SYNCHRONOUS COMMUNICATION PROTOCOL

FOR THE

BCC 500/ELF DATA CONNECTION

1.0 GENERAL

The line will be operated with 8-bit bytes. Although the line is synchronous, it will be used by the software at either end in a pseudo-asynchronous fashion. That is, the line interfaces will be operated to ignore sync characters (SYNC). When no information is to be passed or when the transmitting function at either end of the line is late in producing the next byte for transmission, a SYNC (026B) will be sent automatically by the sending interface. The same character on arrival at the receiving interface will not produce a service interrupt and so will be ignored.

Since SYNC is ignored, the combination 026B may not be sent on the line, and the line cannot be used for transmitting arbitrary information. To overcome this limitation the protocol is hereby established whereby the special byte DLE (020B) will precede the transmission of an 066 and will act as an escape, signifying that the 066 must have 040B subtracted from it before further interpretation.

The dedication of DLE for this purpose means that it, too, cannot be sent as information on the line. But the same convention may be used to send the value 020: that is, the combination 020 060. To summarize, any 8-bit byte may be sent on the line at the convenience of the sender except SYNC and DLE. Of these, SYNC is reserved for transmission during idle time and DLE is used to indicate that the following byte requires a subtraction of 040B. The byte 026 thus appears on the line as 020 066, and 020 appears as 020 060.

2.0 CHARACTER FRAMING SYNCHRONIZATION

The line is short and does not involve modems and transmission facilities. The expected error rate will therefore be small. In particular, the loss of clocks or other untoward events which might cause loss of character sync will be most infrequent. The action of line synchronization will be undertaken seldom -- usually only when restarting the machine at either end of the line.

The actual act of synchronization is initiated by placing a receiver in "search mode." In this mode the receiver ceases to function as such, but compares at each new bit time the last 8 bits received with the sync character. When at most two such characters are received successively, character framing is established and the receiver reverts to normal activity.

The establishment of sync does require the cooperation of the other end of the line. To obtain this cooperation in turn requires a signalling scheme. By convention, then, when either end of the line is initialized or otherwise determines that it has lost character sync, the following procedure will be employed:

Remarks: That end of the line first noticing loss of sync is called syncee; it is actually the initiator of the procedure. The other end of the line is, of course, the syncer and must cooperate with the syncee in order to establish sync. There is a chance that both ends of the line will see loss of sync simultaneously and that either end will think itself to be syncee. This gives rise to a possible race condition. The change of such a race is reduced by having each end wait a different amount of time before initiating action. The race will thus occur with small odds and in any event is non-critical. Note that in the scheme below both directions of transmission are always synchronized.

Procedure:

1. Syncee immediately suspends I/O operations, sets request to send, counts to 1000, and sends continuous syncs. Syncee then verifies at last moment that secondary receive

is low and issues secondary transmit, waiting for secondary receive as a response.

2. Syncer sees secondary receive and suspends I/O operations. Syncer resyncs its receiver, sends request to send, and sends continuous syncs. Syncer then raises secondary transmit and waits for loss of secondary receive.

3. Syncee syncs its receiver, prepares to receive information, and lowers secondary transmit. It waits for loss of secondary receive.

4. Syncer prepares to receive information, drops secondary transmit, and resumes transmissions.

5. Syncee resumes transmissions.

It is important that the syncee check its own secondary receive immediately before requesting a resync. If it finds the line high, then it must instead play the role of syncer.

The request/clear to send signals were designed for signalling with half-duplex mode situations. These signals are not especially useful for our intended use of the line. However, the signals do affect the operation of the line interfaces and so should be kept permanently on. If it is useful information at any time, however, the lack of a clear to send signal at either end of the line is clear evidence of the other end's being not active. The converse is, of course, not necessarily true.

3.0 INITIATING SYNC IN THE 500

The program to initiate and establish sync on the line at the 500 end will reside in CTP code in the CHIO. The CTP is a convenient processor for such code because it is actually a part of the CHIO and has ready access to the CHIO data structures and I/O hardware. The CTP will run a little resident program containing miscellaneous tasks relating to various devices connected to the CHIO. The synchronization task is one of these.

Synchronization will be initiated by a user process (the FTP process) upon initiation or whenever the process believes the line to be out of sync. The FTP process communicates with the synchronization task in the CTP through a special pseudo-terminal line, i.e., a CHIO line not associated with any real line hardware (just entries in the various bit tables). The FTP "acquires" this line and then outputs a command on it. The CTP task periodically reads the line and thus picks up the command. Responses are returned through the other half of the line.

The exact commands and their associated responses and their formats will be spelled out in a separate document.

4.0 ERROR DETECTION AND RECOVERY PROCEDURES

Specification S-1 gives the basic data format on the line and describes the flow control. This approach was developed for use with the DJ-11 asynchronous line interface which may run at rates up to 9600 bps. The synchronous line will run at rates up to 38.4 Kbps but may be more subject to error, especially if a clock is missed. This section describes a modification (actually an enhancement) of the data format/flow control outlines in S-1.

Whereas S-1 mentions four 30-character buffers, the synchronous line will utilize a greater number of buffers, each of which is larger than 30 characters. The actual numbers will have to be determined by negotiation with ELF experts, but in order to maintain a transmission rate anywhere near the 38.4 Kpbs rate a total of at least two second's worth of transmission must be buffered. That is, at 4800 characters/sec, about 10K bytes. Instead of a 16-bit binary count of the number of characters in a buffer-load preceding the buffer on the line, each buffer on the synchronous line will be preceded by a 16-bit header word and followed by a 16-bit cyclic check sum.

The header word will consist of an 8-bit binary character count (as before except now the first of the two bytes) and an 8-bit binary buffer I.D. number. During transmissions the number of each buffer will begin with 128 and increase up to 255 before reverting to 128 and repeating.

The check sum will be transmitted as two 8-bit binary bytes, more significant byte first. The check sum will be formed as follows: Each new character placed in a buffer will be EORed while right-justified in a 16-bit register with the preceding value of the check sum. Then this new value will be left cycled 5 places to form the current value of the check sum. After the appropriate n characters the result is the buffer checksum.

Each buffer transmitted will require a positive acknowledgment and an I.D. check from the receiving end. Flow control will be implicit in these acknowledgments.

As in S-1, three 200 (octal) bytes will be sent by the receiving end at the beginning of data transfer to indicate start of transmission. Similarly at the end of transmission a single 203 byte will be sent as verification that the receiving DTP has terminated. If for any reason the receiving DTP terminates prematurely, a single 204 byte will be sent as notification. Appropriate messages will simultaneously flow on the control line. The 201 and 202 bytes will, however, each be followed by an I.D. number and, while still serving to perform flow control, will also act as acknowledgment messages.

201 will mean positive acknowledgment for buffer n: buffer received correctly and there is now room for another buffer load. 202 will mean positive acknowledgment for buffer n known to be the last buffer; DTP activity will therefore cease with this buffer. 205 will be a negative acknowledgment for buffer n; buffer received in error; re-send, as there is now room to receive.

When a 205 is received, or when some sequence errors are noted the recovery procedure will work as follows. The procedure is designed for simplicity and is based on the expected low error rate. We discuss check sum errors and sequence errors separately. First, check sum errors.

The sending process will retain a copy of the transmitted buffer load until the acknowledgment has been received. At that time if it is negative, the sending DTP will reject and flush from its buffer space all buffers received after a defective one until that buffer reappears. No acknowledgments will be sent for the rejected buffers.