bcc	title: THE COMMUNICATIONS SYSTEM - PHASE II	prefix/class-number.revision: CS /S-24

checked	authors	approval date	revision	date
checked: Larry L Barnes	Paul Heckel	1/5/70		
approved		classification: Specification		
		distribution: Company Private	pages: 21	

ABSTRACT and CONTENTS

This document describes the conventions used by the CHIO and the Remote Concentrator for communication with each other.

Discussed in detail are: the encoding of characters and device numbers on the communication line, the multiplexing algorithms, and Loading the Remote Concentrator. An overview of the Communications System is given, and the functions of all of the control characters listed.

TABLE OF CONTENTS

Introduction

The CPU/CHIO INTERFACE document (FOO/S-28) and the CHIO IMPLEMENTATION PHASE I document (IHTWD/W-38) describe the Phase I CHIO and a substantial part of the Phase II CHIO. The bit scanning subroutines will (probably) be dropped; but all other subroutines will remain.

A detailed description of the Remote Concentrator is given in RC/S-23. The ERROR FREE-COMMUNICATION LINE (FAROUT/W-40), AN EFFICIENT MULTIPLEXING ALGORITHM (ATINFIC/W-37), and LOCAL ECHOING IN THE COMMUNICATIONS SYSTEM (OHWOW/W-41) are also referenced by this document.

This document considers the problem of transmitting information from the CHIO to the Remote Concentrator on the Error-Free Communication Line. It assumes the interface described in EFCL, specifically, characters are put into a buffer called FRED, and taken out of a buffer called HARRY in the other computer. The Valparaiso Observation described in that document will be used.

Overview

The communications system consists of a CHIO and several Remote Concentrators, each connected to the CHIO by a 4800-baud leased line. The CHIO is a microprocessor that can reference the M-1 fast memory. Its function is to take characters output by a CPU process and store them until it can multiplex them on a 4800-baud line for transmission to a Remote Concentrator. The Remote Concentrator demultiplexes the characters and outputs them on the appropriate device. Similarly, characters input from devices are received by the Remote Concentrator, multiplexed on the 4800-baud line and sent to the CHIO. The CHIO demultiplexes them and stores them until a CPU process inputs them.

The high speed line is used to transmit the characters from the CHIO to the Remote Concentrator and from the Remote Concentrator to the CHIO. Conventions have been defined for encoding information on this line. Error detection and recovery procedures have also been defined.

Let us look first at the high speed line and see what type of information is encoded on it. If one looks at the line initially, (in time) one notices that SYN characters are being transmitted on it. Then there appears a series of three or more NULL characters (0). When these NULLS appear on the line (and for this reason this is the only place that the NULL characters may appear) the hardware interface resynchronizes on the SYN character that follows the last NULL. This procedure will be followed whenever an error occurs, or when transmission is initiated. This procedure ensures that the transmitter and receiver are in synchronization, so that the receiver is not reading as one character the last half of one character and the first half of the next character.

The line has now been synchronized. The first character delivered to the receiving computer is the SYN character. Following this character is a short message of about 3 characters which gives some information to the receiver, (the block number that follows). This is followed by an end-of-block EOB and two checksum characters. If these two characters agree with the calculated checksum, the Error resumption function described in EFCL are performed. The block number will be correctly set.

Following this initial message to the RC comes a string of characters, each one D (=13 for a 4800-baud line) characters long. T and D (defined in EFCL) are kept in the CHIO's Remote Concentrator table so that T may vary from one Remote Concentrator to another, depending on the speed of the line and its distance. Each of these strings is followed by check characters (described in EFCL). This is a block of information. For reasons that we shall see, the block is a very important concept. One notices that there are two extra characters in the block, meaning that the channel has lost 2/T of its bandwidth (usually about 14%). The second observation is that if the check characters are removed from the sequence of characters (which is what the EFCL input subroutine for the receiving computer does), the resulting string transmits the multiplexing information, which is then decoded. The EFCL subroutines insert and then delete the checksum characters, but these two subroutines cancel each other out.

The third important fact about the blocking of characters is that each block has a number associated with it (from 0 to 127). Each time a block is sent (received), the output (input) block number associated with it is incremented (mod 128). This block number is available for various routines to make use of; we will see how later.

The error detection and recovery system is discussed in FAROUT/W-40 in detail. We have a basic idea of how the error subroutines work, and we can say that in a certain sense we have an error-free communications line between the two computers. We must now encode the information on this error-free communications line. We have saved 32 characters as control characters to use for such purposes. If a character in the range of 0 to 37B is to be transmitted, it is transmitted as a SHIFT1 followed by 40B plus the character to be transmitted. For transmitting characters to the CHIO we will use one method of encoding the multiplexing information and for transmitting characters to the Remote Concentrator we will use another method. This is because the input and output have different properties. There is less input than output, but input is less regular because most input is provided by people who are slow and irregular and most output is provided by computers which are more verbose but can supply characters fast enough to keep a teletype busy.

