80	
591		E5 20		C DB 0E6H,A,63	
592		E7 20		C DB 0E7H, A, 86	
593	07AL			r	
594	0745	F0 80	4E	C CB OFOH, U, 78 ; Put colors on outboard key	r pad
595	0748	F1 80		C CB 0F1H,U,81	
596		F2 80		C EB 0F2H,U,84	
597		F3 80		C 0F3H,U,87	
598	07B1	F4 20		C 11B 0F4H, A, 7B	
599	07B4	F5 20	51	C DB 0F5H,A,81	
600	07B7	F6 20	54	C 1)B 0F6H,A,84	
601	07BA	F7 20	57	C DB 0F7H,A,87	
602				C c c c c c c c c c c c c c c c c c c c	
603	07BD	00 DO	00	C DB 000H,0, 0 ; End of outbound table char	iges
604				· C	
605				C THELLIDE LISER1038.ASM } RPQ 8K1038 M	DOTETCATIONS
606				C INCLUDE USER1038.ASM ; RPG BK1038 R	
607				C SUBTL Modifications For RPQ EKINIA Keyboard	
608				C · ·	
609				C 。 3 金旗家族生素大学家个女子有些有些不是有些有些不是有些不是有些不是有些不是是不是是不是是不是是是有些有些不是是是是有些不	*****
610					
611				C ;* Modifications required to change Typewriter keyboard	into
612				C :* an RPQ 8K1038 keyboard	
613 614				<b>—</b>	
615				C \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	朱永永安家永太安安-
616				C , and a second	
617	0700			C REKID38 LABEL WORD	
618				c	
619				C ; Inbound translation table changes	
620				-	_
621	0700			C DB NA, 2, 0,0 } Remove F1-12 from keys 2-1	
622	0,00	05 02	00 00		3
623				C DB NA, 3, 0,0	3
	07C4 07C8	05 03 05 04	00 00 00 00	C DB NA, 4, 0,0	3
624	07C4	05 03 05 04 05 05	00 00 00 00 00 00	C DB NA, 4, 0,0 C DB NA, 5, 0,0	3
	07C4 07C8	05 03 05 04 05 05 05 06	00 00 00 00 00 00 00 00	C DB NA, 4, 0,0 C DB NA, 5, 0,0 C DB NA, 6, 0,0	3
624	07C4 07C8 07CC	05 03 05 04 05 05 05 06 05 07	00 00 00 00 00 00 00 00 00 00	C DB NA, 4, 0,0 C DB NA, 5, 0,0 C DB NA, 6, 0,0 C DB NA, 6, 0,0 C DB NA, 7, 0,0	3
624 625 626 627	07C4 07C8 07CC 07D0 07D4 07D8	05 03 05 04 05 05 05 06 05 07 05 08	00 00 00 00 00 00 00 00 00 00 00 00	C DB NA, 4, 0,0 C DB NA, 5, 0,0 C DB NA, 6, 0,0 C DB NA, 7, 0,0 C DB NA, 7, 0,0	3
624 625 626 627 628	07C4 07C8 07CC 07D0 07D4 07D8 07D8	05 03 05 04 05 05 05 06 05 07 05 08 05 09	00 00 00 00 00 00 00 00 00 00 00 00 00 00	C     DB     NA, 4, 0,0       C     DB     NA, 5, 0,0       C     DB     NA, 6, 0,0       C     DB     NA, 7, 0,0       C     DB     NA, 8, 0,0       C     DB     NA, 9, 0,0	3
624 625 626 627 628 629	07C4 07C8 07CC 07D0 07D4 07D8 07D8 07DC 07E0	05 03 05 04 05 05 05 06 05 07 05 08 05 09 05 0A	00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00	C     DB     NA, 4, 0.0       C     DB     NA, 5, 0.0       C     DB     NA, 6, 0.0       C     DB     NA, 7, 0.0       C     DB     NA, 7, 0.0       C     DB     NA, 9, 0.0       C     DB     NA, 9, 0.0       C     DB     NA, 10, 0.0	3
624 625 626 627 628 629 630	07C4 07C8 07CC 07D0 07D4 07D8 07D8 07DC 07E0 07E0	05 03 05 04 05 05 05 06 05 07 05 08 05 09 05 0A 05 0B	00       80         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00	C     DB     NA, 4, 0,0       C     DB     NA, 5, 0,0       C     DB     NA, 5, 0,0       C     DB     NA, 7, 0,0       C     DB     NA, 7, 0,0       C     DB     NA, 8, 0,0       C     DB     NA, 9, 0,0       C     DB     NA, 10, 0,0       C     DB     NA, 110, 0,0	3
624 625 626 627 628 629 630 631	07C4 07C8 07CC 07D0 07D4 07D8 07DC 07E0 07E4 07E8	05 03 05 04 05 05 05 06 05 07 05 08 05 09 05 0A 05 0B 05 0B 05 0C	00       80         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00	C       DB       NA, 4, 0,0         C       DB       NA, 5, 0,0         C       DB       NA, 5, 0,0         C       DB       NA, 7, 0,0         C       DB       NA, 7, 0,0         C       DB       NA, 8, 0,0         C       DB       NA, 9, 0,0         C       DB       NA, 10, 0,0         C       DB       NA, 111, 0,0         C       DB       NA, 112, 0,0	3
624 625 626 627 628 629 630 631 632	07C4 07C8 07CC 07D0 07D4 07D8 07D8 07DC 07E0 07E0	05 03 05 04 05 05 05 06 05 07 05 08 05 09 05 0A 05 0B 05 0B 05 0C	00       80         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00	C       DB       NA , 4, 0.0         C       DB       NA , 5, 0.0         C       DB       NA , 6, 0.0         C       DB       NA , 7, 0.0         C       DB       NA , 7, 0.0         C       DB       NA , 9, 0.0         C       DB       NA , 9, 0.0         C       DB       NA , 10, 0.0         C       DB       NA , 11, 0.0         C       DB       NA , 12, 0.0         C       DB       NA , 13, 0,0	
624 625 626 627 628 629 630 631 632 633	07C4 07C8 07CC 07D0 07D4 07D8 07DC 07E0 07E4 07E8 07E5	05 03 05 04 05 05 05 06 05 07 05 08 05 09 05 0A 05 0A 05 0B 05 0D	00     00       00     00       00     00       00     00       00     00       00     00       00     00       00     00       00     00       00     00       00     00       00     00       00     00       00     00       00     00       00     00	C DB NA, 4, 0,0 C DB NA, 5, 0,0 C DB NA, 5, 0,0 C DB NA, 6, 0,0 C DB NA, 7, 0,0 C DB NA, 8, 0,0 C DB NA, 9, 0,0 C DB NA, 10, 0,0 C DB NA, 11, 0,0 C DB NA, 12, 0,0 C DB NA, 13, 0,0	
624 625 626 627 628 629 630 631 632 633 634	07C4 07C8 07CC 07D0 07D4 07D8 07DC 07E0 07E4 07E8 07EC 07F0	05 03 05 04 05 05 05 06 05 07 05 08 05 09 05 0A 05 0B 05 0C 05 0D 01 3B	00       80         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         40       47	C DB NA, 4, 0,0 C DB NA, 5, 0,0 C DB NA, 5, 0,0 C DB NA, 6, 0,0 C DB NA, 7, 0,0 C DB NA, 7, 0,0 C DB NA, 9, 0,0 C DB NA, 10, 0.0 C DB NA, 11, 0,0 C DB NA, 12, 0,0 C DB NA, 13, 0,0 C DB NA, 13, 0,0 C DB NA, 59,PC,71 ; Hove CLEAR to upper/lower	
624 625 626 627 628 630 631 632 633 633 634 635	07C4 07C8 07C0 07D4 07D8 07D2 07E0 07E4 07E8 07EC 07F0 07F4	05 03 05 04 05 05 05 06 05 07 05 07 05 07 05 07 05 08 05 08 05 00 05 00 05 00 05 00 01 38 02 38	00       80         00       00         80       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         40       47	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 112, 0.0 C DB NA, 13, 0.0	
624 625 626 627 628 630 631 632 633 633 633 633 633 635 636	07C4 07C8 07CC 07D0 07D4 07D8 07DC 07E0 07E4 07E8 07EC 07F0	05 03 05 04 05 05 05 06 05 07 05 07 05 07 05 07 05 08 05 08 05 00 05 00 05 00 05 00 01 38 02 38	00       80         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         00       00         40       47	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 10, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 39, 0.0 C DB NA, 59, 0, 0	case key !
624 625 626 627 629 631 632 633 634 635 635 636 637	07C4 07C6 07C0 07D0 07D4 07D8 07D0 07E0 07E0 07E0 07E0 07F0 07F4 07F8	05 03 05 04 05 05 05 05 05 07 05 08 05 07 05 08 05 00 05 00 01 38 05 38	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         00	C DB NA, 4, 0,0 C DB NA, 5, 0,0 C DB NA, 5, 0,0 C DB NA, 6, 0.0 C DB NA, 7, 0,0 C DB NA, 7, 0,0 C DB NA, 9, 0,0 C DB NA, 10, 0.0 C DB NA, 11, 0,0 C DB NA, 12, 0.0 C DB NA, 13, 0,0 C DB NA, 13, 0,0 C DB NL, 59, PC, 71 ; Hove CLEAR to upper/lower C DB NL, 59, 9C, 71 C DB NL, 59, 0, 0 C	case key !
624 625 626 627 628 630 631 633 633 633 633 633 633 633 633 633	07C4 07C8 07C6 07D0 07D0 07D0 07D0 07D0 07D0 07D0 07E0 07E	05 03 05 04 05 05 05 06 05 07 05 08 05 07 05 08 05 09 05 08 05 00 05 00 05 00 01 38 02 38 05 38 01 30	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         00           00         00	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 112, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NA, 59, 0, 0 C DB NA, 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower	case key !
624 625 626 627 628 629 630 631 632 633 634 635 635 636 637 638 638	07C4 07C8 07C0 07D0 07D4 07D8 07D0 07E0 07E0 07E0 07E0 07F0 07F4 07F8	05 03 05 04 05 05 05 06 05 07 05 08 05 07 05 08 05 09 05 08 05 00 05 00 05 00 01 38 02 38 05 38 01 30	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         00	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 11, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 C DB NL, 59, PC, 71 C DB NA, 59, 0, 0 C C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lowe	case key <u>:</u> Duer key 61
624 625 625 627 631 632 633 634 635 635 635 635 639 639 639 640	07C4 07C8 07CC 07D0 07D4 07D8 07D0 07E0 07E0 07E0 07E6 07F0 07F6 07F6	05         03           05         04           05         05           05         05           05         06           05         07           05         08           05         08           05         08           05         08           05         08           05         08           05         08           05         08           01         38           01         30           02         30	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         40           00         46           80         46	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 11, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 C DB NL, 59, PC, 71 C DB NA, 59, 0, 0 C C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lower/lowe	case key <u>:</u> Duer key 61
624 625 626 627 630 631 632 633 634 635 635 636 635 636 637 638 639 640 641	07C4 07C8 07CC 07D0 07D4 07D8 07DC 07DC 07DC 07DC 07DC 07DC 07DC 07DC	05 03 05 04 05 05 05 06 05 07 05 08 05 09 05 08 05 00 05 00 05 00 01 38 02 38 05 38 01 30 02 30 01 42	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           40         47           00         46           80         46           00         F4	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Hove CLEAR to upper/lower C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 C DB NL, 59, PC, 71 C DB NL, 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower C DB NL, 61, PL, 70 C DB NL, 66, 0, CTL_KEY ; Move PAI to lower case H	case key <u>:</u> Duer key 61
