

Proposed Specification for WidebandData Channels for a Modular One ComputerIntroduction

It is apparent that both data transfer demands and available channel capacities are increasing. The impact of this generalisation on the growing Edinburgh medium speed network forces a reappraisal of communications hardware on our main communications processor, the C.T.L. Modular One. The deficiencies of the present multiplexer at high speeds (say greater than 9.6 kilobauds) are apparent and it is clear that the basic character handling overhead on the processor needs to be reduced for high speed operation. This, fairly naturally, leads to the idea of a block transfer device operating on the standard buffer size of 300-400 8 bit characters currently used in the system.

Local Considerations

All current Edinburgh operations use one of two IBM - derived communications protocols, called B.S.C. (Binary Synchronous Communications) or Multileaving. All high speed communication (above 4.8 kilobauds) uses the more powerful Multileaving technique and this is also the easier of the two to implement in hardware.

The first (and already required) application for a Wideband port is the multi-leaving line connecting the Modular One Concentrator at King's Buildings to the mainframe (370/155) at Buccleuch Place. Both the line and the mainframe I/O port (IBM 2701) are already capable of running at higher speeds (up to 48 kilobauds) and the Modular One is the current limiting factor (we use a 4 buffer synchronous line unit).

Developments in the near future must obviously involve full duplex working and any new communications hardware must provide for this.

The Alternatives

Solutions to the stated problem might fall into three categories:-

- (i) Hardware
- (ii) Microprogrammed
- (iii) Software

Obviously, most practicable schemes will involve mixtures of these three techniques. The solutions studied to date are as follows:-

- (i) C.T.L. proposal

This involves use of a dedicated Minimod processor with 4K of store and a modified communications multiplexer. This is a largely software based solution with the attendant flexibility advantages but with a restriction to 3 operational lines. This is recent verbal information and is currently incomplete, for example, it is not known if the lines are to be half or full duplex. Initial calculations made here indicate 3-4 channels (i.e. 3-4 half duplex or 1-2 full duplex) is the maximum capability of a Minimod processor without very extensive modifications to the multiplexer.

The cost is £8,500 plus £1,000 for software, and delivery is 12 months from receipt of order.

(ii) I.C.L. scanner

This is a microprogram based solution which appears to include facilities for Read-Write chaining. The system is said to be able to support up to 16 channels. The cost is £15,000 and delivery time is about 12 months. The ease with which different protocols might be incorporated into the microprograms is not clear.

(iii) Interdata Model 50 communications processor

This system was examined since it appeared to be the only existing, commercially available, system which fell fairly close to our price/performance requirements. The processor has several communications oriented instructions and would appear to be capable of supporting perhaps 2-3 channels (i.e. 1 channel at full duplex). Unlike solutions (i) and (ii) this system does not include either Modular One interfacing or, less importantly, V35 modem interfacing. The problems of providing both store and processor access between the Model 50 and the Modular One are considerable.

(iv) Hardware data port

A hardware solution obviously offers the maximum of efficiency with the minimum of flexibility. The latter disadvantage might be strongly compensated by the system being locally designed and built. This proposal will not be discussed here since it constitutes the main body of this specification.

(v) Locally constructed microprogram processor

If and when there appears a cheap 'processor on a chip' with easy microprogram capability then this will be the optimum system for a cheap, flexible protocol, high-speed data port. However, currently available components do not yet possess the required power and flexibility. One is still left with the main processor interfacing problem.

None of the possible hardware or hardware/software solutions has been fully specified to date.

Both the purely bought-in options involve high initial costs and long delivery times.

Both the partially re-programmable solutions of (iii) and (v) involve extensive in-house design effort, particularly in producing a Modular One interface.

In addition, it is apparent that the proposed ISO standard protocol from the MDS consortium, which includes ICL, is hardware oriented in at least one area (the technique for providing transparent communication).

