==========

1400 XL Integral Modem

External Reference
Specification

==========

Revision 1 CBP

Joe Miller
20 June 1983

C O M P A N Y C O N F I D E N T I A L

REVISION: A
APPROVED by:

Manager, Systems Development — Date 6/23/83

Manager, 1400 CPU Development — Date 6-23-83

Manager, Product Test — Date 6/23/83

Manager, Quality Assurance — Date

Manager, Publications & Packaging — Date

Product Manager, Marketing — Date 6/28/83

Table of Contents

1.0 Purpose

1.1 Introduction and Product Description

This document is the reference specification for the software interfaces to be used by applications programs and the operating system to control the 1400 Direct-Connect Integral Modem (DCIM).

The 1400 DCIM is an onboard peripheral which connects to the ATARI 1400/1450 by means of the Parallel Bus Interface, and connects to the telephone line by means of an RJ-11C jack. A detailed description of the 1400 DCIM may be found in the 1400 Product Specification.

1.2 Consumer Profile

The 1400 DCIM handler will be used by consumers of telecommunications application software.

1.3 Interface with Other Products

The 1400 DCIM is an input-output peripheral, and performs no actions without instructions from an applications program.

To use the 1400 DCIM from a high-level language, the following types of commands are required:

OPEN

CLOSE

GET (or equivalent READ, INPUT, etc.)

PUT (or equivalent PRINT, WRITE, etc.)

POKE

PEEK

1.4 Family of Products

The 1400 DCIM is an input-output peripheral used in telecommunications applications.

2.0 Applicable Documents

ATARI Personal Computer System Hardware Manual, October 1980, HCD document #CO16555

SURELY External Reference Specification for the Resident Operating System, April 1983, HCD document #<p/n>

ATARI Parallel Bus Interface (PBI) Specifications, HCD document #CO61902

3.0 Requirements

3.1 Interfaces

3.1.1 Physical Requirements

There are no physical requirements for the DCIM within the scope of this specification.

3.1.2 Logical Requirements

The handler shall reside within the address space of the 1400/1450 at D800 through DFFF (hex). It co-resides in this area with the speech device (1400) and the disk firmware (1450XLD). The Parallel Bus server in the operating system shall be responsible for properly multiplexing these devices into the address space.

The handler shall use RAM for input and output buffers, variables for communicating with the user, and internal variables. Internal variables shall be located in the RAM page at D6xx (hex), permanently allocated for use by this handler.

User communication variables are the system IOCB's and the zero page IOCB, the command flag CMCMD, and ring-detect counter NRINGS. IOCB use is described in the ATARI PERSONAL COMPUTER SYSTEM OPERATING SYSTEM USER'S MANUAL, and here in section 3.2.5. Use of CMCMD and NRINGS is described in section 3.2.2.

Input and output buffers can be allocated for the DCIM handler's use by the application program. The two buffers will be allocated in a single contiguous RAM area. The first 20 bytes of this area shall be used for buffer control variables by the handler. Applications use of these 20 bytes is described in section 3.1.4. The remainder of the buffer area is reserved for the buffers. The use of the buffer area is up to the handler. The application program has no defined way of directly accessing data in this area.

3.1.3 Human/Machine Interface

There are no special requirements.

3.2 Functional Requirements

3.2.1 The Six Standard Input/Output Command Interfaces

The applications program accesses the DCIM handler by calling on CIO, which in turn calls the handler. There are six such I/O calls.

OPEN

An OPEN shall be required to be performed before any other modem operations are allowed. Any of the IOCB's may be used. Aux1 specifies the I/O direction: bit 2 for input, bit 3 for output, both for bidirectional. This directionality shall apply only to communications data; initialization parameters may be passed in either direction after any OPEN. Aux2 shall be ignored by the handler.