624 625 626 627 628 631 632 633 633 635 634 635 636 638 638 639 641 642	07C4 07C8 07CC 07D0 07D4 07D8 07D0 07E0 07E0 07E0 07E6 07F0 07F6 07F6	05 03 05 04 05 05 05 06 05 07 05 08 05 09 05 08 05 00 05 00 05 00 01 38 02 38 05 38 01 30 02 30 01 42	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         40           00         46           80         46	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 C DB NA, 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lo C DB NL, 66, 0, CTL_KEY ; Move PA1 to lower case F C DB NA, 66, 0, 0	case key ! Swér key 6] Key 66
624 6256 627 6289 6311 6333 63345 63345 63345 6336 63345 6337 6339 6412 6412 6433	07C4 07C8 07CC 07D0 07D4 07D8 07D0 07C0 07C0 07C0 07C0 07C0 07F0 07F6 07F6 07F6 07F6 0800 0804 0808	05 03 05 04 05 05 05 06 05 07 05 08 05 07 05 08 05 07 05 08 05 07 05 08 05 02 05 00 01 38 05 38 01 30 02 30 01 42 05 42	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 C DB NA, 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lo C DB NL, 66, 0, CTL_KEY ; Move PA1 to lower case F C DB NA, 66, 0, 0	case key ! Swér key 6] Key 66
624 625 627 628 630 631 633 633 633 633 6336 6336 6336 6336 6336 6336 6336 6336 6411 6423 6441 6442 6444	07C4 07C8 07CC 07D0 07D4 07D8 07DC 07E0 07E0 07E0 07E0 07F0 07F6 07F6 07F6 07F6 07F6 07F6 07F	05 03 05 04 05 05 05 06 05 07 05 08 05 07 05 04 05 05 07 05 04 05 07 05 04 05 07 05 08 05 07 05 08 06 05 07 05 08 07 05 08 07 05 08 05 07 05 08 00 07 05 08 00 07 05 08 00 07 00 01 38 00 02 38 00 01 30 00 01 30 00 01 30 01 30 0000000000	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         00           00         46           80         46           00         F4           00         00	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Hove CLEAR to upper/lower C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 C DB NL, 59, PC, 71 C DB NL, 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower C DB NL, 66, 0, CTL_KEY ; Move PA1 to lower case H C DB NA, 66, 0, 0 C DB NA, 66, 0, 0 C DB NL, 67, 0, ALT_KEY ; Move PA2 to lower case H	case key ! Swér key 6] Key 66
624 625 627 628 630 631 633 633 633 633 633 633 641 234 642 390 641 234 645	07C4 07C8 07CC 07D0 07D4 07D8 07D0 07C0 07C0 07C0 07C0 07C0 07F0 07F6 07F6 07F6 07F6 0800 0804 0808	05 03 05 04 05 05 05 06 05 07 05 08 05 07 05 04 05 05 07 05 04 05 07 05 04 05 07 05 08 05 07 05 08 06 05 07 05 08 07 05 08 07 05 08 05 07 05 08 00 07 05 08 00 07 05 08 00 07 00 01 38 00 02 38 00 01 30 00 01 30 00 01 30 01 30 0000000000	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 C DB NA, 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower C DB NL, 66, 0.CTL_KEY ; Move PA1 to lower case F C DB NA, 66, 0, 0 C DB NA, 667, 0, 0	case key ! ouer key 6] key 66 key 67
624 6256 62789 6312 63312 63345 6336 6336 6336 63378901 64423 64456 64456	07C4 07C8 07CC 07D0 07D4 07D8 07D7 07D7 07E8 07EC 07F0 07F6 07F6 07F6 07F6 0800 0804 0808 0804 0808	05         03           05         04           05         05           05         05           05         07           05         08           05         07           05         08           05         09           05         00           01         38           01         30           01         42           05         42           01         43           05         43	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         00           00         00           00         46           00         00           00         00           00         00           00         00           00         00	C DB NA, 4, 0.0 C DB NA, 5, 0,0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 8, 0.0 C DB NA, 8, 0.0 C DB NA, 10, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NL, 59,PC,71 ; Hove CLEAR to upper/lower C DB NU 59,PC,71 C DB NU 59,PC,71 C DB NU 59,PC,71 C DB NL, 59, 0, 0 C DB NL, 61,PL,70 ; Move CURSR SEL to upper/lower C DB NU 61,PU,70 C DB NL, 66, 0,CTL_KEY ; Move PAI to lower case F C DB NL, 66, 0, 0 C DB NL, 67, 0,ALT_KEY ; Move PA2 to lower case F C DB NL, 67, 0, ALT_KEY ; Move PA2 to lower case F C DB NL, 67, 0, 0	case key ! ouer key 6] key 66 key 67
624 6256 62256 62289 633123456 63366336 63366336 6336633890 64412344566 644445667	07C4 07C8 07CC 07D0 07D4 07D8 07DC 07E0 07E0 07F0 07F6 07F6 07F6 07F6 07F6 07F6 07F	05 03 05 04 05 05 05 06 05 07 05 08 05 09 05 08 05 00 05 00 05 00 05 00 05 38 01 30 02 30 01 42 05 42 01 43 05 43 01 42	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         00           00         46           80         46           00         F4           00         00           00         50           00         50           00         50           00         50           00         50           00         50           00         50           00         50           00         50           00         50	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 66, 0, CTL_KEY ; Move PA1 to lower case F C DB NL, 66, 0, 0 C DB NL, 67, 0, ALT_KEY ; Move PA2 to lower case F C DB NL, 76, PL, 59 ; Move PF1-12 to lower case F	case key ! ouer key 6] key 66 key 67
624 6256 626789 631234 633234 6335678901234 6335678901234 6442345678901234 6445678901234	07C4 07C8 07CC 07D0 07D4 07D8 07DC 07E0 07F0 07F0 07F0 07F0 07F0 07F7 07F6 07F6	05 03 05 04 05 05 05 05 06 05 07 05 08 05 07 05 08 05 00 05 00 01 38 05 38 01 30 02 30 01 42 05 43 01 42 05 43 01 42 01 44	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         44           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         38	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 10, 0.0 C DB NA, 11, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 C DB NA, 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower C DB NL, 66, 0, CTL_KEY ; Move PA1 to lower case F C DB NL, 66, 0, 0 C DB NA, 66, 0, 0 C DB NL, 67, 0, ALT_KEY ; Move PA2 to lower case F C DB NL, 76, PL, 59 ; Move PF1-12 to lower case F C DB NL, 77, PL, 60 ; keypad	case key ! ouer key 6] key 66 key 67
624 6256 6289 6312 63312 633456 6336 633456 6339 634423456 6444456 644456 644789	07C4 07C8 07CC 07D0 07D4 07D8 07D7 07D7 07F0 07F0 07F0 07F6 07F6 07F6 0800 0804 0808 0804 0808 0800 0814 0814	05 03 05 04 05 05 05 05 05 06 05 07 05 08 05 08 05 08 05 08 05 08 05 02 05 00 01 38 05 38 01 30 02 30 01 42 05 42 01 43 05 43 01 42 01 42	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         447           40         47           00         00           00         46           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         30	C DB NA, 4, 0.0 C DB NA, 5, 0,0 C DB NA, 6, 0.0 C DB NA, 7, 0,0 C DB NA, 8, 0.0 C DB NA, 9, 0.0 C DB NA, 10, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 15, 0.0 C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 C DB NL, 59, 0, 0 C DB NL, 61, PL, 70 C DB NL, 66, 0, CTL_KEY; Move PA1 to lower case 1 D DB NA, 66, 0, 0 C DB NL, 67, 0, ALT_KEY; Move PA2 to lower case 1 C DB NL, 77, PL, 50 C DB NL, 76, PL, 59 C DB NL, 76, PL, 59 C DB NL, 76, PL, 50 C DB NL, 77, PL, 60 C DB NL, 78, PL, 61	case key ! ouer key 6] key 66 key 67
$\begin{array}{c} 624\\ 625\\ 6225\\ 6228\\ 6228\\ 901\\ 2334\\ 56334\\ 56334\\ 56338\\ 901\\ 1234\\ 5644\\ 6445\\ 6445\\ 6446\\ 6447\\ 890\\ 1234\\ 567\\ 890\\ 1234\\ 566\\ 644\\ 644\\ 644\\ 644\\ 644\\ 644\\ 64$	07C4 07C8 07C6 07D0 07D4 07D8 07D7 07D6 07D6 07D6 07F0 07F6 07F6 07F6 07F6 07F6 07F6 07F	05 03 05 04 05 05 05 06 05 07 05 08 05 07 05 08 05 07 05 08 05 07 05 08 05 07 05 08 05 02 05 08 05 02 01 38 05 38 01 30 02 30 01 42 05 42 01 43 05 43 01 42 01 44 01 44 01 44	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         00           00         46           80         46           00         F4           00         00           00         3D           00         3C           00         3Z           00         3E	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 8, 0.0 C DB NA, 9, 0.0 C DB NA, 10, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 66, 0, 0 C DB NL, 66, 0, CTL_KEY ; Move PA1 to lower case P C DB NL, 66, 0, CTL_KEY ; Move PA1 to lower case P C DB NL, 67, 0, ALT_KEY ; Move PA2 to lower case P C DB NL, 76, PL, 59 ; Move PF1-12 to lower case C DB NL, 76, PL, 59 ; Move PF1-12 to lower case C DB NL, 77, PL, 60 ; keypad C DB NL, 79, PL, 62	case key ! ouer key 6] key 66 key 67
624 6256 6276 6289 6312 63312 6336 6336 6336 6336 6336 6336 6336 6336 6336 6336 6336 6412 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6445 6455 6551	07C4 07C8 07CC 07D0 07D4 07D8 07D2 07E0 07E0 07F6 07F6 07F6 07F6 07F6 07F6 07F6 07F	05 03 05 04 05 05 05 06 05 07 05 08 05 09 05 08 05 00 05 00 05 00 05 00 05 00 05 00 05 00 05 00 01 38 05 38 01 30 02 30 01 42 05 42 01 43 05 43 01 42 01 44 01 44 01 45 01 50	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         46           80         46           00         50           00         50           00         50           00         50           00         38           00         32           00         35	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 10, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 C DB NL, 59, PC, 71 C DB NL, 59, 0, 0 C DB NL, 66, 0, CTL_KEY ; Move PA1 to lower case F C DB NL, 66, 0, CTL_KEY ; Move PA1 to lower case F C DB NL, 66, 0, 0 C DB NL, 66, 0, 0 C DB NL, 67, 0, ALT_KEY ; Move PA2 to lower case F C DB NL, 76, PL, 59 ; Move PF1-12 to lower case F C DB NL, 77, PL, 60 ; keypad C DB NL, 78, PL, 61 C DB NL, 78, PL, 62 C DB NL, 78, PL, 63	case key ! ouer key 6] key 66 key 67
