Specification of the HASP transparent multileaving protocol as accepted at ERCC for communications in the regional network

J.I. Davies
E.R.C.C
7.6.73

Contents

		Page
1.	Introduction	1
2.	References	2
3.	Line Handler	3
4.	Line Manager	4
5.	Line User	6
6.	Conclusion	7

Introduction

This specification is an attempt to clarify the requirements which a data terminal must satisfy in order to communicate with a HASP multileaving workstation and/or a HASP mainframe.

The specification is at three levels

(i) Physical character and block transmission and reception mechanism
(line handler)

(ii) Physical data exchange mechanism
(line manager)

(iii) Logical data exchange mechanism
(line user)

The gross functions and interfaces of these three entities are illustrated in Fig. 1.

The present version of the specification concentrates upon level (ii) since level (i) is essentially a subset of the IBM binary synchronous protocol and level (iii) is largely specific to the particular type of user terminal.

The specification is deliberately in terms of a single terminal and the total system may be assumed to be symmetrical except where otherwise stated.

As an overall description of Multileaving there exists none better than that of the HASP manual (ref (iii) first paragraph)

> "In a gross sense, Multileaving can be defined as the fully synchronised, pseudo-simultaneous, bi-directional transmission of a variable number of data streams between two or more computers using binary synchronous communications facilities".

Throughout this specification an attempt has been made to exclude any implementation dependent information except for purposes of clarification. Such information will be found enclosed by square brackets.