The device name shall be "T1:". The remainder of the filename description field may be used to pass initialization commands to the handler. Each command shall consist of an ASCII ESC (1B hex, 027 decimal) followed by the ASCII command letter and any necessary parameters. Refer to section 3.2.2 for details of each command. A maximum of 250 characters may be used within the OPEN 'file specification' for the purpose of sending initialization commands. These escape-sequences shall be processed by the handler as commands, regardless of the value of OS variable CMCMD. The filename description field shall be terminated by an EOL (9B hex, 155 decimal)

The command flag (CMCMD) and the telephone ring counter (NRINGS) shall be set to zero by the handler during OPEN.

CLOSE

The CLOSE command shall terminate input/output and release the IOCB. Any data waiting in the output buffer shall be sent first. The buffer area shall then be relinquished and any input data in the buffer shall be lost. When the output buffer is empty, the phone shall be placed on hook, and the modem shall be placed in its power-on state. Refer to section 3.2.5 for details.

[SYSTEM.RESET] shall produce similar results to CLOSE except that the output buffer shall not be sent.

INPUT

Any input or get type operation through CIO will result in the DCIM handler being asked to provide data for the user.

If no data is available, the handler shall wait for a character. There shall be no time-out when waiting for received data. This waiting shall be terminated when either [BREAK] or [SYSTEM.RESET] is pressed. The waiting shall also be terminated if carrier is lost, and error 136 shall be returned.

As characters are received over the telephone line by the modem, they are stored in a buffer area (up to its capacity.) The CIO request for data is then satisfied from this FIFO buffer. See section 3.2.4.

Communications data shall be processed for parity and then translation from ASCII to ATASCII. These operations are described under the appropriate commands in section 3.2.2.

OUTPUT

Any PRINT or PUT type operation to CIO will result in one or more bytes for the handler to hand to the modem.

This data shall be interpreted either as set-up commands and parameters, or as communications data, as controlled by the command flag byte CMCMD. (See section 3.2.2.)

Communications data shall be processed for translation from ATASCII to ASCII, and then parity. These operations are described under the appropriate commands in section 3.2.2.

Once processed for translation and parity, the output bytes shall be placed by the handler into the FIFO buffer area as fast as they come from CIO. If the FIFO buffer fills, the handler shall loop when given another character from CIO until space becomes available in the buffer There is no time-out when the handler is waiting for buffer space. This looping shall be broken when either [BREAK] or [SYSTEM.RESET] is pressed. Loss of carrier shall also break the loop, and returns CIO error 136.

Any time the buffer contains data the modem shall be kept busy sending, until the buffer empties.

STATUS (COMBINED WITH PEEK)

The STATUS command shall cause the handler to place four bytes of status information in the OS status area DVSTAT through DVSTAT+3 (02EA - 02ED). These four bytes shall contain error and status information from both the modem and from the handler. The application program can then PEEK at these four bytes.

The four bytes contain the following bit-encoded data:

DVSTAT+0: Communications error flags

The internal flags corresponding to the status bits in this byte shall be reset upon STATUS request.

Bit 7: Framing error encountered on received data.

Bit 6: Byte-level overrun encountered: The modem lost one or more characters due to slow interrupt handling. This is caused either by slow reading of data by the application when no buffers are in use or incorrectly programmed main CPU.

Bit 5: Parity error encountered on received data.

Bit 4: Receive data buffer overflow: The receive data buffer overflowed as the application program did not input data from the buffer in time. If no receive buffer has been supplied, only error bit 6 shall be set.

Bit 3: Zero (Reserved for expansion)

Bit 2: Zero (Reserved for expansion)

Bit 1: Zero (Reserved for expansion)

Bit 0: Illegal command (recognized by handler). This error shall apply only to the most recent command. This error shall be cleared upon acceptance of a legal command.

DVSTAT+1: Modem status flags

Bit 7: Carrier detect. 1 when carrier present.

Bit 6: Sound enabled. 1 when sound is connected from the telephone to the audio bus.

Bit 5: Loop test enabled. 1 when analog loopback test is active.

Bit 4: Originate/Answer. Zero=originate, 1=answer.

Bit 3: Data/Voice. Zero=data, 1=voice.

Bit 2: POKEY/Speech enabled. 1 when audio generated by POKEY or the speech device is connected to the telephone.