624 6256 6289 63312 633456 633456 63345678901234567890122 6444567890122	07C4 07C8 07CC 07D0 07D4 07D5 07D6 07D6 07D6 07E6 07E6 07F6 07F6 07F6 07F6 07F6 07F6 07F6 07F	05 03 05 04 05 05 06 05 06 05 07 05 08 05 08 05 08 05 08 05 00 05 00 01 38 05 38 01 30 02 30 01 42 05 42 01 43 01 42 01 44 01 45 01 51	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           40         47           00         00           00         46           80         46           00         50           00         50           00         30           00         32           00         32           00         35           00         34	C DB NA, 4, 0.0 C DB NA, 5, 0,0 C DB NA, 6, 0.0 C DB NA, 7, 0,0 C DB NA, 8, 0.0 C DB NA, 9, 0.0 C DB NA, 10, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL 59, PC, 71 C DB NL 59, PC, 71 C DB NL 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower C DB NL 66, 0, CTL_KEY ; Move PA1 to lower case 1 D B NL 66, 0, 0 C DB NL 67, 0, ALT_KEY ; Move PA2 to lower case 1 D B NL 76, PL, 59 ; Move PF1-12 to lower case 1 C DB NL 77, PL, 60 ; keypad C DB NL 79, PL, 62 C DB NL 60, PL, 63 C DB NL 79, PL, 62 C DB NL 71, PL, 64	case key ! ouer key 6] key 66 key 67
$\begin{array}{c} 624\\ 625\\ 6225\\ 6228\\ 901\\ 2334\\ 56334\\ 56334\\ 56334\\ 56334\\ 56334\\ 5644\\ 12345\\ 6444\\ 6444\\ 6444\\ 6447\\ 890\\ 123\\ 655\\ 23\end{array}$	07C4 07C8 07C6 07D0 07D4 07D8 07D7 07E0 07E0 07E0 07E0 07E0 07E0 07E0	05 03 05 04 05 05 05 05 05 06 05 07 05 08 05 07 05 08 05 07 05 08 05 07 05 08 05 07 05 08 05 02 05 00 01 38 05 38 01 30 02 30 01 42 05 42 01 43 05 43 01 42 01 44 01 44 01 44 01 45 01 55	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         30           00         30           00         32           00         32           00         34           00         34	C DB NA , 4, 0.0 C DB NA , 5, 0.0 C DB NA , 6, 0.0 C DB NA , 7, 0.0 C DB NA , 8, 0.0 C DB NA , 9, 0.0 C DB NA , 10, 0.0 C DB NA ,11, 0.0 C DB NA ,12, 0.0 C DB NA ,13, 0.0 C DB NA ,13, 0.0 C DB NA ,59, PC,71 ; Move CLEAR to upper/lower C DB NL ,59, PC,71 ; Move CLEAR to upper/lower C DB NL ,59, PC,71 ; Move CURSR SEL to upper/lower C DB NL ,59, PC,71 ; Move CURSR SEL to upper/lower C DB NL ,61, PL,70 ; Move CURSR SEL to upper/lower C DB NL ,61, PL,70 ; Move PA1 to lower case P C DB NL ,66, 0, CTL_KEY ; Move PA1 to lower case P C DB NL ,66, 0, CTL_KEY ; Move PA2 to lower case P C DB NL ,67, 0, ALT_KEY ; Move PA2 to lower case P C DB NL ,76, PL,59 ; Move PF1-12 to lower case P C DB NL ,76, PL,61 ; keypad C DB NL ,79, PL,62 C DB NL ,80, PL,63 C DB NL ,81, PL,64 C DB NL ,82, PL,65	case key ! ouer key 6] key 66 key 67
$\begin{array}{c} 624\\ 625\\ 6228\\ 6228\\ 6228\\ 6331\\ 2334\\ 56335\\ 6335\\ 6335\\ 6335\\ 6335\\ 6336\\ 6344\\ 1234\\ 56445\\ 6445\\ 6445\\ 64552\\ 34\\ 55234\\ 6552\\ 34\\ 555\\ 655\\ 655\\ 655\\ 655\\ 655\\ 655\\ 65$	07C4 07C8 07CC 07D0 07D4 07D8 07D2 07D2 07D2 07D2 07D2 07D2 07D2 07D2	05         03           05         04           05         05           05         06           05         06           05         07           05         08           05         09           05         08           05         00           05         00           05         00           01         30           01         42           05         43           05         43           01         42           01         42           01         42           01         42           01         42           01         42           01         42           01         42           01         42           01         42           01         43           01         42           01         42           01         42           01         51           01         51           01         53           01         53           01         <	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           40         47           00         00           00         46           80         46           00         50           00         50           00         50           00         50           00         38           00         32           00         35           00         37           00         44	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 C DB NA, 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower C DB NL, 66, 0.CTL_KEY ; Move PA1 to lower case N C DB NL, 66, 0.CTL_KEY ; Move PA1 to lower case N C DB NL, 66, 0, 0 C DB NL, 67, 0, ALT_KEY ; Move PA2 to lower case N C DB NL, 76, PL, 59 ; Move PF1-12 to lower case N C DB NL, 76, PL, 59 ; Move PF1-12 to lower case N C DB NL, 76, PL, 60 ; keypad C DB NL, 70, PL, 60 C DB NL, 70, PL, 60 C DB NL, 81, PL, 65 C DB NL, 81, PL, 65 C DB NL, 83, PL, 66	case key ! ower key 6] key 66 key 67
624 6256 6289 6312 633456 633456 633456 633456 633456 644456 644456 644456 65512345 6553455	07C4 07C8 07CC 07D0 07D4 07D8 07D7 07D7 07D7 07F0 07F0 07F7 07F6 07F7 0800 0804 0808 0804 0808 0804 0808 0804 0808 0810 0814 0816 0812 0820 0814 0822 0822 0824 0826 0824 0826 0824 0826 0824 0826 0826 0826 0826 0826 0826 0826 0826	05         03           05         04           05         05           05         05           05         06           05         07           05         08           05         09           05         08           05         00           01         38           01         30           02         34           05         43           05         44           01         42           01         42           01         42           01         43           05         43           01         42           01         44           01         42           01         44           01         44           01         44           01         44           01         54           01         54           01         54           01         54           01         54           01         54           01         54           01         <	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         46           00         00           00         00           00         50           00         00           00         00           00         00           00         00           00         30           00         30           00         30           00         40           00         41           00         43	C DB NA, 4, 0.0 C DB NA, 5, 0,0 C DB NA, 6, 0.0 C DB NA, 8, 0.0 C DB NA, 8, 0.0 C DB NA, 9, 0.0 C DB NA, 10, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 12, 0.0 C DB NA, 33, 0.0 C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 ; Move CLEAR to upper/lower C DB NL, 59, PC, 71 C DB NA, 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower C DB NL, 66, 0, CTL_KEY ; Move PA1 to lower case 1 D B NL, 66, 0, 0 C DB NL, 66, 10, 0 C DB NL, 76, PL, 59 ; Move PA1 to lower case 1 D B NL, 76, PL, 59 ; Move PF1-12 to lower case 1 D B NL, 76, PL, 59 ; Move PF1-12 to lower case 1 D B NL, 76, PL, 60 ; keypad C DB NL, 79, PL, 60 ; keypad C DB NL, 82, PL, 65 C DB NL, 82, PL, 65 C DB NL, 84, PL, 67	case key ! swer key 61 key 66 key 67
$\begin{array}{c} 624\\ 625\\ 6425\\ 6425\\ 6425\\ 6427\\ 89\\ 6331\\ 2334\\ 56336\\ 6335\\ 6335\\ 6336\\ 6336\\ 6340\\ 6442\\ 3445\\ 6445\\ 6445\\ 64552\\ 34\\ 55234\\ 655234\\ \end{array}$	07C4 07C8 07CC 07D0 07D4 07D8 07D7 07D7 07D7 07F0 07F0 07F7 07F6 07F7 0800 0804 0808 0804 0808 0804 0808 0804 0808 0810 0814 0816 0812 0820 0814 0822 0822 0824 0826 0824 0826 0824 0826 0824 0826 0826 0826 0826 0826 0826 0826 0826	05         03           05         04           05         05           05         06           05         06           05         07           05         08           05         09           05         08           05         00           05         00           05         00           01         30           01         42           05         43           05         43           01         42           01         42           01         42           01         42           01         42           01         42           01         42           01         42           01         42           01         42           01         43           01         42           01         42           01         42           01         51           01         51           01         53           01         53           01         <	00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         00           00         46           00         00           00         00           00         50           00         00           00         00           00         00           00         00           00         30           00         30           00         30           00         40           00         41           00         43	C DB NA, 4, 0.0 C DB NA, 5, 0.0 C DB NA, 6, 0.0 C DB NA, 7, 0.0 C DB NA, 9, 0.0 C DB NA, 9, 0.0 C DB NA, 11, 0.0 C DB NA, 11, 0.0 C DB NA, 12, 0.0 C DB NA, 13, 0.0 C DB NA, 13, 0.0 C DB NA, 59, PC, 71 ; Move CLEAR to upper/lower C DB NA, 59, 0, 0 C DB NL, 59, PC, 71 C DB NA, 59, 0, 0 C DB NL, 61, PL, 70 ; Move CURSR SEL to upper/lower C DB NL, 66, 0.CTL_KEY ; Move PA1 to lower case N C DB NL, 66, 0.CTL_KEY ; Move PA1 to lower case N C DB NL, 66, 0, 0 C DB NL, 67, 0, ALT_KEY ; Move PA2 to lower case N C DB NL, 76, PL, 59 ; Move PF1-12 to lower case N C DB NL, 76, PL, 59 ; Move PF1-12 to lower case N C DB NL, 76, PL, 60 ; keypad C DB NL, 70, PL, 60 C DB NL, 70, PL, 60 C DB NL, 81, PL, 65 C DB NL, 81, PL, 65 C DB NL, 83, PL, 66	case key ! ower key 6] key 66 key 67