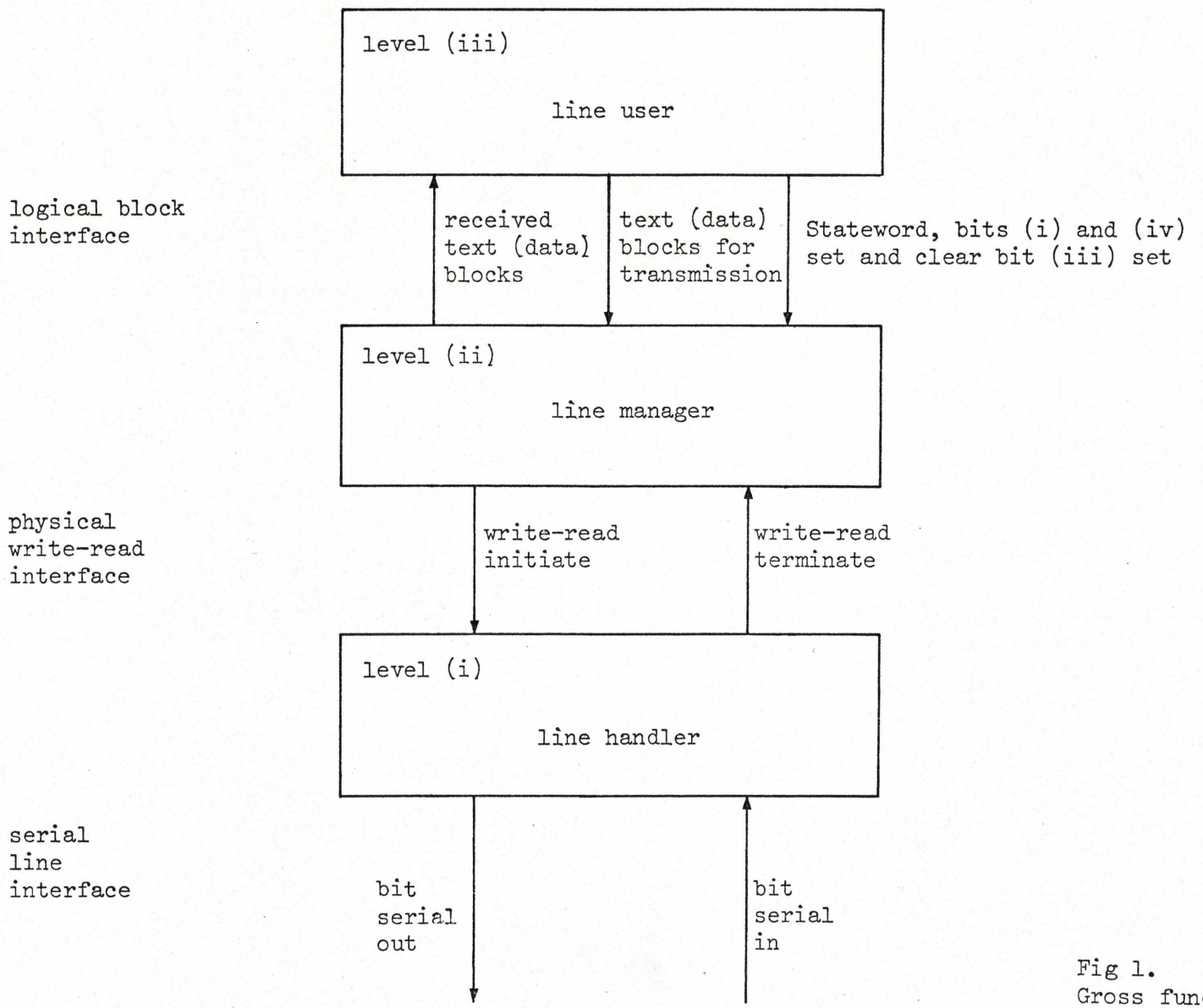


Fig 1.
Gross functions and
interfaces of a terminal.

References

(i) IBM Systems Reference Library, form A22-6864
'2701 Data Adapter Unit Component Description'

This IBM hardware is the normal means of interfacing a 360/370 mainframe to a binary synchronous communications line. The document contains information relevant largely to level (i) of the present specification.
[The IBM 3705 communications controller, which will replace this hardware at ERCC, will initially run in an emulation mode allowing functionally identical operation at the HASP and communications line interfaces].

(ii) IBM document, form GA27-3004
'General Information - Binary Synchronous Communications'

This is an IBM document covering the binary synchronous communicatons system of which HASP Multileaving is a subset. The document is relevant mainly to level (i) of the specification.

(iii) IBM Contributed Program Library - Houston Automatic Spooling Priority System with Remote Job Entry.
Program Description, Section 12.11

This section of the HASP manual describes the internal constituents of a Multileaving block including the FCS field which contains both the WABT flag and the individual stream control bits. It is relevant to levels (ii) and (iii) of this specification.

(iv) 'Proposed Specification for a Wideband data port for a Modular One computer' - ERCC document

This describes a hardware system for handling a transparent Multileaving line at the block transfer level. This document and subsequent (more detailed) specifications will necessarily give a precise definition of the protocol at the physical character/block level.

Physical character and block transmission and reception mechanism (line handler)

This level receives write-read commands from level (ii) and handles communication down to the bit level on the line. The commands accepted are:-

Write (ACKO) - Read
Write (NAK) - Read
Write (Block) - Read

and the following Write-Read terminations are passed back to level (ii)

Block with correct CRC (good block)
ACKO
NAK
Invalid block (ie character sync acquired but sync patterns not followed by a valid character sequence)
Receive timeout (ie no character sync seen within a specified interval)*.
block with incorrect CRC

The format of blocks is defined in refs. (ii) and (iii) and the ACKO and NAK sequences are defined in refs. (i) and (ii).

The WAIT mechanism demanded by the line manager is logically independent of the write-read functions of the handler but may conveniently be included here since it involves access to a real time clock mechanism similar to that required for the receive timeout and since the WAIT and write-read functions are mutually exclusive. The WAIT interval must be significantly less than the timeout interval†.

This specification covers only those versions of multileaving which allow full EBCDIC transparency since this is the recommended standard for the network.

* The interval at present implemented is 3 seconds.

†[ERCC implementations use a one second wait interval.]

Physical data exchange mechanism
(line manager)

This part of the specification deals only with the means by which a multileaving terminal maintains control of the half-duplex data exchange mechanism on its line. It is at this level that decisions on exchange of control between the two ends of the line are decided. [The criteria on which these decisions are made have been modelled on the HASP scheme but are not necessarily identical to them .]