Outline Proposal

A hardware I O port system, built in-house, has the following features when placed in comparison with the alternative solutions.

- (i) It will be available (with two channels for full duplex operation) by mid-August 1973, subject to availability of engineering manpower (estimated 15 man months initially).
- (ii) It costs (materials only) no more than £1500 including the above two channels and each additional pair of channels costs £400 at present material costs.
- (iii) The construction uses the existing 1.09 termination module, power supplies, and 1.01 interface.
- (iv) The IBM multileaving transparent protocol is implemented initially but additional channels can be designed to implement other protocols (e.g. the GPO EPSS protocol, the MDS proposed HDLC protocol). The additional design effort for a new protocol is relatively small.
- (v) The system could be expanded to handle up to about four full duplex 48K baud lines or a lesser number at this speed with a mixture of other, slower speed, lines. Expansion beyond this would imply the possibility of visible degradation of performance of the host processor due to excessive hesitating. The maximum number of lines (transmit plus receive) of any speed would be fifteen (due to peripheral address restrictions).

The proposed system consists of a series of distinct receive and transmit channels sharing a bi-directional 16 bit wide data bus. The channels are paired and attached to common modem interface units. Allocation of the bus is controlled by an arbitration unit which also controls timing of the data transfer cycles but which is otherwise as transparent as possible to the standard C.T.L. interface cycles. Character transfers to and from store occur by processor hesitation. This involves some loss of processor time but calculations indicate that this will not be significant for any envisaged level of character traffic. The hesitation technique allows access to the whole store address space of the host processor.

The proposed initial configuration is shown in the attached diagram. Page 9.

The Bus System

The bus system is an attempt to provide most of the flexibility of the Modular One peripheral access technique whilst sharing the logic associated with bus control.

The channels are addressable in the standard Modular One sub-channel manner and recognise their addresses individually. The bus controller responds correctly in each case when no channel responds to a broad cast address, allowing insertion and removal of channels without hardware difficulties.

Functions of the transmit channel

In order to act as the transmit channel of a multi-leaving facility the channel must be capable of transmitting two types of message

- (i) A pair of control characters preceded by a fixed number of sync characters.
- (ii) A transparent text message preceded by a fixed number of sync characters and a pair of leading control characters and followed by a pair of terminating control characters and a pair of cyclic redundancy check characters. All occurrences of the special DLE character within the text to be duplicated. The leading pair of characters to be always the same but the trailing pair to be one of two, command selected.

Functions of the receive channel

In order to act as the receive of a multi-leaving facility the channel must be capable of receiving a message preceded by:-

- (i) The requisite number of sync characters and a pair of text start characters (always the same pair) preceding a block of text.
- (ii) The requisite number of sync characters and any other pair of characters than those above, to be placed in store to be examined by the software.

In a block of text the following special character pairs need to be recognised and appropriate action taken:-

DLE-SYNC - ignored

DLE-DLE - transformed into DLE

DLE-anything else - interrupt, illegal character sequence

A text block should be recognised as terminated by one of two possible pairs of control characters.

DLE-ETB - end of block

DLE-ETX - end of block, end of file

A receive command should also terminate after a fixed time if no message start is recognised or after a fixed time if a message start, but no message end, is recognised.

Error Check

A cyclic redundancy check is the standard means of error detection in Multi-leaving. The check digits (16 consecutive bits at the end of a text message) are accumulated according to a set of rules as follows:-

All text characters after a valid text start pair are included in the CRC accumulation, with the exception of the pair DLE-DLE which counts as DLE and the pair DLE-SYN which is ignored. At end of text the pair DLE-ETB or DLE-ETX is accumulated as ETB or ETX respectively.