.

	155		4,64	41,262	156	
657 658	083C 01 56 80 38 0840 01 57 80 3C	c c	DB DB	NL ,86,PU,59 NL ,87,PU,60		
659 660 661	0844  02 4C 00 08 0848  02 4D 00 09	с с с	DB	NU ,76,PL, 8	; Nove numeric keys to up	per case on
662	084C 02 4E 00 0A	C	DB DB	NU ,77,PL, 9 NU ,78,PL,10	; outboard key pad:	
663	0850 02 4F 00 05	C	DB	NU ,79,PL, 5	; 789	
664 665	0854 02 50 00 06 0858 02 51 00 07	C C	DB DB	NU ,80,PL, 6 NU ,81,PL, 7	; 456	,
666	0850 02 52 00 02	C	DB	NU ,82,PL, 2	i 0>1	
667 668	0860  02 53 00 03 0864  02 54 00 04	с с	DB DB	NU ,83,PL, 3 NU ,84,PL, 4		
669	0868 02 55 00 0B	С	DB	NU .85.PL.11		
670 671	086C 02 56 00 34 0870 02 57 00 0F	с с	0B	NU ,86,PL,52 NU ,87,PL,15		
672		c	DB	NU ,0/,PL,13		
673 674	0874 05 4C 80 3D 0878 05 4D 80 3E	C	DB	NA ,76,PU,61	; Move PF13-24 to ALT cas	e on outboard
675	087C 05 4E 80 3F	C C	DB DB	NA ,77,PU,62 NA ,78,PU,63	; keypad	
676	0880 05 4F 80 40	C	DB	NA ,79, PU,64		
677 678	0884 05 50 80 41 0888 05 51 80 42	C C	0B 0B	NA ,80,PU,65 NA ,81,PU,66		
679	088C 05 52 80 43	С	DB	NA ,82,PU,67		
680	0890 05 53 80 44	C	DB	NA ,83,PU,68		
681 682	0894 05 54 40 38 0898 05 55 40 3C	C C	08 08	NA .84,PC.59.		
683	089C 05 56 40 3D	č	DB	NA ,86,PC,61		
684 685	08A0 05 57 40 3E	с с	DB	NA ,87,PC,62		
686	08A4 00 00 00 00	č	DB	0, 0, 0, 0	; End of inbound table cha	inges
687		С С ;	Dutha	und translation t	the changes	
688 689		C ; C	00100			
690	08A8 92 00 3D	c	DB	092H,L,61	; Move CURSR SEL to lower ; Move CLEAR to lower case	case Key ol e kev 59
691 692	08AB 93 00 3B 08AE 96 00 00	с с	DB, DB,	093H,L,59 096H,0, 0	; Remove dead key	
693	0881 A0 80 42	С	DB	DADH, U, 66	; Hove DUP to upper case     Move PA1 to lower case	key 66 Kev 66
694 695	0884 A1 00 42 0887 A2 80 43	с с	DB DB	0A1H,L,66 DA2H,U,67	; Hove Field Mark to upper	r case key 67
696	088A A3 00 43	С	DB	0A3H, L, 67	; Move PA2 to lower case !	key 67
697 698	08BD B0 00 4C	С . С	DB	0B0H, L, 76	; Move F1-24 to outboard	keypad
699	08C0 B1 00 4D	c	DB	0B1H,L,77		
700 701	08C3 B2 00 4E 08C6 B3 00 4F	C C	DB	082H,L,78 083H,L,79		
702	08C9 B4 00 50	С	DB	0B4H,L,80		
703 704	08CC 85 00 51 08CF 86 00 52	с с	DB DB	085H,L,81 086H,L,82		
705	0802 87 00 53	C.	CB	0B7H, L, 83		
706 707	08D5 88 00 54 08D8 89 00 55	c c	DB DB	088H, L, 84		
708	08DB BA 00 56	c	DB	089H,L,85 08AH,L,86		
709 710	08DE 88 00 57 08E1 8C 20 4C	C C	DB DB	088K, L, 87	-	
711	08E1 BC 20 4C 08E4 BD 20 4D	c	DB	0BCH, A, 76 0BDH, A, 77		
712	08E7 BE 20 4E	c	DB	0BEH, A, 78		
713 714	08EA BF 20 4F 08ED C0 20 50	2 2	08 08	08FH,A,79 0C0H,A,80		
715	08F0 C1 20 51	C	DB	0C1H,A,81		
716 717	08F3 C2 20 52 08F6 C3 20 53	C C	DB DB	0C2H,A,82 DC3H,A,83		
718	08F9 C4 20 54	С	DB	DC4H,A,84		
719 720	08FC C5 20 55 08FF C6 20 56	C C	DB DB	0C5H,A,85 0C6H,A,86		
721	0902 C7 20 57	С	DB	0C7H,A,87		
722 723	0905 00 00 00	с с	DB	000H,0, 0	; End of outbound table c	hanges
724		С				
725 726		с с	INCLU	DE USEDATA.ASM	; DATA ENTRY X	LAT TABLE
727		C		'L Data Entry Ke	yboard	
728		C C				
729 730		C ;*	*****	****	* 专家学家教育学家教育教育学校学校学校学校学校学校学校学校	******
731 732		C ;* C ;*	Data	Entry Keyboard Ti	ranslation Table	
733		<b>C</b> + <b>X</b>				
734 735		C ;* C	****	*****	****	
736	0908	C DA	TA_ENTRY LA	BEL BYTE		
737 738		с с				
739		С;	ALPHA	CASE		
- 740 741	0908 00 29 00 3B 80 03	с 3 С	DB	L,41,L,59,U,	3,0, 6,0, 9	; 1-5
742	80 06 80 09	С				; 6-10
743 744	0912 80 33 00 F6 00 F2 00 0C 00 35	2 C C	DB	-	P,0,PAUSE_KEY,L,12,L,53	
745 746	091C 00 3C 00 3D 00 31 00 0E 00 0F		DB	L,60,L,61,L,6	2, L, 14, L, 15	; 11-15
747	0926 80 10 80 11 80°1	2 C	DB	U, 16, U, 17, U, 1	8,U,19,U,20	; 16-20
748 749	80 13 80 14 0930 80 15 80 16 80 11		DB	U,21,U,22,U,2	3,0,24,0,25	1 21-25
750	80 18 80 19	С				•

