DEC STD 111 REV. A

TERMINAL SYNCH-RONIZATION

TITLE: DEC STANDARD FOR TERMINAL SYNCHRONIZATION

ABSTRACT: DC1 and DC3, 21(8) and 23(8) formerly XON and XOFF respectively, are to be used for synchronization of terminal keyboards in the manner described in the standard, DC2 and DC4, 22(8) and 24(8) formerly TAPE and NOT-TAPE respecti·ely, are reserved for future use, likely for synchronization as well.

Reviewed and reaffirmed July 1975

DATE	ECO#	AUTHOR	APPROVED	REV	SEC	PAGES
06-Mar-75	--	Ken Fine	R. Amann	A	--	See Index

SIZE	CODE	NUMBER	REV.
A	DS	EL00111-00	A

INDEX

PAGE REVISION CONTROL

PAGE NO.	PAGE REVISIONS	PAGE NO.	PAGE REVISIONS
1	A		
2	A		
3	A		
4	A		
5	A		
6	A		
7	A		
SEC. REV	-	SEC. REV	
STD REV	A	STD REV	

1.0 INTRODUCTION

This standard is intended to describe the synchronization mechanism for software which interfaces computers to terminal devices; it is both a hardware and a software standard. For purposes of this document, it is necessary to be able to classify any given terminal as either a CRT or a TTY. A terminal is a CRT if the terminal:

1. Operates at speeds greater than 300 buad and does not automatically provide a hard copy history of all the data it has receive; and/or

2. is able to control local peripheral devices via direct command from the host computer (e.g., through ESC sequences or XON/XOFF); and/or

3. provides for certain functions which likely include, but are not limited to, cursor movement (up or down, left, right, or random addressing), scrolling (up or down), or erasing data which has been previously displayed.

(These three classes of features are collectively called "CRT-specific features".)

A terminal is a TTY if none of the CRT-specific features is available.

A piece of software supports a terminal in CRT MODE if the terminal is a CRT and if any of the CRT-specific features is used. A piece of software supports a terminal in TELETYPE REPLACEMENT MODE if the terminal is a TTY, or if the terminal is a CRT and none of the CRT specific features is used. Thus, a piece of software may support a CRT in TELETYPE REPLACEMENT MODE if speed is restricted to 300 baud or less, no peripheral devices local to the terminal are operated via direct command to the terminal, and no display formatting features are used.

2.0 SCOPE

This standard applies to the VT50 project and to all future CRT's designed and built by DIGITAL.

Every operating system

1. which is supported by Software Engineering, AND

2. for which another release is planned, AND

3. which contains a central terminal support facility

must conform to this standard. Conformance is required by the next planned release.

Other software which supports only TTY's or provides only TELETYPE

REPLACEMENT MODE of support for CRT's is not required to conform to this standard. Software which provides CRT MODE support for CRT's must conform to this standard. The software documentation must clearly state which mode of support is provided. In particular, documentation which is intended to define a piece of software, such as a project plan or a functional specification, must state which mode of support is to be provided.

This standard is applicable in full duplex mode of operation only.

This synchronization specification is compatible with the ASR Model Teletype synchronization specification (e.g., when data is being transmitted from the Teletype to the computer through the low speed reader). Thus, software which conforms to this specification will operate correctly on Teletype as well.

3.0 SUMMARY

DC1 and DC3 (21(8) and 23(8), formerly XON and XOFF, respectively), are to be used for synchronization in the manner described in the body of this standard; DC2 and DC4 (22(8) and 24(8), formerly TAPE and NOT-TAPE, respectively), are reserved for future use, likely for synchronization; thus they presently should convey no special meaning to the software. Software which currently uses DC2 and DC4 is doing so in violation of this standard; such software must be changed when these characters are needed by a future terminal.

To the best of our ability to interpret the existing conventions, these usages are consistent with the ANSI standard as intended, if not as generally implemented.

4.0 THE NEED FOR SYNCHRONIZATION

There are several applications where it is desirable for a terminal to have some way to request that the host computer (temporarily) cease sending data; specifically:

1. Screen-Operator Synchronization

   At high baud rates, an operator cannot possibly read and digest information from the host computer before it is scrolled up and off of the screen. At very high baud rates, uncontrolled scrolling reduces the display to a useless blur.