Input is encoded quite simply. First, a character is sent that indicates the device number, then the character that was input by that device,etc. This wastes precisely half of the bandwidth (on top of the 14% already wasted), but this is a simple straightforward algorithm which should work quite well. Because there might be some medium speed devices connected to the computer, as well as teletypes, the algorithm is a little more complicated. In front of the stream of characters just described (alternating device number, character) a control character that tells us that we are doing just that (ADC) is inserted. If characters start to come in from a medium speed device, we now put a block stream character (BST) and follow it with the device number. All the characters in the string up to the next BST or ADC will be presumed to come from that device.

Output, you remember, is a different story. There is more of it, so we need an efficient method of encoding the characters on the line. This method is described elsewhere (see ATINFIC/W-37). It requires cycling through the lines that are active (have output going to them) in a prescribed manner. The Remote Concentrator scans the active lines in the same way. The only information that must be encoded besides the characters for the device is information that indicates a new device has been activated, or that an active device has been deactivated. This adds three characters per output stream. It is expected that the total wasted bandwidth due to this will not exceed 20% on average. It might be as low as 10%. Because of the nature of the Meta Algorithm for synchronizing the output line scan, it would require an undetected error on the communications line or a hardware error someplace to cause a phase error (output going to the wrong teletypes).

We now have a basic idea of what the communications conventions are. We haven't considered some other problems. Given in detail elsewhere is the control character that can be used to load the Remote Concentrator (LRC), or to request the Remote Concentrator to send part of its core to the CPU. This is normally used only for initial loading of the Remote Concentrator or for some debugging situations. The normal method is to use the Control Task, but this requires that the CT exist.

Each of the operations not discussed in another document will now be dealt with in detail.

Output Multiplexing

Because the CHIO uses the Meta Algorithm for output line scanning, the CHIO and the Remote Concentrator use the same scanning algorithms. Both computers maintain a character count (CII), which is used to measure time. Every time a character is put into FRED, CII is decremented.

Time is divided into (semi-) arbitrary periods called _intervals_. During each interval at least 100 milliseconds will pass. On a 4800 baud line an interval contains 52 information characters and 8 check characters (4 blocks), making 60 characters per 100 milliseconds. On a 2400-baud line there would be 26 information characters and 4 check characters (2 blocks) in an interval.

At the start of each interval CII is initialized to ICII (52 for a 4800-baud line). The output scanner linearly searches the active devices. For each device the scanner sends as many characters as the device can dispose of in 100 ms, up to a maximum of 3. If CII is still positive, when all of the devices have been scanned, the scanner will send characters to the medium speed devices (printers, etc.). When CII becomes negative, no more characters are sent to the high speed output devices. If there are no more devices to send characters to, then the character 40B is sent. The null line is the one assumed to be selected to receive this trash. At the beginning of each cycle a CHS is sent to check that both ends of the line are in agreement about where the beginning of the cycle is. It should be pointed out that the only way that CII can ever get below -1 is for low speed devices alone to overload the line (slowing all devices equally).

The Remote Concentrator will have a 5 character buffer for each device. This will accommodate devices of any speed up to 20 characters per second, and also 30 CPS devices. The way that this number is calculated is rather subtle so it is described here. First, suppose at the very end of one interval a burst of characters is sent to device X, and then at the beginning of the next interval a second burst is sent to X. (This is quite unlikely but possible if all but one of the teletypes is turned off suddenly). The burst length is 3 characters for a 30-character-per-second device so we would need 6 characters that have to be buffered. One of the characters will go into the output buffer (which is not included in the buffer count), decreasing the count by 1.

But things can be worse than that. If the number of characters a device can have in an interval is not integral, then it is possible that an extra character's worth of buffering is required. (If you don't believe this, consider a device that sends .9 characters per interval and assume that the character is sent at the end of the first interval, and at the beginning of all other intervals).

The device table gives the number of characters a device in an interval may receive as a fractional number. (INCI is the integer part, INCIF is the fractional part). INCI.INCIF is 1 for a Model 35, 1.5 for a Model 37, and 1.37 for a 2741. The algorithm for determining the number of characters to send to a device in an interval is:

```
ONCI ← NCI
NCI.NCIF ← .NCIF + INCI.INCIF
```

NCI is the number of characters that will actually be sent in this interval. Note that this calculation is done once for each device type in each interval; not once for each device!