Commands from Processor to Wideband Ports

All commands will be issued by the processor in the standard Modular One peripheral addressing manner (i.e. using the top 8K of address space).
Commands common to receive and transmit channels:-

- (i) Read status general
Passing status bits in the slave word
- (ii) Abort
Returning the channel to its idle state
- (iii) Initialise
Loading non-transient parameters from a set of consecutive store addresses (passing base address in master word)
- (iv) Standard CTL commands (Test, Set, and Reset channel lockout)

Commands to the transmit channel only:-

- (i) Write control character pair
Which sends the indicated pair of control characters preceded by sync characters.
- (ii) Write data without end of file
Which sends sync characters, a pair of standard text start characters, text, a non-end-of-file terminating pair and the two CRC characters.
- (iii) Write data with end of file
Which does the same as (ii) but uses the end-of-file terminating character pair.

Commands to the receive channel only:-

- (i) Read
- (ii) Read and start timer
- (iii) Start timer

Commands to Wideband Communication Channels

Command Name	Obedied by :-	Slave Word	Master Word	Rejected if:-	Remarks
Test channel lockout	Channel or Arbitration unit?	-	-	As defined for 1.01	These commands are to be obeyed appropriately at all times
Set channel lockout	"	-	-	"	
Reset channel lockout	"	-	-	"	
Read Status General	Channel (either Tx or Rx)	Status Bits	Null	"	
Abort	"	"	Null	"	This command is to be used only at system start-up
Initialise	"	Null	Address of start of parameter area	As above + Busy	
Write control	Transmit Channel	Null	Pair of control characters	"	
Write with- or EOF	Transmit Channel	"	Address of start of parameter area	"	
Write, with EOF	Transmit Channel	"	"	"	These four commands require extra parameters from store
Read	Receive Channel	"	"	"	
Read and Start Timer	Receive Channel	"	"	"	
Start Timer	Receive Channel	"	Number of clockticks required	"	This facility may be omitted upon more detailed consideration

Certain commands require additional parameters to those provided in a 16 bit master word. In these cases the master word of the command points to a fixed format parameter area in store from which the channels registers are loaded after the command but before any communication activity takes place. The channel will reject commands ('busy') while this takes place. These commands fall into three types for the purpose of parameters.

- (i) Init - a number of parameters (as yet unspecified) are loaded. They include:-
 - a. Interrupt dedloc address for channel
 - b. Number of clock ticks for receive timeout
 - etc
- (ii) Read (with or without timeout)
 - a. Receive buffer address
 - b. Maximum number of received characters
- (iii) Write (with or without End-of-file)
 - a. Transmit buffer address
 - b. Number of characters to be transmitted

Interrupts from the Wideband ports

Each commanded operation terminates itself in an interrupt to the processor. Each channel has one dedloc address to which it interrupts. Additional information about the interrupt is passed in the masterword. In most cases this is a simple numeric identifier indicating the nature of the interrupt. The only exception to this is a receive channel termination after successful reception of a block, in which case a count of received characters is placed in the lower bits, the top bit is set as an identifier, and the next as an end of file indicator.

The following distinct interrupts are generated by both transmit and receive channels

- (i) Initialisation completed
- (ii) Processor rejects hesitations - store not operational

The following distinct interrupts are generated by transmit channels

- (i) Write done
- (ii) Transmit clock lost (modem fault)