Bit 1: Pulse/DTMF Dialing. Zero=pulse mode, 1=DTMF tones.

Bit 0: Off hook. 1 when phone is off hook.

DVSTAT+2: Address of Communication Buffer. (low byte)

DVSTAT+3: Address of Communication Buffer. (high byte)

If the application has not previously defined a buffer area, this address will point to the handler's internal buffer. The user can determine the locations of INCNT and OUTCNT from this information. Please refer to section 3.2.4 for details on this buffer space.

SPECIAL (XIO)

This call is not supported by the handler. The handler shall return error 92 (hex) 146 (decimal), Function Not Supported.

3.2.2 Parameters Passed as Data Strings

Control functions such as setting translation and parity handling, establishing an applications-supplied buffer area, sending a break signal, dialing the telephone, and so forth, are handled by passing the commands and their parameters to the handler as output data.

There shall be two types of commands: those with parameters and those without. Commands shall start with a two-byte escape-sequence output to the modem handler. This escape-sequence identifies the start of the command and specifies which command it is. If the command has parameter bytes, they shall be sent immediately after the escape-sequence.

In normal operation, CIO output deals only with communication data. The handler shall check the OS RAM area command flag byte CMCMD as it receives each output byte from CIO. (CMCMD is fixed in the OS database at location 0007.) If the flag is zero, the handler shall treat the byte as communications data. If the flag is nonzero, the handler shall treat the byte as a command.

Whenever the handler is not processing a command, output bytes sent with CMCMD nonzero shall be checked for a command and shall not be transmitted. The handler shall start processing a command when: 1) it is not processing a previous command; 2) CMCMD is nonzero; and 3) the output data byte received is ESC (1B hex, 027 decimal). Once a command sequence starts, the handler shall finish the sequence regardless of the state of CMCMD. The number of output and input bytes is constant for any given command (except the DIAL command). When the command bytes have been passed, the handler shall again check CMCMD with each output byte.

Commands start with two output bytes: ESC followed by the command letter. Commands with output parameters contain those two bytes followed by the parameter.

The handler and modem shall work in such a way that communications being received are placed in the input buffer even when commands are being processed; communications data shall not be lost.

Each command is completed when the handler returns from accepting the final byte. Execution of a command may result in an error which shall be returned by the handler to the application when the command completes execution (see section 3.2.3).

Illegal commands (command letter not valid) shall produce error 84 (hex) 132 (decimal), invalid command, and shall not be processed.

The commands are defined as follows:

COMMAND	ACTION
ESC-A-p1-p2	Set Translation Characteristics
ESC-B-p1	Set Baud Rate
ESC-C-p1	Set Parity Characteristics
ESC-D-p1-p2-p3-p4	Establish User-defined I/O Buffers
ESC-E	End-of-Commands
ESC-F	Return Modem Status Information
ESC-G	Begin Auto-answer Monitor
ESC-H	Transmit 500 msec. BREAK signal
ESC-I	Set Originate Mode
ESC-J	Set Answer Mode (for 'manual' answer)
ESC-K-p1-.....-pN	Auto-dial Telephone
ESC-L	Take Telephone 'Off-hook'
ESC-M	Place Phone 'On-hook'
ESC-N	**<reserved for expansion>**
ESC-O	**<reserved for expansion>**
ESC-P	Enable POKEY/Speech Audio to Phone Line
ESC-Q	Disable POKEY/Speech Audio to Phone Line
ESC-R	Enable Modem/Telephone Audio to TV Speaker
ESC-S	Disable Modem/Telephone Audio to TV Speaker
ESC-T	Wait for carrier confirmation
ESC-U-p1-p2	Establish DTMF Dialing Vector
ESC-V	Set Voice Mode (for non-data applications)
ESC-W	Set Analog Loopback Mode
ESC-X	Clear Analog Loopback Mode
ESC-Y	**<reserved for expansion>**
ESC-Z	**<reserved for expansion>**

SET TRANSLATION: (A ASCII, 41 hex, 065 decimal)

Two bytes of parameter shall follow this command. The first specifies the translation options. The second is the will-not-translate substitution character. Translation shall always follow parity checking on input, and shall always precede parity setting on output. The translation options are interpreted as follows:

Bit 7: Shall be ignored by the handler.