The data exchange mechanism is defined in terms of paired write-read transactions on the line. This allows definition of the required actions and state transitions from the point of view of a single terminal (with the implied assumption that a similar mechanism must exist at the distant end of the communications line). [In ERCC implementations the line manager is a software activity while the handler is a combination of hardware and interrupt-driven software, and any defects of over-definition in this specification may be attributable to this].

The line manager activity handles all terminations of write-read cycles and certain state changes initiated by the line user activity. When the line manager is activated it possesses the following items of information:-

(i) A line stateword of four bits

(ii) Type of activation - line handler termination or user

(iii) An indication of whether there are any changes in the local inbound stream control bits (FCS)

(iv) In case of line handler termination, type of the termination.

When the activation terminates the manager will have performed any or all of the following actions:-

(i) modified the line stateword

(ii) initiated a write-read or WAIT action in the line handler

(iii) terminated a WAIT action in the line handler

On entry to the line manager certain initial decisions are taken (see Fig. 2) and certain initial states will result in exit before referring to the decision matrix (Fig. 3).

On entry to the decision matrix the following actions are performed:

(i) index to the appropriate entry using the line stateword and the entry type

(ii) initiate the indicated action in the line handler

(iii) go to the new indicated state (ie modify the line stateword).

State Number		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Local WABT		CLEAR	CLEAR	CLEAR	CLEAR	CLEAR	CLEAR	CLEAR	CLEAR	SET	SET	SET	SET	SET	SET	SET	SET
Remote WABT		CLEAR	CLEAR	CLEAR	CLEAR	SET	SET	SET	SET	CLEAR	CLEAR	CLEAR	CLEAR	SET	SET	SET	SET
New text buffer ready to go		NO	NO	YES	YES	NO	NO	YES	YES	NO	NO	YES	YES	NO	NO	YES	YES
Any open, non suspended, inbound streams		NO	YES	NO	YES	NO	YES	NO	YES	NO	YES	NO	YES	NO	YES	NO	YES
Line handler termination entry	Unchanged local FCS since last entry	(2)	(4)	(1) → 0	(1) → 1	(2)	(4)	(4)	(4)	(2)	(2)	(1) → 8	(1) → 9	(2)	(2)	(2)	(2)
	Changed local FCS since last entry	(3)	(3)	(1) → 0	(1) → 1	(3)	(3)	(3)	(3)	(3)	(3)	(1) → 8	(1) → 9	(3)	(3)	(3)	(3)
User stateword modification during WAIT period	Unchanged local FCS last entry	(0)	(4)	(1) → 0	(1) → 1	(0)	(4)	(4)	(4)	(0)	(0)	(1) → 8	(1) → 9	(0)	(0)	(0)	(0)
	Changed local FCS since last entry	(3)	(3)	(1) → 0	(1) → 1	(3)	(3)	(3)	(3)	(3)	(3)	(1) → 8	(1) → 9	(3)	(3)	(3)	(3)
State number		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Fig. 3 Decision Matrix

Key to decision matrix

(n) means perform action number n, (0) means perform no action

→ m means go to state number m, absence of → indicates no state change

actions

(1) Write text block, Read

(2) WAIT

(3) Write null block*, Read

(4) Write ACK0, Read

* The null block informs the remote end of the local FCS (including WABT) status

The line stateword bits are:-

(i) Local WABT ((wait-a-bit)see ref. iii) - when set indicates that the line user is unable to accept text (ie data) blocks from the line (and indicates that no text buffer is available).

(ii) Remote WABT - when set indicates that the remote end is unable to accept text (ie data) blocks from the line (this does not forbid the sending of null text, ie control, blocks).

(iii) New buffer ready to go - when set indicates that the line user has provided a text buffer for transmission and that this buffer has not yet been sent. [ie that the queue of buffers for sending is non-empty .]

(iv) Any, open, non-suspended, inbound streams - when set indicates that there exist one or more text data streams inbound to this terminal whose individual control flags (FCS bits)* are in the set (non-suspended) state.