These variables are kept in the device table. A problem presents itself, however, if this algorithm is followed and there are say, 40 model 37 teletypes. Then odd intervals would have 80 characters, and even intervals would have 60 characters (including 20 sent to a null device). ONCI, unused till now, is used as the interval count for devices whose CPU input line number divided by two is odd, NCI is used if it is even. This does not insure perfection; but about half of the two character pairs are sent in one cycle, half in the next.

The format of the CHIO's Remote Concentrator table is shown in the Appendix.‡ Part of the table contains the local number to CPU line number table. Each active device is indicated by having the A bit set. The Remote Concentrator also has a copy of this table. Both scanners search the table linearly looking for entries with the A bit set.

The V bit, which the scanner does not see, is set to activate a line. The Find to Insert algorithm checks to see if the next line that has either an A or V bit set has the V bit set. If there is such a line, then the Insert New Line (INL) character is sent followed by the line number, and the Insert New Line Subroutine (INLS) is called. INLS zero the V bit and sets the A bit.

Deleting a line is similar. If the line selected by the Meta Algorithm for Transmit attempts to read a character for an output line and finds that the line is empty, it sends the Delete Old Line character (DOL). DOLS, the Delete Old Line Subroutine, resets the A bit, preventing that line from being selected until it is reactivated by a call of INLS.

---

‡ to be published later.

Input Multiplexing

There are two control characters for multiplexing input, BST (Burst String), and ADC (Alternate Device and Character). If a BST is sent, it is followed by the device number and then a string of characters for the selected device. The string is terminated by a second BST or an ADC control character. This method is used to encode medium speed devices.

Low speed input devices are encoded by sending an ADC and then following it with several character pairs. The first character in the pair is the local line number; and the second is the character for the device. This sequence of pairs is stopped by a BST or an ADC.

The scanning algorithm is quite simple. The low speed input devices are scanned to find a device that has an input character; this character is multiplexed in ADC mode. If there are no low speed devices with input characters, then medium speed devices are scanned. The output to such a device is encoded as a BST followed by the high speed input device number, followed by the characters. If there is no input device with a character for the CHIO, a succession of ADCs is sent since some input character must be sent. This is consistent with the definition of ADC.

Each of the low speed input devices has five one-character buffers. These are loaded by the input bit scanner. Local echoing is done when characters are fed into the error-free communication line. These five buffers offer some slop that is used to absorb a temporary burst of characters or to hold characters when the communications line fails. Because characters are not echoed until they are fed into EFCL, a character which is lost because the buffer is full is not echoed. Thus the user, if he sees a character, knows that it has been input to the system, and if he doesn't see it, he thinks he didn't hit the key hard enough.

This interface is important because a clear understanding of it is necessary to properly understand the communications system. The use of the buffers to absorb a temporary burst of characters is obvious. Even though the average rate at which characters come into the system is small enough to prevent the 4800-baud line from being fully utilized, occasionally there will be more characters input in the span of say 1/10 of a second, than the Input Multiplexor can handle.

The second use of these buffers is to hold some characters when an error on the 4800-baud line occurs. The line will hold enough characters for one-half second's worth of information for all typed input. If any characters have to be dropped, the input is replaced with a SUBI and not echoed.

The third reason allows the specified implementation of Local Echo Resumption (OHWOW). It is the reason that the buffering is done in device buffers rather than elsewhere.

The CHIO has a straightforward algorithm to process this string. If it finds an ADC, it sets a short mode switch (SMS). It resets SMS if it finds a BST. In both cases it sets an Expect device Number Switch (EDNS). When a non-control character is input, and EDNS is on, the character input specifies the local device number by being saved in SPL. EDNS will then be reset.

If EDNS is reset when a character is input, the character is put in the line specified by SPL. EDNS is then set if SMS is on. Thus in long mode a string will be sent to the selected line, while in short mode only one character will be sent to the selected line.

Remote Concentrator Initialization and Loading

We understand how the communications system works when it is operating. The Remote Concentrator may be initialized by sending loading information in Messages or Acknowledgments. (See EFCL). The LRC (Load Remote Concentrator) is specified in the section on Control Characters.

This control character is interpreted by the output line multiplexor, so it can appear anyplace in the multiplexor string. Loading and storing the Remote Concentrator by sending messages to the Control Task is also possible. This is discussed in the Remote Concentrator document (RC).

Control Characters

Characters followed by an asterisk (*) are meaningful to a user program and may be received by the user, transmitted by the user, or both.