### 

		157
751	093A	00 3F 00 40 80 0F C
752 753	0944	07 FC 80 1E C 80 1F 80 20 80 21 C
754 755	094E	80 22 80 23 C
756		80 1A 00 41 C
757 758	0958	00 42 00 1C 00 FB C 00 43 80 2C C
759 760	0962	80 2D 80 2E 80 2F C 80 30 80 31 C
761 762	096C	80 32 00 33 00 34 C 00 44 00 FA C
763 764	0976	00 FB 00 1C 40 46 C 01 46 00 00 C
765 766	0980	00 00 00 00 40 4F C 80 37 00 F7 C
767 768	098A	00 F4 00 F3 01 52 C 00 53 00 48 C
769 770	0994	00 50 00 4B 00 4D C 00 F9 C0 39 C
771	099E	00 00 00 00 00 00 C
773	0948	00 00 00 00 00 00 C
774 775	09B2	00 00 00 00 00 00 C
776 777		00 00 C C
778 779		C ; C
780 781	098A	80 29 00 38 80 04 C 00 33 80 05 C
782 783	0904	DO 34 OD F6 OD F2 C DO 0C DO 0B C
784 785	09CE	DO 3C DO 3D OD 3E C 80 0E DO 0F C
786 787	0908	80 00 80 0C 80 0B C 00 1A 00 2B C
788 789	09E2	80 28 00 02 00 03 C 00 04 80 08 C
750 791	09EC	00 3F 00 40 80 0F C 00 FC 00 1B C
792 793	09F6	80 34 80 27 00 27 C 80 07 00 28 C
794	0040	00 05 00 06 00 07 C
795 796	0A0A	80 1B 00 41 C 00 42 00 1C 00 FB C
797 798	0A14	00 43 00 00 C 80 35 80 28 00 0D C
799 800	OAIE	80 02 80 0A C 00 08 00 09 00 0A C
801 802	0428	DD 44 00 FA C DD F8 00 1C 40 46 C
803 804	0A32	81 46 00 00 C 00 00 00 00 40 4F C
805 806	0A3C	80 37 00 F7 C 00 F4 00 F3 01 52 C
807 606	0A46	00 53 00 48 C 00 50 00 48 00 4D C
809 810	0A50	00 F9 80 39 C 00 00 00 00 00 00 C
811 812	DASA	00 00 00 00 00 00 00 00 00 00 00 00 00
813 814	0464	00 00 00 00 00 C 00 00 00 00 00 C
815 816		00 00 C C
817 818		C ; C
819 820	0A6C	00 00 00 00 40 03 C 00 00 00 00 C
821	0476	00 00 40 07 00 00 C 40 0C 00 00 C
822 823	DABO	DO DO 40 00 00 00 C 40 DE 00 00 C
824 825	DASA	40 10 40 11 40 12 C
826 827	0A94	40 15 40 16 40 17 C
828 829	CA9E	40 1A 40 2B 40 47 C
830 831	BAAD	40 1F 40 20 40 21 C
832 833	DAB2	40 22 40 23 C 40 24 40 25 40 26 C
834 835	OABC	40 1B 00 00 C 00 00 00 00 00 FB C
836 837	0AC6	00 00 40 2C C 40 2D 40 2E 40 2F C
838 839	OADO	40 30 40 31 C 40 32 00 00 00 00 C
840 841	DADA	00 00 00 FA C 00 F8 00 00 00 00 C
642 643	DAE4	00 00 00 00 00 C 00 00 00 00 00 00 C
844 845	OAEE	40 37 00 00 C 00 00 00 00 00 00 C
643	VALL	

.

100	
L,63,L,64,U,15,0,NUM_LOCK,U,30	; 26-30
U, 31, U, 32, U, 33, U, 34, U, 35	; 31-35
U, 36,U, 37,U, 38,U, 26,L,65	; 36-40
L,66,L,28,0,NUM_SHIFT,L,67,U,44	; 41-45
U,45,U,46,U,47,U,48,U,49	; 46-50
U,50,L,51,L,52,L,68,0,ALPHA_SHIFT	1 51-55
0,RESET,L,28,C,70,L OR 8,70,0,0	; 56-60
0,0,0,0,C,79,U,55,0,CLICK	; 61-65
0,CTL_KEY,0,ALT_KEY,L OR B,82,L,83,L,72	; 66-70
L,80,1,75,1,77,0,ALT_SHIFT,1,57	; 71-75
0, 0,0, 0,0, 0,0, 0,0, 0	; 76-80
0, 0,0, 0,0, 0,0, 0,0, 0	; 81-85
0, 0,0, 0,0, 0,0, 0	; 86-89

### NUM CASE

DB	U,41,L,59,U, 4,L,51,U, 5	; 1-5
DB	L,52,0,DEC_INP,0,PAUSE_KEY,L,12,L,11	; 6-10
DB	L,60,L,61,L,62,U,14,L,15	; 11-15
DB	U,13,U,12,U,11,L,26,L,43	; 16-20
DB	U,43,L, 2,L, 3,L, 4,U, 8	; 21-25
DB	L,63,L,64,U,15,0,NUM_LOCK,L,27	; 26-30
DB	U.52,U,39,L,39,U, 7,L,40	; 31-35
DB	L, 5,L, 6,L, 7,U,27,L,65	; 36-40
DB	L,66,L,28,0,NUM_SHIFT,L,67,0, 0	; 41-45
DB	U,53,U,40,L,13,U, 2,U,10	; 46-50
DB	L, 8,L, 9,L,10,L,68,0,ALPHA_SHIFT	; 51-55
DB	0,RESET,L,28,C,78,U OR 8,70,0,0	; 55-60
DB	0, 0,0, 0,C,79,U,55,0,CLICK	; 61-65
DB	0,CTL_KEY,0,ALT_KEY,L OR B,82,L,83,L,72	; 66-70
DB	L,80,L,75,L,77,0,ALT_SHIFT,U,57	; 71-75
0B	0, 0,0, 0,0, 0,0, 0,0, 0 <u>.</u>	; 76-80
DB	0, 0,0, 0,0, 0,0, 0,0, 0	; 81-85
DB	0, 0,0, 0,0, 0,0, 0	; 86-89