2. Screen-Device Synchronization

   Optional devices (e.g., a hard copy device, a bulk storage device) are typically much slower than the display, so it is likely that an output device would be busy with data about to be overwritten due to a scroll up.

3. "Function" Synchronization

It is possible that certain CRT-specific functions, e.g., erase the entire screen, could take several character times to complete, and, that during processing of these functions, it would be undesirable to receive new data.

On the other hand, there are times when it is desirable for the host computer to have some way to request that a terminal (temporarily) cease sending data. Specifically, a block transmit feature could attempt to unload the screen, typically on the order of 2000 characters, at high speed, potentially overflowing the host computer's buffers.

5.0 THE SOLUTION

1. With Respect to the Terminal Receiving

The terminal will transmit a DC3 (23(8), formerly XOFF) at any time to inform the host computer to cease sending data. The origin of the DC3 could be the keyboard, but more likely would be some logic in the terminal, in which case the terminal is obliged to internally buffer some number of characters (possibly differing with different model terminals) in a buffer, henceforth called the silo. Should the host computer persist in sending data, then the silo will eventually overflow, the data being processed in first-in-first-out (FI/FO) order, potentially causing undesirable side effects, such as a blurred image due to rapid scrolling. It was decided to force the processing of data upon silo overflow, as opposed to ignoring it at either the host computer end or the display end of the silo, so that urgent communications could always be received.

When the terminal is once again ready to receive, signalled either by (special) keystroke or some event having completed, then the silo is emptied and processed in FI/FO order, and DC1 (21(8), formerly XON) is transmitted to the host. DC1 is transmitted when the silo is actually empty, in order to allow for synchronization during the processing of those characters temporarily buffered in the silo.

The terminal documentation must clearly and precisely show the number of characters (and corresponding minimum real time intervals) which can be buffered in the silo at the various send/receive buad rates. The host software should be designed to react to DC3 as soon as it possibly can. It is suggested that the terminal contain some sort of signal that synchronization is in progress (e.g., a DELAY lamp), so as to minimize operator anxiety and attendant physical reaction.

2. With Respect to the Terminal Sending

The host computer will transmit a DC3 at any time to inform the terminal to cease sending data. The terminal (temporarily) suspends transmission upon receipt of the DC3 character. Given the send/receive baud rate, the temporary buffer size in the host computer required to effect this synchronization can be easily calculated. Should the terminal malfunction and not respond to the DC3, it can be expected that some undesirable side effect, such as buffer overflow in the host, will occur. The host also must not ignore incoming data, so that it, too, can receive urgent communications, even though it has sent a DC3 to the terminal. It should be expected that data will be lost when this happens.

The host computer will transmit a DC1 when it once again is ready to receive. The terminal responds by continuing the transmission at the point suspension occurred.

6.0 SOLUTIONS WHICH WERE REJECTED

1. Make the screen memory bigger, thus buffering more data in the terminal.

   This was rejected because it is too expensive and cannot possibly satisfy the requirements of an operator who wishes to examine data on the screen for long periods.

2. Somehow, cause the terminal to interact with the "ready bit" in the interface.

   This was rejected because such a terminal could not interface to the host computer over phone lines, given our present terminal interfaces.

ADVANTAGES IN THIS METHOD

1. The synchronization can be implemented in the terminal's micro-program, in which case, it is "free".

2. The host software is independent of terminal characteristics with respect to synchronization.

3. So long as it is very clear as to who is sending and who is receiving, this synchronization will function properly even with binary or image data; the receiving end simply sends the transmitting end a synchronization charter. This simple approach is certainly not intended to preclude incorporation of a more comprehensive communications protocol.

State Diagram

The table below depicts the meaning which the terminal gives the character (sent by the host computer), given the state.

Char/State	Block Transmit in Progress	Block Transmit Suspended	Neither
DC1	Ignore	Continue	Ignore
DC3	Suspend	Ignore	Ignore