Bit 6: When 1, specifies appending linefeed (0A hex, 010 decimal) whenever a carriage return (0D hex, 013 decimal) is transmitted out. This bit shall be ignored if bits 5-4 of this parameter byte are 10 or 11. The carriage return character may come either from an untranslated ATASCII graphics character with code 0D hex, or from the translation of ATASCII EOL (9B hex, 155 decimal) to CR. This translation option shall have no effect on input data.

Bits 5 & 4: When 10 or 11, specifies no translation. This option shall suppress bit 6.

When 00, specifies minimal ("light") translation. On input, carriage return (0D hex, 013 decimal) shall be converted to EOL (9B, 155). All other codes shall be unchanged. On output, EOL shall be converted to CR. No other codes shall be changed.

When 01, specifies major ("heavy") translation. Codes whose interpretations in ATASCII do not correspond to the ASCII interpretation shall be untranslatable. Specifically, codes below Blank (20 hex, 032 decimal) and codes above Vertical Bar (7C hex, 124 decimal) shall be untranslatable, with the exception of Carriage Return and EOL. CR and EOL shall be translated just as in mode 00. On output, untranslatable codes shall be suppressed and not transmitted. On input, untranslatable codes shall be converted to the will-not-translate substitution character supplied by the user.

Bits 3 - 0: Shall be ignored by the handler.

SET BAUD RATE: (B ASCII, 42 hex, 066 decimal)

One byte of parameter shall follow this command. A value of five (5) shall indicate 110 baud communications. A value of ten (10) shall indicate 1200 baud communications. Any other value shall cause the modem to operate at 300 baud (default). The current hardware configuration of the modem will not support the 1200 baud mode. It is expected that field upgrades for the higher speed may be available in the future. At 300 and 1200 baud, the modem shall transmit and receive 8 bits of data with one start bit and one stop bit. At 110, the modem shall transmit/receive 8 bits of data with one start bit and two stop bits.

SET PARITY: (C ASCII, 43 hex, 067 decimal)

One byte of parameter shall follow this command:

Bits 7 - 4: Shall be ignored by the handler.

Bits 3 & 2: Specify input parity operation. The input parity operation shall precede the translation operation. 00 specifies no parity operation. 01 specifies an odd parity check. 10 specifies even parity checking. Should any parity check fail, the parity error status bit shall be set. 11 specifies no parity checking. In all but mode 00, the parity bit shall be cleared after the check. The parity bit is always bit 7 (high-order bit) of the 8-bit form of the character.

Bits 1 & 0: Specify output parity generation. The output parity operation shall follow the translation. If LF is appended to CR, each shall be treated separately for parity. 00 specifies no change to parity bit. 01 specifies odd parity. 10 specifies even parity. 11 specifies forcing the parity bit to 1. The parity bit is always bit 7 (high-order bit) of the 8-bit form of the character.

ESTABLISH BUFFERS: (D ASCII, 44 hex, 068 decimal)

Four bytes of parameter shall follow this command: a buffer address (low, high), followed by a buffer length, NN (low, high). The buffer shall be at least 54 bytes in length. The first twenty bytes shall be used subsequently as a buffer control and parameter area between the handler and the application (refer to the section on buffer use, 3.2.4). The remaining bytes of the buffer area shall be reserved by the handler: 32 bytes shall be used as the output buffer; and the remainder (NN-52) shall be used for the input buffer.

The buffer address and length shall be checked to ensure that the buffer is at least 54 bytes and that it does not wrap around the high end of memory (buffer end must be greater than buffer beginning). If there is an error, the buffer shall not be allocated and the handler shall return error 98 (hex) 152 (decimal), illegal user-supplied buffer.

Once the handler has been informed of the location and size of its buffer area, that area belongs to the handler and should not be touched by the application. Results will be unpredictable should the application attempt to use the buffer while communications are in progress.