### PERSONAL COMPUTER CONTROL CASE

DB	0, 0,0, 0,C, 3,0, 0,0, 0	; 1-5
DB	0, 0,C, 7,0, 0,C,12,0, 0	; 6-10
DB .	D, D,C, 0,0, 8,C,14,0, 8	; 11-15
DB	C,16,C,17,C,18,C,19,C,20	; 16-20
DB	C,21,C,22,C,23,C,24,C,25	; 21-25
DB	C,26,C,43,C,71,C,NUM_LOCK,C,30	; 26-30
DB	C,31,C,32,C,33,C,34,C,35	; 31-35
DB	C,36,C,37,C,38,C,27,0, 0	; 36-40
DB	0, 0,0, 0,0,NUM_SHIFT,0, 0,C,44	; 41-45
DB	C, 45, C, 46, C, 47, C, 48, C, 49	; 46-50
DB	C,50,0, 0,0, 0,0, 0,0,ALPHA_SHIFT	; 51-55
DB	0,RESET,0, 0,0, 0,0, 0,0, 0	; 56-60
DB	0, 0,0, 0,0, 0,0,55,0, 0	; 61-65
DB	0, 0,0, 0,0, 0,C OR A,83,C,73	; 66-70

			4,64	41,262	
	159			160	
OAF8	60 53 40 49 40 51 00 00 00 00	C C	DB	C,81,0, 0,0, 0,0,ALT_SHIFT,C,57	; 71-75
	00 F9 40 39	С			; 76-80
0202	00 00 00 00 00 00 00 00 00 00 00	с с	DB	C, D,O, D,O, O,O, Ô,Ô, Ô	
OBOC	00 00 00 00 00 00 00 00 00 00	C C	CB	0, 0,0, 0,0, 0,0, 0,0, 0	; 81-85
0B16	00 00 00 00 00 00	С	DB	0, 0,0, 0,0, <b>0,0, 0</b>	; 86-89
	00 00	с с			
		C ; C	PERSON	AL COMPUTER ALT CASE	
OBIE	00 00 20 02 20 03	С	DB	0, 0,Å, 2,Å, 3,Å, 4,Å, 5	; 1-5
0B28	20 04 20 05 20 06 20 07 20 08	с с	DB	A, 6, A, 7, A, 8, A, 9, A, 10	; 6-10
0332	20 09 20 0A 20 0B 20 0C 20 DD	C C	DB	A,11,A,12,A,13,0, 0,0, 0	; 11-15
0B3C	00 00 00 00 20 10 20 11 20 12	C C	DB	A, 16, A, 17, A, 18, A, 19, A, 20	; 16-20
	20 13 20 14	С			; 21-25
CB46	20 15 20 16 20 17 20 18 20 19	с с	DB	A,21,A,22,A,23,A,24,A,25	
0850	00 00 00 00 00 00 00 FC 20 1E	с С	DB	0, 0,0, 0,0, 0,0,NUM_LOCK,A,30	; 26-30
CE5A	20 1F 20 20 20 21	c c	DB	A, 31, A, 32, A, 33, A, 34, A, 35	; 31-35
0864	20 22 20 23 20 24 20 25 20 26	C	DB	A, 36, A, 37, A, 38, 0, 0, 0, 0	; 36-40
086E	00 00 00 00 00 00 00 00 00 FB	с с	DB	0, 0,0, 0,0,NUM_SHIFT,0, 0.A,44	; 41-45
0878	00 00 20 2C 20 2D 20 2E 20 2F	C C	08	A.45, A.46, A.47, A,48, A.49	: 46-50
	20 30 20 31	С		A,50,0, 0,0, 0,0, 0,0,ALFHA_SHIFT	\$ 51-55
0B82	20 32 00 00 00 00 00 00 00 FA	с с	DB		; 56-60
OBAC	00 F8 00 00 00 00 00 00 <b>00</b> 00	C C	DB	0,RESET,0, 0,0, 0,0, 0,0, 0	
0B96	00 00 00 00 00 00	C C	DB	0, 0,0, 0,0, 0,0, 0,0, 0	; 61-65
OBAD	00 00 00 00 00 00 00 00 00 00 00	C	DB	D, C,C, D,C, O,B, C,C, D	; 66-70
OBAA	00 00 00 00 00 00 00	с с	-DB	0, 0,0, 0,0, 0,0,ALT_SHIFT,A,57	; 71-75
0BB4	00 F9 20 39 00 00 00 00 00 00	C C	DB	0, 0,0, 0,0, 0,0, 0,0, 0	; 76-80
	00 00 00 00	С		0, 0,0, 0,0, 0,0, 0,0, 0	; 81-85
OBBE	00 00 00 00 00 00 00	с с	DB		; 86-89
0BC8	00 00 00 00 00 00 00 00	с с	DB	0, 0,0, 0,0, 0,0, 0	,
		С	ALT CA	<pre><pre></pre></pre>	
		C		L, 1,0, 0,0, 0,0, 0,0, 0	; 1-5
0BD0	00 01 00 00 00 00 00 00 00 00	с с	DB		; 6-10
OBDA	00 00 00 00 00 00 00 00 00 00	с с	DB	0, 0,0, 0,0, 0,0, 0,0, 0	
0884	00 00 00 00 00 00	C C	DB	0, 0,0, 0,0, 0,0, 0,L,79	; 11-15
OBEE	00 00 00 4F 80 3B 80 3C 80 3D	С	DB	1,59,0,60,0,61,0,62,0,63	; 16-20
0BF8	80 3E 80 3F 80 40 80 41 80.42	с с	DB	U,64,U,65,U,66,U,67,U,68	; 21-25
	80 43 80 44 00 00 00 00 00 47	с с	DB	0, 0,0, 0,L,71,0,NUM_LOCK,C,59	; 26-30
	00 FC 40 38	с	DB	C,60,C,61,C,62,C,63,C,64	; 31-35
	40 3C 40 3D 40 3E 40 3F 40 40	c c			; 36-40
0016	40 41 40 42 40 43 40 44 00 00	C C	DB	C,65,C,66,C,67,C,68,D, 0	; 41-45
0C20	00 00 40 1C 00 FB	c c	DB	0, 0,C,28,0,NUM_SHIFT,0, 0,A,59	
0C2A	00 00 20 3B 20 3C 20 3D 20 3E	С	DB	A,60,A,61,A,62,A,63,A,64	; 46-50
0034	20 3F 20 40 20 41 20 42 20 43	с с	DB	A,65,A,66,A,67,A,68,0,ALPHA_SHIFT	; 51-55
0035	20 44 00 FA D0 F8 40 1C 00 00	с с	DB	0,RESET,C,28,0,0,C,71,0, 0	; 56-60
	40 47 00 0D	C C	DB	C,81,0, 0,0, 0,0, 0,D, 0	; 61-65
	40 51 00 00 00 00 00 00 00 00	С		0, C.O, O,O,SWITCH_MODE,O, O,L,73	3 66-70
0052	00 00 00 00 00 F5 00 00 00 49	с с	DB		; 71-75
0C5C	00 51 40 4B 40 4D 00 F9 00 39	c c	DB	L,81,C,75,C,77,0,ALT_SHIFT,L,57	
0066	00 00 00 00 00 00	C C	DB	0, 0,0, 0,0, 0,0, 0,0, 0	; 76-80
0070	00 00 00 00 00 00 00	с	DB	0, 0,0, 0,0, 0,0, 0,0, 0	; 81-85
0C 7A	00 00 00 00 00 00 00 00	с с	DB	0, 0,0, 0,0, 0,0, 0	; 86-89
	00 00	C C			
		с;	EHCOD	ED KEYSTROKE TRANSLATION TABLE	
0082	00 00 00 00 00 00	C C	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0	1 00-07
	00 00 00 00 00 00 00 00 00 00	с с			
0092		C C	DB	L,14,L,15,0, 0,0, 0,0, 0,L,42,0, 0,0, 0	; 08-0F
	00 00 00 00 00 EM	-			

### 162

; 10-17

; 18-1F

; 20-27

; 28-2F

; 30-37

; 38-3F

; 40-47

; 48-4F

; 50-57

; 58-5F

; 60-67

; 68-6F

; 70-77

; 78-7F

; 80-87

; 88-8F

; 90-97

; 98-9F

; AD-A7

; A6-AF

; 80-87

; 88-BF

; C0-C7

; C8-CF

; DO-D7

; D8-DF ; EO-E7

; E8-EF

; F0-F7

; F8-FF

. .