Bit (i) is set and cleared by the line user process, the only absolute constraint being that if it is clear then the user process must provide a text input buffer.

Bit (ii) is set by the receipt of a block (either null or text) with the WABT flag set in the FCS field and
is cleared by the receipt of a block with WABT clear or by receipt of an ACKO control sequence.

Bit (iii) is set by the line user process (when it provides a block for sending) and cleared by the line manager when it takes the buffer [ie finds the queue of outbound buffers empty .]

Bit (iv) is set and cleared by the line user (upon checking the individual inbound stream control flags).

Note that this physical protocol specification makes no attempt to describe the means of opening and closing streams†.

† see ref. (iii).

* [In present practice, stream control bits will be off for a closed stream]. Note also that the logical polarity of the WABT and stream control bits is opposite.

Logical Data Exchange Mechanism
(line user)

This activity controls the multileaving line as a multistream logical i/o device. It provides buffer space to the line manager, supplies it with filled outbound buffers and takes inbound buffers to be passed to their receivers.

It is the responsibility of the line user to switch total data input on and off using the local general WABT, switch off data output based on the remote FCS bits and to control the rate of input on individual inbound streams by means of their FCS bits. [In ERCC implementations this activity consists of a queue scanning mechanism to control rates of data transmission to receiver processes, a means of obtaining free buffers for input and hence control total data input rate, and a mechanism for halting outbound streams when the remote FCS stream control bits are turned off.]

The data buffer format which is dealt with at this level is described in reference (iii) which also includes a detailed description of the stream control functions.

It is at this level and above that the system becomes fundamentally unsymmetrical in that streams are recognised as readers, printers, etc. The data buffer format of ref. (iii) does not apply to the initial terminal SIGNON record in respect of the string control functions. The text of the signon appears immediately after the SRCB.

Neither the stream open mechanism nor the FCS control bit (described in (iii)) are used by either the inbound or the outbound console streams. Console messages consist of the message text in standard form terminated by stream end-of-file.

Conclusion

The HASP multileaving communications technique has been described directly and by means of references. The description has been at three levels. It is clear that there exists some possibility of a clear logical division between levels (ii) and (iii) allowing the use of the levels (i) and (ii) as a pure transparent data freight mechanism with the individual stream control functions of level (iii) kept distinct. Level (iii) then appears as a logical i/o handler for a multiple-stream i/o device. In this manner it may be possible to replace one or other of these two entities at each end of a link allowing either a totally different multi-streaming system or a new data freight mechanism to be introduced independently if and when this is required.

RJE: Bad Protocol by HASP

Hasp II version 3.1 has the following unfortunate departures from accepted BSC protocol which can cause problems when running RJE. This list is based on discoveries during development of RJE2 and is presumably incomplete:-

(a) If there is already a terminal signed on with the remote number (RMnn) which RJE is trying to use, then Hasp drops communication without an error message.

(b) If a block sent to Hasp contains a card-image which deblocks to more or less than 80 characters, Hasp also drops communication.

(c) If a block sent to Hasp is for a stream which has not been correctly opened (by request and accept), Hasp sends "wait-a-bit" forever. The wait-a-bit block is X'90CFCF'.

(d) Hasp does not use the FCS stream-bits to stop input of data when the System is full, nor does it delay the granting of permission when a new input is requested; instead it sets "wait-a-bit" until there is room in the System, thus breaking all communication. (The wait-a-bit block is as in (c) above, but the situation will eventually clear.)

(e) Hasp does not honour the FCS bit for terminal console output; it continues to send output unless the terminal sets "wait-a-bit".

(f) If Hasp cannot deal with a console input message immediately (e.g. $DA), it sets wait-a-bit until it can do so; see (c) above for the block format.

(g) If Hasp times out, it does not send NAK but retransmits the last block.

(h) If Hasp receives a block with a bad BCB-count, it does not transmit a BCB-error-reset block; it acknowledges the block as correct, but ignores it, and then flushes all jobs being read in as though a "$ERMnn.CRn" console

command had been entered. Note that if the block was a console input, this is lost. RJE is unable to detect any error.