NULL:* (ØØ) The NULL character, which is always legal, may be inserted into or deleted from any string at will. It is deleted by the communications system whenever it is found. NULL is never transmitted on the multiplexor lines (except as part of the resynching procedure).

SYN: Synchronous Idle is transmitted on the multiplexor line if no character has been supplied by the computer. It may be inserted at any place and will not change the message, as the SYN character is not counted as part of a block or included in the checksum. SYN is also used as part of the line resynching procedure by the Communications System.

SHIFT1:* The character following SHIFT1 is taken as MOD 32. This character must be used to transmit one of the first 32 characters.

ADC: Alternate Device Character: This character sets the mode to 'single character.' It is followed alternately by a device number and a character until an ADC or BST is encountered. ADC is used for input multiplexing only.

REC: Resume Echo Character: This character is used by the CHIO to tell the Remote Concentrator to resume local echoing. It is followed by the resume echo block number. It and the character following it are counted as one character by the scanning algorithm. This character will also do the DOL function whether or not the attempt to resume echoing succeeds.

LRC:* Load Remote Concentrator: This character is used for loading and dumping the Remote Concentrator. It is sent by a (protected) CPU program to the Remote Concentrator.

1) The character following LRC is interpreted as a type code consisting of three bits: LB, DB, ID.
2) If the ID (Ignore Devices) bit is set, then the Remote Concentrator ignores all external devices until another such message with the ID bit reset is received.
3) If LB (Load Bit) is set, the command is interpreted as a load Remote Concentrator command. The next character is a word count (less than 224), and the two characters after that specify a loading address.
4) If DB (Dump Bit) is set, the next three characters specify, in the same way, the portion of core to be dumped. If both LB and DB are set, the effect

is to echo back the characters stored in core - thus providing error checking.

5) When a dump is returned to the CHIO because DB was set, it is preceded by the same four characters that were sent to the Remote Concentrator. This provides error checking on the loading address.

MAF: Message or Acknowledgment Follows: This character follows the SYN in a resynchronization. It is used to indicate that the characters following it transmit information needed by EFCL, such as a request to retransmit a block. The following messages are defined:

1) R n: Retransmit messages starting with block n. This is a request to retransmit all blocks starting with the indicated block because an error has occurred. (This is a Message).

2) F n: Following block is: This specifies the next block number of the block following this block. This is an Acknowledgment.

3) Z n: Zero block count: This specifies the number to which the input block count should be set.

CHK: Check Character: This character indicates that the next two characters contain the checksum for the characters that preceded it, starting after the SYN.

This is only used in Messages and Acknowledgments.

SNULL:* Special Null: SNULL guarantees a delay of 1/10 of a second at the output device.

SUB:* Substitute is substituted for a character that becomes munged, gronked, lost, or has fallen on any other evil that can occur to characters. If a string of characters is lost, then an equal length string of SUBs is sent. Thus the user knows the number of missing characters.

SUBI:* Substitute Indefinite: This character is used to indicate that an indefinite number of substitutions has taken place. This character is usually generated by the Remote Concentrator if it must encode more than 8 substitutions.

SUBn: Make N substitutions $2 \leq n \leq 8$: Because SUBn can encode several SUB characters efficiently, it will be used by the input multiplexor if the 4800-baud line is saturated. A SUBn will never be passed on to a user; a string of SUBs will be constructed from the SUBn character in the CHIO.

CBREAK: Break in Line: This character is sent if a break in the input line is detected by the bit scanner.

DOL: Delete Old Line: The output line receiving characters is deleted from the list of active output lines. Note that REC also performs this function.

INL: Insert New Line: The character following the INL specifies the local line number to be activated. This and DOL are used by the Meta Algorithm for output multiplexing. The characters are not counted as going to any line.

BST: Burst String: The character immediately following the BST specifies the local input line number that the characters next following until the next BST or ADC come from.

ORC: Other Remote Concentrator: This control character indicates that the block following it goes to the other Remote Concentrator in half-fast mode. It sets a switch in the Remote Concentrator so that MXIN will send the next block to the 2400-baud line.

SRT: Start Remote Concentrator Task: The character following SRT is decoded to determine whether one of 32 possible Remote Concentrator tasks should be turned on or off. The low order 5 bits represent the task number. The second bit, if on, starts the task, if off stops the task.

CHS: Check Synchronization. This character is sent by the Meta Algorithm for transmission each time GNL gets back to the beginning of its scan. The Meta Algorithm for Receive will indicate an error if CHS is received at the wrong time or not received at the right time, and the line scan will be reset.

TAG: Sent by a CPU program TAG has the following useful properties. 1) local echo mode (if on) is turned off and 2) it is echoed back to the CPU program by the Remote Concentrator.