END OF COMMANDS: (E ASCII, 45 hex, 069 decimal)

The handler shall set the command flag CMCMD to zero when it receives this command. Thus the user has the option of ending command mode either by POKEing CMCMD to zero or by including this command in the output stream.

STATUS: (F ASCII, 46 hex, 070 decimal)

This command shall perform the same as the CIO STATUS call, that is, DVSTAT through DVSTAT+3 shall be filled in the OS RAM area. (This alternate status call is supplied for those high-level languages lacking the CIO STATUS call.) No data shall be returned in the input stream.

AUTO-ANSWER: (G ASCII, 47 hex, 071 decimal)

This command shall cause the handler to wait for the next ring to occur on the telephone line. When the ring is detected, the modem shall seize the line, pause two seconds for the billing delay specified in Part 68 of the FCC's regulations, and turn on the transmitter to send the answering mode mark signal (2225 Hz.), unless a 'SET VOICE MODE' command was previously issued. If the modem is in VOICE mode, the mark signal will not be transmitted. (See page 18 for details.)

There shall be no time-out when the handler is waiting for the ring signal. The user may terminate this waiting by pressing [BREAK] or [SYSTEM.RESET].

The handler maintains a counter of the number of rings detected, subsequent to the most recent OPEN, in an OS database variable called NRINGS at location 0251 (hex). The application may PEEK or POKE this variable at any time to determine that the phone is ringing. For example, if an application needs to answer the phone on the fifth ring, NRINGS may be set to zero, repeatedly PEEKed until it is equal to four, then the application can issue the 'ESC-G' command to the modem.

Please note that this command does **not** ensure acquisition of carrier with the originating modem. It is the responsibility of the application to determine that carrier is present, either by issuing an 'ESC-T' command or via STATUS, before attempting communications.

BREAK: (H ASCII, 48 hex, 072 decimal)

A break signal shall be transmitted. The handler shall first completely flush any pending output data, then a continuous 'space' is sent for the following 500 ms. If carrier has not been previously established, the handler shall return an EOF error (88 hex, 136 dec.)

SET ORIGINATE: (I ASCII, 49 hex, 073 decimal)

The modem shall be set in originate mode (the default).

SET ANSWER: (J ASCII, 4A hex, 074 decimal)

The modem shall be set in answer mode. This command shall **not** take the telephone 'off-hook', it prepares the modem to use 'answering' frequencies when receiving and transmitting data.

DIAL DIGITS: (K ASCII, 4B hex, 075 decimal)

A string of digits to dial shall follow this command. Each digit shall be held in one byte. The modem ignores the most significant four bits of each digit byte, and interprets the low-order four bits as follows:

0 - 9	Digit to dial, 0 through 9
B	End-of-digits
C	3-second delay
A,D,E,F	Ignored (reserved for future expansion)

The modem will be responsible for inter-digit timing. That is, if the line is not off-hook, the modem will assume this is the first digit of the phone number, take the phone off-hook, wait two seconds (for the dial tone), and dial the digit. After dialing each digit, the modem will wait approximately 800 mseconds before accepting the next digit. The modem will dial each new digit immediately as commanded to do so.

The end-of-digits character shall always be required to terminate each DIAL DIGITS command. The handler shall interpret each character received as a digit for dialing until an 'xB' (hex) character is received. (Note: ATASCII EOL, 9B hex, may be the terminating character.)

It is not required that the user dial a complete telephone number with one use of the DIAL DIGITS command; the user may use more than one DIAL DIGITS command to dial a phone number. There is no signal to the modem of the last digit of the complete phone number. The modem assumes each DIAL DIGITS command may be the last, and monitors for carrier following each. The application program must, by performing STATUS calls or by executing the ESC-T command, determine when carrier has been achieved. It is suggested, if periodic STATUS calls are made, that the application establish a waiting-for-carrier timeout. At the end of this period, if carrier is still not present, the application should issue an 'ON-HOOK' command to release the telephone line.