(i) During the period between Hasp's requesting permission to start a printer or punch stream and the terminal accepting it, $DJnn will show the job as "on printer/punch" but $DRMnn will show the printer or punch as "inactive".

(j) During a long console output, if a timeout or other error occurs, Hasp diverts the output to the printer. (This may be part of a more general characteristic.)

It is evident that programmers developing modified versions of RJE need to be aware of these sorts of problems. In addition the operators need to have some instructions for dealing with permanent (or semi-permanent) "wait-a-bit". RJE issues the console message "15 seconds wait-a-bit" repeatedly (every 15 seconds) while the situation persists; the operator has to distinguish between cases (c), (d) and (f). All cases except (c) will clear eventually, but this may take several minutes. In a working version of RJE, case (c) can only occur as a result of overwritten program in the Interdata or of an Interdata machine-error.

(R) 20/7/76

A Primer of HASP multi-leaving

P. T. Barry
J. I. Davies

12.7.73

CONTENTS

(i) Introduction to the HASP Multi-leaving Primer

(ii) BSC Manual

IBM document, form GA27-3004
'General Information, Binary Synchronous Communications'

(iii) HASP Manual

Extract from 'Houston Automatic Spooling Priority System with Remote Job Entry'
Program Description

(iv) ERCC specification

Specification of the HASP transparent multi-leaving protocol.

(v) Appendix to the ERCC specification

Restrictions and extensions

INTRODUCTION to the HASP Multi-leaving Primer

1. Introduction

HASP multi-leaving is an IBM system allowing serial data communication between computers. The basic multi-leaving control procedures are a development of the BSC limited conversational mode (Section iii, page 23). In this development, a station is permitted to reply to a text block with a text block. This provides for pseudo-simultaneous bi-directional transmission of multiple byte oriented data streams over synchronous half duplex communication lines.

It has become necessary as a matter of information and definition to have a description of this system for use within the evolving regional communication network, based upon the Universities of Edinburgh, Glasgow, and Strathclyde.

This set of documents is intended to give a comprehensive picture of the structure and facilities of HASP multi-leaving, assuming only a basic knowledge of serial data communications. It also contains data specific to the ERCC implementation of standard multi-leaving and certain local extensions. While particular portions of this primer may have more general applicability, its intention is to describe point-to-point, transparent mode, communications only.

This primer is made up of a number of documents written by different people, and because of this, a certain amount of duplication is inevitable, and for this the editors apologise in advance.

Nevertheless, it seemed the only approach capable of covering the subject at every level including:

(a) The physical BSC transmission frame protocol and control characters for block transfer (Section ii)

(b) The HASP modifications to this protocol for simple bi-directional block transfer (Section iii, part I)

(c) The full HASP record structure within the blocks (Section iii)

(d) The ERCC implementation of this system (Sections iv and v).

2. Use of this Primer

The material contained within this set of documents may be used in a number of ways. In each case, part 3 of this introduction will provide a starting point.

(i) An introductory explanation of the functions and capabilities of the system:

BSC manual - Introduction and BSC concepts with particular reference to point-to-point operation and the control characters: SYN, STX, ETB, ENQ, ACK, NAK, and DLE - additional data link capabilities, specifically transparent text mode, limited conversational mode.

HASP manual - First part up to the detail description of the individual bytes and fields.

(ii) An aid to the analysis of the possible use of the system for a particular application.
The system may be applicable in two ways, either as a total system with or without the extensions described in the appendix to the ERCC multi-leaving specification, or as a simple bi-directional block data transfer mechanism as described in the conclusion to the ERCC specification. In either case the documents are relevant as follows:

All items mentioned under (i), plus
ERCC Multi-leaving specification
HASP manual - detail portions no included in (i),
(specifically the descriptions of BCB and FCS for block data transfer mechanism alone and the whole of the document in the application of the total system).
ERCC multi-leaving specification, appendix - information about the standard and extended multi-leaving as currently implemented, applicable only to a total system implementation.