				4,64	1,262
		161			162
941 942	0CA2	00 00 00 00 00 00 00	C C	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0,
943	UCAL	00 00 00 00 00 00	c c		
944 945	OCB2	00 00 00 00 00 00 00	С	DB	0, 0,0, 0,0, 0,4,63,0, 0,0, 0,0, 0,0, 0
946 947		20 3F <b>00 00 00 00</b> 00 00 00 00	с с		
948 949	0002	CC 46 E0 31 80 2F 60 03 80 05 00 04	c c	DB	1,75,0,49,0,47,0, 3,0, 5,1, 4,0,25,0,35
950		80 19 80 23	С		
951 952	0CD2	80 32 80 12 00 05 80 10 00 34 00 09	с с	DB	U.50,U,18,L, 5,U,16,L,52,L, 9,L,53,L,10
953 954	DCE2	00 35 00 0A 80 0A 80 16 80 17	с с	DB	U, 10, U, 22, U, 23, U, 24, U, 36, U, 37, U, 38, U, S1
955 956		80 18 80 24 80 25 80 26 80 33	C C		
957 958	OCF2	80 34 80 35 80 20	С	DB	U,52,U,53,U,32,U,33,t, 6,U,48,U,31,U,46
959		80 21 00 06 80 30 80 1F 80 2E	c c		
960 961	0002	00 03 00 1E 00 31 00 2F 00 20 00 12	C C	DB	L, 3,L,30,L,49,L,47,L,32,L,18,L,33,L,34
962 963	0012	00 21 00 22 00 23 00 17 00 24	c c	DB	L,35,L,23,L,36,L,37,L,38,L,51,L,50,L,24
964		00 25 00 26 00 33	С .	00	
965 966	0D22	00 32 00 18 00 19 00 10 <b>00 13</b>	C C	DB	L,25,L,16,L,19,L,31,L,20,L,22,L,48,L,17
967 968		00 1F 00 14 00 16 00 30 00 11	С С <sup>,</sup>		
969 970	0D 32	00 2E 00 15 00 2D 00 00 80 14 00 00	c c	08	1,46,1,21,1,45,0, 0,0,20,0, 0,0, 0,0,17
971 972	0D42	CO DO 80 11 CO DO 80 00 00 00	c c	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0
973	0042	CO CO CO CO CO OO	С	UB	
974 975	0052	00 00 00 00 00 00 00 00 00	с с	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0
976 977		00 00 00 00 00 00 00 00 00 00 00	C C		
978 979	0D62	00 00 00 00 00 00 00 00 00 00 00 00 00	c c	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0
980 981	00.70	00 00 00 00	с		0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0
982	0072	00 00 00 00 00 00 00 00 00 00 00 00 00	C C	DB	
983 984	0D82	UO DO DO OD 30 15 80 22 80 13	c c	DB	U,21,U,34,U,19,L,14,L,15,L,28,L,42,A,28
985 986		00 DE 00 0F 00 1C 00 2A 20 1C	C C		
987 988	0D 92	00 38 20 38 00 <b>39</b> 20 00 02 00 00 <b>01</b>	с с	DB	L,56,A,56,L,57,A, 0,BA,0,L, 1,L,39,0, 0
989 990	0042	00 27 00 00	c c	0.0	L,58,A,58,L,59,A,59,L,60,A,60,L,61,A,61
991	ODA2	20 3B 00 3C 20 3C	С	DB	Li30(#,30)(Li37)#i37)(100/#i00/Li01/100/
992 993	0DB2	00 3D 20 3D 00 3E 20 3E <b>00 3</b> F	c c	DB	L,62,A,62,L,€3,A,63,L,64,A,64,L,65,A,65
994 995		20 3F 00 40 20 40 00 41 20 41	C C		
996 997	ODC2	00 08 00 42 00 07 00 43 00 44 20 44	C C	DB	L, 8,L,66,L, 7,L,67,L,68,A,68,L,69,A,69
998 999	0002	00 45 20 45 00 46 20 46 00 47	c c	DB	L,70,A,70,L,71,A,71,L,72,A,72,L,73,A,73
1000	0002	20 47 00 48 20 48	С	00	
1001 1002	0DE2	00 49 20 49 00 02 00 08 00 0C	c c	DB	L, 2,L,11,L,12,L,13,L,26,L,27,L,40,L,41
1003 1004		00 0D 00 1A 00 1B 00 28 00 29	C C		
1005 1006	ODF2	00 2C 00 36 00 00 00 00 00 00 00 00	C C	DB	L,44,L,54,0, 0,0, 0,0, C,0, 0,0, 0,0, 0
1007	0E02	00 00 00 00 00 00 00	с С	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0
1009	0202	00 00 00 00 00 00	c	05	
1010	0E12	CO DO OD OD CO DO OD OD <b>DO</b>	C C	DB	0, 0,0, 0,0, 0,0, 0,0, 0.0, 0,0, 0,0, 0
1012		D2 D0 00 00 00 00 00 00 00 00 00 00 00 00	с с		
1014 1015	0E22	<b>00 00 00 00 00 00 00</b> 00	с С	DB	0, 8,0, 8,0, 6,0, 0,0, 0,0, 0,0, 0,0, 0
1016	0572	00 00 00 00 00 00 00	C .	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0
1017	0635	CC 00 00 00 00 00	с	00	
1019 1020	0E42	00 00 00 00 00 00 00	C C	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0
1021 1022		00 00 00 00 00 00 00 00 00 00 00 00	с с		
1023	0E52	00 00 00 00 00 00 00 00 00 00 00	C C	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0
1025		00 00 00 00	c c	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0
1026	DE62	00 00 00 00 00 00 00 00 00 00 00 00 00	С.	<u>.</u>	•, •,•, •,•, •,•, •,•, •, •, •
1028 1029	0E 72	00 00 00 00 00 00 00	c c	DB	0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0,0, 0
1030 1031		00 00 00 00 00 00 00 00 00 00 00	с с		
1032			с с		
1034			c c	INCLUDE	USEKP.ASM 3 DATA ENTRY K Modifications For Data Entry Keypunch Keybo
1035			•	300112	

RY KEYPUNCH MODS Leyboard

### 4,641,262

### 164

			10	13			104
1036				C			
1037				C			
1038 1039				C	*******	*****	。 1. 张子文大学》》完全主要是在学校学校学校教育教育教育教育教育教育教育教育教育教育教育教育教育教育教育教育教育
1040				C C	141 141	M. J. K.	ations required to change Data Entry keyboard into
1041				c	3 M		Entry Keypunch keyboard
1042				c	34		, in the second s
1043				C	*****	*****	"我我就这家学家才当我在不能是我不能能能能这些你的我们都是我们不能能能不能没有这些我们不是我们不能不能能不能。"
1044				c			
1045 1046	0E82			C C	KP	LABEL	NORD
1047				_	;	Tobound	d translation table changes
1048				Č	•		
1049	0E82	01 08		C		DB	NL ,11, 0,RESET ; Put RESET on key 11
1050 1051	0686	02 0B		C		DB	NU ,11, O,RESET
1052	DE8A DE8E	03 08 04 08		C C		08 08	NPC,11, 0,RESET NPA,11, 0,RESET
1053	0E92	05 0B		č		DB	NA ,11. O,RESET
1054				C			
1055	0E96	01 DC		c		DB	NL ,12,PL,60 ; Put PF2 on key 12
1056 1057	0E9A 0E9E	02 DC 04 DC		C C		DB DB	NU ,12,PL,60 NPA,12,PA,11
1058		** **	20 00	č		00	NFA / 12 / FA / 1 1
1059	0EA2	01 OD		С		DB	NL ,13,PL,61 ; Put PF3 on key 13
1060 1061	OEA6	02 0D		c		DB	NU ,13,PL,61
1061	OEAA	04 OD	20 OC	C C		DB	NPA,13,PA,12
1063	OEAE	01 OE	00 3E	с с		DB	NL ,14,PL,62 ; Put PF4 on key 14
1064	OEB2	02 OE	00 3E	С		DB	NU ,14,PL,62
1065	OEB6	03 OE		c		DB	NPC,14, 0, 0
1066 1067	OEBA	04 OE	20 00	C C		DB	NPA,14,PA,13
1068	OEBE	01 1A	00 10	с С		DB	NL ;26,PL,28 ; Put ENTER on key 26
1069	DEC2	02 JA		č		DB	NU ,26,PL,28
1070	OEC6	05 1A	40 IC	С		DB	NA ,26.PC,28
1071				C			
1072 1073		01 1B 02 1B		C C		DB	NL ,27, PL, 63 ; Put PF5 on key 27
1074	ULUL	VC 10	00 3/	č		DB	NU ,27,PL,63
1075	DED2	01 28	00 40	c		DB	NL :40,PL,64 ; Put PF6 on key 40
1076	OED6	02 28	00 40	c		DB	NU ,40,PL,64
1077 1078	OFDA	01 29	00.41	C C		0.0	NY 61 DI 65 I Did BEZ on Koy 61
1079		02 29		с С		DB DB	NL ,41,PL,65 ; Put PF7 on key 41 NU ,41,PL,65
1080			•••••	č			
1081		D1 2A		с		DB	NL ,42,PL,66 ; Put PF8 on key 42
1082	OEE6	D2 2A		C C		DB	NU ,42,PL,66
1083 1084	OEEA	05 2A	00 00	с с		DB	NA ,42, 0, 0
1085	OEEE	01 36	00 OE	č		DB	NL 154,PL,14 ; Put Backspace on key 54
1086	0EF2	02 36		c		DB	NU ,54, PU,14
1087	OEF6	03 36	40 OE	c		DB	NPC,54,PC,14
1088 1089	OEFA	01 38	00 44	C C		DB	NL ,56,PL,68 ; Put PF10 on key 56
1090	OEFE	02 38		č		DB	NU ,56,PL,68
1091	0F02	03 38		С		DB	NPC.56, 0, 0
1092	0F06	04 38		c		DB	NPA,56, 0, 0
1093 1094	OFOA	05 38	00 00	C C		DB	NA ,56, 0, 0
1095	OFOE	00 00	00 00	c		DB	0, 0, 0, 0 ; End of inbound table changes
1096				С			
1097				C	;	Outbourn	d translation table changes
1098	0512	09 00	74	C C		80	ANDU L EA Dut Backgross on how EA
1099 1100	0F12 0F15	08 00 83 00		с с		DB DB	008H,L,54 ; Put Backspace on key 54 083H,L,54
1101	0F18	0D 00		с С		DB	00DH,L,57 1 Put RETURN on key 57
1102	OFIB	86 00	39	С		DB	086H,L,57
1103		88 00		c		DB	088H,L,11 ; Put RESET on key 11
1104 1105	OF 21 OF 24	8A 00 B1 00		с с		DB	08AH,L,26 ; Put ENTER on key 26
1105	0F27	B2 00		с С		DB DB	0B1H,L,12 ; Put PF2 on key 12 0B2H,L,13 ; Put PF3 on key 13
1107	OF2A			č		DB	0B3H,L,14 ; Put PF4 on key 14
1108	0F2D	B4 00	18	С		DB	084H, L, 27 ; Put PF5 on key 27
1109	0F30	B5 00		c		DB	085H,1,40 ; Put PF6 on key 40
1110 1111	0F33 0F36	B6 00 B7 00		c		DB	0B6H, L, 41 ; Put PF7 on key 41
1112	0F39	B9 00		с с		DB	0B7H, L, 42 ; Put PF8 on key 42
1113				c		DB	089H,L,56 ; Put PF10 on key 56
1114	0 F 3C	00 00	00	С		DB	000X,0, 0 ; End of outbound table changes
1115				C			· · · · · · · · · · · · · · · ·
1116 1117				C			
1118	OF 3F				TABLES	FNTIS	
1119						END	