By default, the 1400 DCIM will use the 'line pulse' method of dialing. The application may, however, instruct the handler to dial digits using the tone method (see 'SET DTMF DIALING'.) Using the latter approach, the handler will pass each DIAL command parameter to the application-supplied tone dial routine except those which contain 'xB' or 'xC' (hex.) The DCIM handler will interpret these as the end-of-digits and 3-second delay characters, respectively. All other codes will be passed to the DTMF routine as eight-bit values via the accumulator. Also, in this DTMF mode, the application-supplied dialing routine shall be responsible for appropriate inter-digit timing.

GO OFF-HOOK: (L ASCII, 4C hex, 076 decimal)

The modem takes the phone off-hook. This command has no effect on the control of telephone sound to the television.

GO ON-HOOK: (M ASCII, 4D hex, 077 decimal)

The modem places the phone on-hook (the default). This command is used to terminate dialing, or to terminate waiting for carrier following dialing. The modem will turn off the telephone sound to the TV.

ENABLE POKEY/SPEECH TO TELEPHONE: (P ASCII, 50 hex, 080 decimal)

Audio signals generated by POKEY or by the onboard speech device are transmitted over the telephone line. This allows a diversity of applications, such as, DTMF dialing sequences, automatic messaging, and call screening. Also, this command shall unconditionally cause the telephone sound to be connected to the TV monitor, RF modulator, serial bus, and parallel bus.

DISABLE POKEY/SPEECH TO TELEPHONE: (Q ASCII, 51 hex, 081 decimal)

Audio generated by POKEY or the speech device shall be disconnected from the phone line. This command shall have no effect on the state of the phone line audio to the TV.

TELEPHONE SOUND ON: (R ASCII, 52 hex, 082 decimal)

Telephone sound is routed by the modem via the parallel bus to the television, monitor, and serial bus. This allows the user to monitor dialing or communications. The state of POKEY/SPEECH audio to the phone line shall be unaffected by this command.

TELEPHONE SOUND OFF: (S ASCII, 53 hex, 083 decimal)

Telephone sound is disconnected from the audio bus. This command shall also cause POKEY/SPEECH audio to be disconnected from the phone line.

WAIT FOR CARRIER: (T ASCII, 54 hex, 084 decimal)

The handler shall wait for confirmation of carrier before returning to the application. This waiting may be prematurely terminated by the user by pressing [BREAK] or [SYSTEM.RESET]. This command shall only be processed if the phone has been taken 'off-hook'. If the phone is not 'off-hook', as a result of a previous ESC-G, ESC-K, or ESC-L command, the handler shall return an 'Invalid Command' error to CIO (84 hex.)

SET DTMF DIALING: (U ASCII, 55 hex, 085 decimal)

Two bytes of parameter shall follow this command. The first byte shall be the low-order address of a user-supplied tone dialing routine. The second byte shall be the high-order address. After issuing this command, all digits to be dialed (see DIAL DIGITS) shall be passed to this routine via the accumulator. Once the required POKEY tones and inter-digit delay have been generated, this external routine must return to the handler via RTS. The DCIM handler shall be responsible for acquiring the phone line, performing 3-second pause increments, and processing the end-of-digits character. Prior to commencing the actual dialing process (via ESC-K), the application must enable POKEY audio to the telephone line by executing an ESC-P command.

SET VOICE MODE: (V ASCII, 56 hex, 086 decimal)

This command causes the handler to be configured for speech or audio transmissions only. In this mode, the modem is actually powered down by the handler and data communications can not take place. This state may be cancelled by executing 'SET ORIGINATE MODE', 'SET ANSWER MODE', either of the analog loopback modes, by performing a CIO CLOSE, or by pressing [SYSTEM.RESET].

SET ANALOG LOOP: (W ASCII, 57 hex, 087 decimal)

The modem shall be placed in analog loopback mode. This command allows an application to execute its own modem test procedures.

CLEAR ANALOG LOOP: (X ASCII, 58 hex, 088 decimal)

The analog loopback mode is turned off. This is the default state.