(iii) A guide to the implementation of the system in a particular environment:
All the material of (ii) plus a more detailed look at the ERCC specification, in particular the details of line control in the line manager. The appendix to the ERCC specification may also be relevant in particular cases where a standard or extended implementation is required. At this stage is may become necessary or advisable to consult listings of existing implementations (in particular where this set of documents proves inadequate or inexplicit - cases where this occurs would be of particular interest to the editors of this primer).

Additionally, details of the specific communications hardware to be used will be required (see for example, references (i) and (iv) within the ERCC specification).

3. Outline Primer

The following chapter is a short overview of HASP multi-leaving which provides a foundation for readers not intimately familiar with the system. (Much of this section is drawn (with thanks) from a document by E. Gaitskell of ICL, Dalkeith).

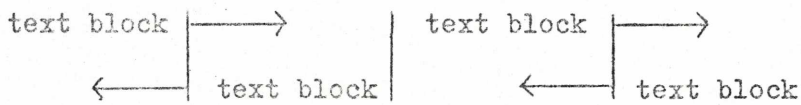
Mode of Operation

Multi-leaving dialogue is an extended version of the BSC limited conversational mode, with the restrictions on ETB and indefinite exchange of blocks removed, and the block format restricted to the form

DLE.STX - text - DLE.ETB

It differs from BSC standards in that it can transmit text block responses to any received block i.e. the primary method of working is alternate, bi-directional, half duplex transmission of text blocks.

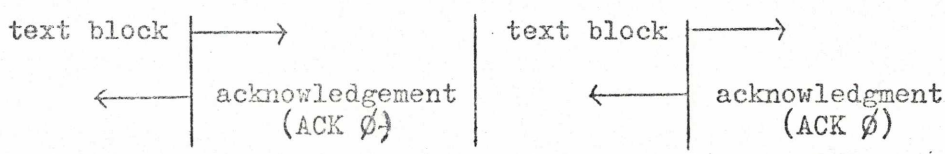
e.g.



The receipt of a correctly sequenced text block in response to a transmitted text block is considered an indication of the correct delivery of the transmitted block. The data streams in either direction are normally independent of each other.

If a station is not in a position to transmit data then it may transmit an acknowledgment (ACK Ø) instead.

e.g.



The BSC alternating ACK Ø and ACK 1 acknowledgements are not used. This level of block sequencing is achieved by utilising a sequence number held within the text block.

Multi-leaving Control Characters

The following BSC control characters are used during Multi-leaving transmissions to remote terminals. (The characters are named together with typical use and any differences in the use from BSC standards, see Section ii, page 11).

STX	start of text	- as BSC standard
DLE	data link escape	- used in BSC transparency conventions and in even positive acknowledgement sequences otherwise not used.
SOH	start of header	- header sequences are not used in Multi-leaving. SOH is used in the sequence to establish initial connection.
ACK Ø	even positive acknowledgement	- acknowledgement parity significance is not used in Multi-leaving. ACK Ø is used to indicate correct reception of a data block in the case when data traffic is logically uni-directional.
NAK	negative acknowledgement	- used by a sink to indicate that the preceding input block or control sequence was corrupt on receipt.
ENQ	enquiry	- used in the sequence to establish initial connection. Not used in error recovery conditions as in BSC standards.
SYN	synchronising character	- used in standard BSC context.
ETB	end of transmitted block	- terminator for all text blocks causes bcc checking and line turn-around. BSC control characters are not used to delimit messages.

NB. No other BSC control characters are used by a Multi-leaving terminal.

Note how a different logical record structure could be built on the basic physical envelope, and that the system is transparent to the data by use of the DLE sequences described in Section ii, page 23. The HASP system permits the records for several streams to be sent in one block. The Control Bytes on the previous diagram are fully described in Section iii, but it should be noted here that it is the FCS that controls the actual Multi-leaving of streams within the system. These control bytes permit a station to switch on and off the incoming streams.