We claim:

1. A personal computer attachment for a display station of the type that communicates with a host computer for the purpose of accessing data and running programs on said host computer, said display station having a display unit and a keyboard, said display unit including display means for displaying an image, buffer means for supplying image data to said display means, keyboard adapter means for receiving keyboard signals from said keyboard, feature bus means for connecting optional features to said display station, and means for providing an interface to a controller external to said display station and for providing a communication link between said buffer means, said keyboard adapter means and said feature bus means, said personal computer attachment comprising:

- a personal computer system unit including a system bus, a microprocessor, memory means and keyboard adapter connected to said system bus, and input/output means connected to said system bus for providing an interface to external devices, said input/output means including a display adapter having a buffer for supplying image data to a display means,
- switch means disposed between said display means and said buffer means in said display station and also connected to said display adapter for selectively supplying image data from said buffer means or said display adapter to said display means, and
- an attachment adapter including an input/output interface, a switch control, a two-way keyboard adapter, and a feature bus adapter, said keyboard being connected to said two-way keyboard adapter, and each of said switch control, two-way keyboard adapter and feature bus adapter communicating with said system bus of said personal computer system unit via said input/output interface, said switch control further being connected to a control input of said switch means, said two-way keyboard adapter further being connected to said keyboard adapter means in said display station and to said keyboard adapter in said personal computer system unit, and said feature bus adapter further being connected to said feature bus means in said display station whereby keystroke signals from said keyboard are transmitted by said two-way keyboard adapter via said input/output interface and system bus to said memory means of said personal computer for interpretation by said microprocessor and then retransmitted back to said two-way keyboard adapter and either to said keyboard adapter means in said display station or to said keyboard adapter in said personal computer system unit and said switch control being responsive to a unique keystroke signal generated by said keyboard to control said switch means.

2. An attachment as recited in claim 1 wherein said input/output means in said personal computer system unit further includes a bulk storage media adapter and a printer adapter, said attachment further including bulk storage media means and printer means connected respectively to said bulk storage media adapter and said printer adapter.

3. An attachment as recited in claim 1 further comprising means responsive to a keyboard command for unloading said buffer means in said display station and transmitting the image data that was stored in said buffer means to said personal computer via said feature bus means and said feature bus adapter.

4. An attachment as recited in claim 1 further comprising means for transmitting data from a host computer to either said buffer means in said display unit or said personal computer system unit via said feature bus means and said feature bus adapter.

5. An attachment as recited in claim 1 further comprising means for transmitting data from said personal computer to said host computer via said feature bus adapter and said feature bus means, said data being generated either by signals from said keyboard or a program run on said personal computer.

6. An attachment as recited in claim 1 wherein said two-way keyboard adapter includes logic that accepts keystroke information from said keyboard and directs it either to said display unit or said personal computer, said logic being powered by said display unit so that if there is a failure in said personal computer, the operator can switch to a host mode of operation using a keystroke sequence.

- 5 7. In a data processing system of the type having a central host computer and a plurality of display stations, each of said display stations having a keyboard and a display unit, said display unit having a display means for displaying an image, first buffer means for supplying
- image data to said display means, first keyboard adapter means for receiving signals from said keyboard, and interface means for providing an interface between said host computer and said buffer means and keyboard adapter means, the improvement comprising a personal
- 15 computer attachment for at least one of said display stations, said personal computer attachment comprising:

microprocessor means, memory means, second keyboard adpater means and input/output means, each

20 connected to a common system bus, said system bus being connected to said interface means and said input/output means including display adapter means having second buffer means for supplying image data to said display means,

25 switch means disposed between said display means and said first and second buffer means for selectively supplying image data from said first or second buffer means to said display means,

two-way keyboard adapter means disposed between
 said keyboard and said first and second keyboard
 adapter means for selectively supplying signals
 from said keyboard to said first or second keyboard
 adapter means, and

control means responsive to an operator input for controlling said switch means and said two-way

8. The personal computer attachment as recited in claim 7 wherein said input/output means further comprises printer means and print command means response.

50 sive to an operator input for selectively printing the content of said first or second buffer means by said printer means.

9. The personal computer attachment as recited in claim 7 wherein said input/output means further com-

55 prises bulk storage means and copy command means responsive to an operator input for selectively copying the content of said first or second buffer means in said bulk storage means.

10. The personal computer attachment as recited in <sup>0</sup> claim 7 further comprising file transfer means for transferring files between said host computer and said personal computer.

11. The personal computer attachment as recited in claim 7 wherein said display means is a monochrome display and said second buffer means is a color graphics adapter means providing at least three color output signals, said switch means including color to grey scale conversion means for converting said color output signals to variable intensity level signals which are supplied to said display means in the personal computer mode.

12. The personal computer attachment as recited in claim 7 wherein said display means is a monochrome display and said second buffer means may be either a monochrome adapter means or a color graphics adapter 5 means, said color graphics means providing at least three color output signals, said switch means including color to grey scale conversion means for converting said color output signals to variable intensity level signals, and display adapter detection means connected to 10 said second buffer means for automatically detecting whether said second buffer means is a monochrome adapter means or a color graphics adapter means and, if said second buffer means is a monochrome adapter means, inhibiting the output of said grey scale conver- 15 sion means to said display means in the personal computer mode.

13. The personal computer attachment as recited in claim 12 wherein said display means requires synchronizing pulses of longer duration than those supplied by 20 station of the type that communicates with a host comeither said monochrome adapter means or said color graphics adapter means, said switch means further including pulse stretching means connectable to either said monochrome adpater means or said color graphics adapter means for generating synchronizing bulses of 25 said image data to said display means, said attachment the required duration for said display means.

14. The personal computer attachment as recited in claim 12 wherein the horizontal frequency signals from said monochrome adapter mseans and said color graphics adapter means are different, further comprising 30 means in said display unit responsive to said display adapter detection means for changing the ramp slope of said display means to accomodate the horizontal frequency signal of the detected adapter means.

15. A personal computer attachment for a display 35 station of the type that communicates with a host computer, said display station having a display unit and a keyboard, said display unit having display means for displaying image data and first buffer means for receiving image data from said host computer and supplying 40 said image data to said display means, said attachment comprising:

a personal computer including a system bus, a microprocessor and memory means connected to said system bus, said memory means including second <sup>45</sup> buffer means for storing image data,

- switch means connected to said display means and to said first buffer means in said display unit and to said second buffer means in said personal computer for selectively supplying image data from said first or second buffer means to said display means,
- bus means for connecting said personal computer system bus to said host computer through said display station for providing a communications link therebetween for selectively transmitting data between said host computer and said personal computer,
- keyboard means for selectively connecting said keyboard to said host computer via said display station or to said personal computer system bus, and
- control means responsive to keyboard inputs for controlling said switch means, said bus means and said keyboard means.

16. A personal computer attachment for a display puter, said display station having a display unit and a keyboard, said display unit having display means for displaying image data and first buffer means for receiving image data from said host computer and supplying comprising:

- a personal computer including a system bus, a microprocessor and memory means connected to said system bus, said memory means including second buffer means for storing image data,
- switch means connected to said display means and to said first buffer means in said display unit and to said second buffer means in said personal computer for selectively supplying image data from said first or second buffer means to said display means,
- bus means for selectively connecting said personal computer system bus to said first buffer means for transmitting data in said first buffer means to said memory means of said personal computer,
- keyboard means for selectively connecting said keyboard to said host computer via said display station or to said personal computer system bus, and
- control means responsive to keyboard inputs for controlling said switch means, said bus means and said keyboard means.