3.2.3 Error Handling

Three basic exceptional conditions may occur: errors in calling on and controlling the modem, communications errors, and asynchronous interruption ([BREAK] key and [SYSTEM.RESET]). Errors of the first type always occur as a result of an I/O call to the modem handler through CIO, and the error condition shall be signalled to the application via a CIO error return. The second error type includes buffer and modem overrun, framing errors, parity errors, and so on. These errors occur during interrupt processing or inside the modem and shall not be made known to the application with a CIO error. Instead, these errors shall set internal error flags which the application may check with the STATUS command (see section 3.2.1). System reset shall result in closing an open IOCB to the modem (see section 3.2.5).

Certain errors may occur on any I/O call to the handler:

ERROR	FROM
80 hex (128 dec) [BREAK] key I/O abort	handler
85 hex (133 dec) IOCB not open	CIO
86 hex (134 dec) bad IOCB specified	CIO

Other errors are specific to certain I/O calls:

CALL	ERROR	FROM
OPEN	81 (129) IOCB already open	CIO
	84 (132) Invalid command (no I/O direction, or error found in command string)	handler
	96 (159) Port (modem) already open	handler
INPUT	83 (131) Write only	handler
	88 (136) EOF (means carrier loss)	handler
OUTPUT	87 (135) Read only	handler
	88 (136) EOF (means carrier loss)	handler
XIO	92 (146) Function not supported	handler
OUTPUT any command	84 (132) Invalid command (not recognized)	handler
establish buffers	98 (152) Illegal user-supplied buffer	handler

[SYSTEM.RESET] shall result in the closing of the modem IOCB. In this case, any data not yet sent from the output buffer shall not be sent. [SYSTEM.RESET] effects the file closing by "mashing" the system to its reset state. This operation is described in section 3.2.5.

3.2.4 Buffer Use

The user-supplied input/output buffer area allows effective concurrent operation between the modem performing full duplex communications and the 6502 CPU running an application. As the modem receives a communications byte, it causes a parallel bus interface IRQ and interrupts the 6502. The handler fields the interrupt and saves the incoming character in the input buffer. Data held in the buffer is subsequently read by the application via an input call through CIO to the modem handler. Similarly, output data first goes into the output buffer. As the modem completes sending each output byte, it interrupts the 6502 for more data. The handler takes the interrupt and gives the next buffered byte to the modem for transmission.

The modem handler shall be responsible for keeping track of the data in the buffers as the data arrives and is later sent. Status shall be maintained by the handler about how much data each buffer holds at any time, and the application may get this information by PEEKing at it. Each buffer shall be circular (a ring), holding up to one fewer byte than the buffer's size. Each buffer shall be described by five two-byte pointers/counters held at the front of the buffer area. These two 10-byte blocks shall be identical in format, and describe first the input buffer and then the output buffer. The buffer area shall be formatted as follows:

BUFFER+0	INCNT	Number of bytes currently in input buffer
BUFFER+2	INSTRT	Start address of input buffer
BUFFER+4	INEND	Last address of input buffer
BUFFER+6	INHEAD	Input buffer in address
BUFFER+8	INTAIL	Input buffer out address
BUFFER+10	OUTCNT	Count of bytes currently in output buffer
BUFFER+12	OUTSRT	Beginning of output buffer
BUFFER+14	OUTEND	Last address of output buffer
BUFFER+16	OUTHED	Output buffer in address
BUFFER+18	OUTTAL	Output buffer out address
BUFFER+20	OUTBUF	Output buffer, 32 bytes
BUFFER+52	INBUF	Input buffer, NN-52 bytes

All of the control information held in the first 20 bytes of the buffer area are for internal use by the handler, with the exception of the two count fields (INCNT and OUTCNT). No control bytes should be modified by the application; if this rule is violated the results are undefined. The application may PEEK at the count fields at any time to determine the number of bytes being held in the input or output buffers.

The buffer area shall always be formatted the same, even if the modem is being used for input only or output only.

The application program may PEEK at INCNT and OUTCNT to determine the number of bytes currently in the input or output buffer. All other parts of the buffer area are for handler use only; there is no defined way for the application to use any of this data.