Initialisation and Signon

The system uses the following signon sequence:

secondary

S E P O N A H Q D	S E P O N A H Q D	D E B F L T C C SIGNON TEXT E X B S	R D E C L T B E B

repeat at 3 second intervals

connection established

primary

DAP
LCA
EKD

Signon text in the ERCC implementation is truncated card image of the form:

/*SIGNON ␣␣␣␣␣␣␣ REMOTE5␣

preceded by: general control RCB, signon SRCB - and without the usual terminating RCB. (Note that string control bytes are absent from this record).

Line Maintenance during Idle Periods

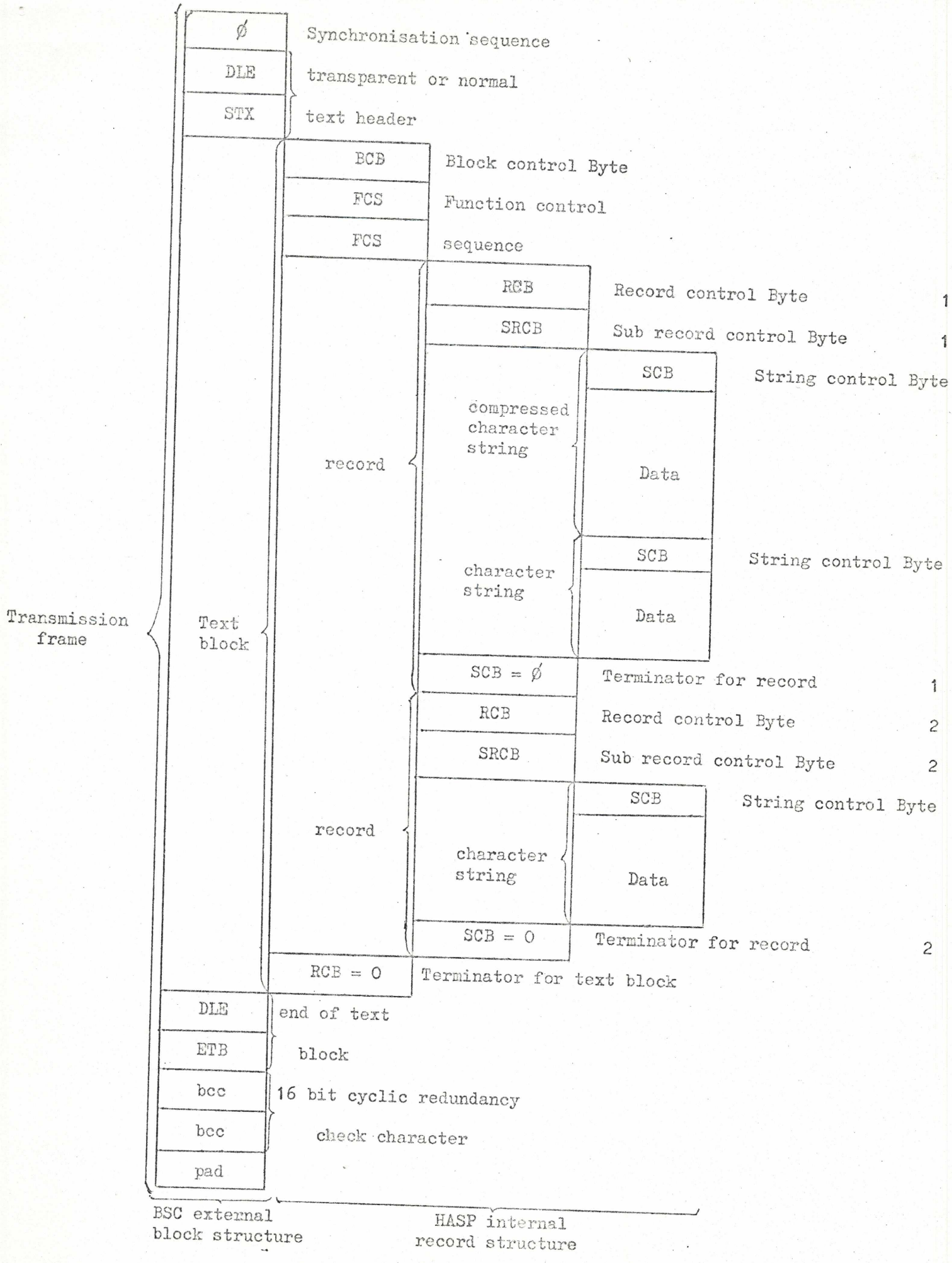
Once connection has been established, the Remote Terminal station should ensure that the connection is maintained by transmitting a dummy data block or idle control sequence every two seconds. For further details see Section iv.