The following distinct interrupts are generated by receive channels

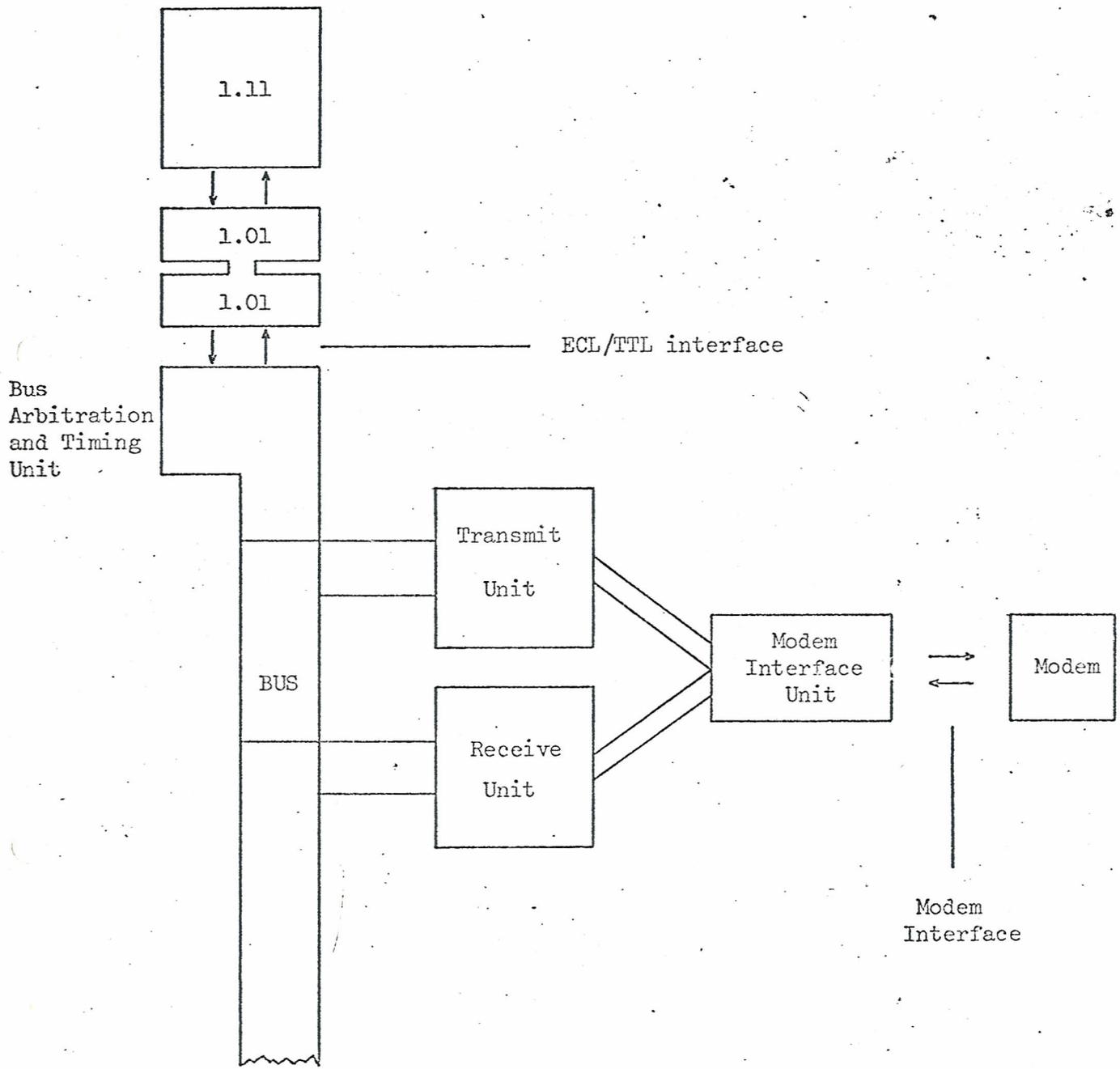
- (i) Block successfully read with EOF
- (ii) Block successfully read without EOF
- (iii) Leading characters not valid text start pair
- (iv) Read overflow (too many characters for buffer)
- (v) CRC fail on incoming block
- (vi) Timeout
- (vii) Illegal character sequence in text

Conclusion

A system which provides block transfer facilities for communication lines has been defined. Detail design (additional to the information in this document) has proceeded to the point where a large degree of confidence in the attainment of the specified aims has been acquired.

The system has been deliberately designed to reduce the extent of software rewriting required.

Examination of two proposed new protocols indicates that they are implementable within the structure of the proposed system.



Initial configuration of Wideband data port system with one full duplex line.