The output buffer shall always be 32 bytes in length. As soon as data is put into it, it shall be drained by the handler through the modem at the maximum rate of the modem until the buffer empties. Should the buffer fill, the handler shall loop for space, waiting for the modem, and shall return to the application through CIO only when the byte is placed in the buffer. If the application does not wish to be held, it will have to check the buffer use before making the output request.

The input buffer may be any size above 2 bytes. It may be drained by the application as fast as the application can handle the data, until the buffer empties. When this happens, the handler shall loop waiting for data from the modem. If the application does not wish to be held, it will have checked the buffer use before making the input request. Should the buffer overflow, an error flag shall be set which may be interrogated by STATUS (see section 3.2.1). Input buffer overflow means data is lost: the handler shall retain the data in the buffer and discard the new data from the modem.

If the application does not specify a buffer area via the ESTABLISH BUFFERS command, the handler shall use a small internal buffer formatted in exactly the same fashion as described on page 21. The internal buffers shall provide 32 bytes for the output ring and 32 bytes for the input ring, decimal. The application may perform a STATUS call to determine the location of this buffer area. The same rules apply to this internal buffer as were described for the user-defined buffers.

3.2.5 Handler/Operating System Interface

The handler shall interface to the operating system in the following ways: initialization (cold start), system reset (warm start), break key flag, interrupt handling (IRQ), POKEY timer 4, and CIO (including the "put byte" call by which the application bypasses CIO). There are also a number of fixed bytes/words in the OS RAM area reserved for use by the modem module handler.

Calling interfaces between the application and CIO, and between CIO and any handler, shall be as described in the ATARI PERSONAL COMPUTER SYSTEM OPERATING SYSTEM USER'S MANUAL, chapters 5 and 9. In addition to operating under CIO access, the handler shall support the PUT BYTE entry made by 8K BASIC directly (bypassing CIO). BASIC enters PUT BYTE with the A, X, and Y registers set as CIO sets them. However, the zero page IOCB is not established (except ICAX1Z and ICAX2Z), and any checking normally done by CIO is bypassed and shall be performed by the handler.

The handler shall monitor the system break key flag (BRKKEY) whenever executing any loop which either may continue forever (as when waiting for the modem where there is a possibility the modem may not respond) or whenever the loop may exceed one second. If the [BREAK] key is detected, the handler shall return to the caller with error 80 (hex) (section 3.2.3). The handler/modem interface shall be designed such that such a [BREAK] key abort does not produce any undesirable side effects (preferably, the modem should abort an operation when the [BREAK] key is struck just as the handler aborts).

The 1400 DCIM occupies a slot on the parallel bus, and, as a PBI device, the handler shall adhere to protocols and conventions set forth in the SURELY OPERATING SYSTEM External Reference Specification.

The power-on default conditions shall be as follows. All errors shall be cleared. The telephone shall be placed on hook and no communications are in progress. Telephone to TV sound shall be set off. Modem direction shall be originate. Translation and parity controls shall be set to zero. Analog loop-back test shall be off. No user-defined buffers are established. Dialing shall be 'line-pulsed.' The communication rate shall be 300 baud. POKEY/speech shall not be enabled to the phone line.

The effect of a warm-start ([SYSTEM.RESET]) shall be to close any file open to the DCIM and to reinitialize the handler and modem to the power-on state described above.

3.3 Performance Requirements

The DCIM shall operate on eight bits of data at either 110 or 300 baud. At 110, two stop bits shall be transmitted. At 300 baud, one stop bit shall be generated.

3.4 Design Requirements

The 1450XLD DCIM handler shall reside within an 8K ROM at locations 1000 through 17FF (hex.) The DCIM handler for the 1400XL shall reside at locations 0800 through 0FFF (hex) within a 2K ROM. When selected by the operating system, the handler shall be addressed at locations D800 through DFFF (hex.)

3.5 Packaging Requirements

The 1400 DCIM and handler shall reside within the 1400/1450 plastic.

3.6 Special Requirements

None specified.