Transmission Termination

Multi-leaving stations do not use the BSC 'end of trans-

Multi-leaving Block format

FIG. 1



mission' (EOT) character to close down transmission. Instead a flag character 'B' within the SRCB control is used.

BCB sequence failures should normally cause an abort of transmission in both directions. No clear termination need necessarily be provided since operator intervention will be required in this case.

Transmission termination can also occur after the requisite number of timeout errors or NAKs.

NOTES:-

A) Commands 1,2 and 3 should be done in that order remembering to use command (3) once special print job is finished.

B) Do not release a job unless hold by yourself for print, never release any job requiring a magnetic Disk, Tape etc.

C) Do not delete any hold job without informing the 370/155 operator.

D) Inform the 370/155 operator of the HASP No. of any jobs which you reroute to the local printer and the reason for doing so.

E) Do not hold any job without informing 370/155 operator.

Additional Commands for an '1130' type terminal

15 $ERMN.PR1	To restart printer of Remote N
$BRMN.PR1	To Backspace printer of Remote N
$FRMN.PR1	To forward space printer of Remote N
$NRMN.PR1	To repeat printer of Remote N
$PRMN.SSS	To stop (drain) device SSS of Rem N after current job
$SRMN.SSS	To start(used after drain) device SSS of Remote N
$IRMN.SSS $IRM1.RRT	To interupt device SSS of Remote N
$CRMN.SSS	To cancel current function on device SSS of Remote N
$ZRMN.SSS	To Halt device SSS of Remote N
$BRMN.PR1,X	To backspace printer of remote N, X pages
$PRM15.PR1	TO DRAIN PRINTER

N = Remote No.

DRAIN
HOLD
INTERUPT

HASP VERSION 3

For a status report on the 4/75, 370/155 and concentrator phone 667 7491

Remote Commands for A '2780' Type Remote Job Entry Terminal

All HASP commands punched on cards should start with
/*£ e.g. /*£DJNN

All HASP commands typed on a teletype '2780' type system should start with
/*$ e.g. /*$DJNN

All HASP commands on a teletype on an '1130' type system should start with
$ e.g. $DJNN

$DJNN	To display the status of a job where NN is the HASP job number
$D'AAAA'	To display the status of all jobs whose identity starts with AAAA
$LJNN	To locate a job where NN is the HASP job number.
$L'AAAA11AA'	To locate a job whose identity is AAAA11AA.
$HJNN A J603	To hold a job whose HASP number is NN
$AJNN	To release a job whose HASP number is NN
$CJNN	To cancel a job whose HASP number is NN before completion of current activity
$PJNN	To delete a job whose HASP number is NN after completion of current activity
$TJNN,P=X	To alter priority of print job to X where X=1-15
$DN,NN	To display the status of all jobs from Remote NN
$DQ,NN	To display the number of all jobs from Remote NN
$DB	To display the backlog on the 370/155
$DF	To display all requests for N Part paper etc. (1)
$TRMN.PR1,F=000N	To set printer of Remote N to accept a job requiring N Part paper. (2)
$TRMN.PR1,F=R	To reset printer to accept 'normal' forms after above. (3)
$RPRT,RMTN,LOCAL	To Route print all jobs for Remote N to local printer
$RPRT,JOBNN,LOCAL	To route print job N N to local printer
$RPRT,RMTA,RMTB	To Route print all jobs for Remote A to Remote B
$RPRT,JOBNN,RMTB	To Route print job N to Remote B
$TRMN.PR1,C=1	To single step paper until next user job
$DM~~0~~, 'MESSAGE'	To send a message to the 370/155 operator
$D( )	

~~649~~ 812
~~364~~

$RPRT, JOB ~~761~~, RMT14