A SIMPLE ASYNCHRONOUS INTERFACE FOR LINKING SMALL COMPUTERS

Peter J. Lindsay
Department of Computer Science
University of Edinburgh
Edinburgh
Scotland

ABSTRACT

An interfacing technique is described which enables the linking of small computers using serial asynchronous character transmission along standard 75 ohm coaxial cable. This technique can also be applied to the interfacing of any device which requires data transfer rates up to 10,000 bytes per second.

INTRODUCTION

The Department of Computer Science at the University of Edinburgh is responsible for teaching and research in the field of computers and makes use of 4 DEC computers for teaching : a PDP 8, a PDP 9, a PDP 11, and a PDP 15.

On a number of occasions over the past few years the possibility of providing some form of link between a number of the Department's computers has been mooted. Examples of the kind of use envisaged are to enable one machine to obtain access to a peripheral device on another, to link one or more machines to a central input/output controller or file-processor, to provide the basis for experiments in inter-machine communication as such, and so forth. The decision to proceed with the first link was taken when it proved difficult to obtain an interfaced card-reader for the PDP15 at a reasonable cost and it seemed probable that a card input facility for it could be most readily provided via the PDP8 to which a card-reader was already being interfaced. The PDP15/PDP8 link was designed and constructed but the project failed to meet the immediate objective, because of difficulties with the PDP8 card-reader. However, the opportunity that was provided for testing and experimentation proved most valuable and the investigation of different transmission protocols by software made it possible to identify the kind of scheme that would be most convenient and the main problem areas. This investigation led to the realization of a modified hardware design which had many advantages over the original version.

DESIGN CONSIDERATIONS

The basic requirement was stated as the provision of a means of file-transfer from one computer to another at a reasonable cost in terms of hardware and software. At the lowest level this would give a preferable alternative to using paper-tape, for example, as a transfer medium. Even for the more elaborate applications envisaged, a data rate of a few thousand characters per second would be adequate. There was no question, therefore, of requiring to aim for the very high-speed transfer rates which are associated with full-scale interprocessor links using direct access to the fast storage of the machines concerned. In

any case the varying word lengths of the machines concerned (12-bits, 16-bits and 18-bits) would have posed major problems for this approach and the cost would have been unacceptably high. For these reasons, character, rather than word, transmission was indicated. Considerations of hardware simplicity, availability of modules, line costs and flexibility of application resulted in the choice of serial, rather than parallel, operation, while the aim of minimizing the software overhead as well as hardware convenience, indicated asynchronous, rather than synchronous, operation, provided that adequate performance could be achieved and a suitable protocol devised to control transmission. In this respect it should be stressed that, because the basic aim was to provide for the local inter-connection on private lines of machines situated fairly close to each other, the design was not constrained either by performance limitations likely to affect long transmission lines or by the need to operate within recognised standards. In particular it was hoped that it would be possible to achieve a baud rate which would be sufficiently high not to be a limiting factor on effective transmission rates at all.

In summary, the balance of the relevant factors pointed clearly to serial asynchronous transmission on a character by character basis using a sufficiently high baud rate to cover the required performance range. For a given data transfer operation the link would be handled as an output device by the sending computer and an input device by the receiving computer, similar to other such devices on the respective machines.

EVOLUTION

The first version of the link, as originally implemented on the PDP15/PDP8, consisted essentially of a speeded-up 'Teletype' interface on each machine with a line connecting the transmitter on the PDP15 to the receiver on the PDP8 and another line connecting the transmitter on the PDP8 to the receiver on the PDP15. Since a Teletype interface (on these machines) comprises a logically independent input and output device, this arrangement provides independent data paths in either direction.

In this form the link was handled by software in terms of the software echo protocol, whereby data transferred in one direction is echoed on a character-by-character basis by the receiving machine on the other line. The purpose of the echo is twofold: firstly it provides a check on the accuracy of transmission and secondly it provides a control on the effective character transmission rate by requiring the transmitting machine to wait for the echo of one character before sending the next. In this way no requirement is placed on the receiver-handling software either to guarantee acceptance within a given time or to buffer blocks of information.

In terms of the input/output instruction organisation of the PDP8 and PDP15, the steps required to send a character are: transmit character; ignore transmitter flag; wait for receiver flag; read character and compare with original. The steps required to receive a character are: wait for receiver flag; read character; re-transmit character; ignore transmitter flag. In essence, for the sending machine it is the receiver flag rather than the transmitter flag which signals that the transaction is done. In practice this protocol suffers from the following drawbacks:

a) Transmission can be in one direction only at a time, although at the hardware level there is a full-duplex capability.

b) The effective transmission rate is limited to half the baud rate because two transmissions are required for each character of data (not in fact relevant at the baud rate used).

c) The error-checking capability of the set-up suffers from the disadvantage that an error (or apparent error) is detected by the sending end rather than the receiving end, which is rather inconvenient. In practice, however, the link proved monotonously accurate, so that it was clear that less rigorous checking, such as character parity, would be adequate.

d) The software to handle the link in the two computers was reasonably simple, but not as simple as for the most straightforward character input or output devices.

This was mainly because both sending and receiving involved operations on two devices (receiver and transmitter) rather than one. An ideal system would permit link transfers to be handled in exactly the same way as, say, a paper-tape reader (for input) or a teleprinter or paper-tape punch (for output).

An analysis of these problems and consideration of various other protocols which might be implemented on the same hardware led to the conclusion that there was no way of eliminating them. The use of a form of the 'constant chit-chat' conventions typically employed by synchronous systems would have permitted two-way transmission, but at the expense of complicating the software (by imposing a requirement for buffering and timing-out) rather than simplifying it.

From this it followed that the problems could only be dealt with by modifying and extending the hardware. If this step was to be taken, it seemed reasonable to adopt the aim of producing the ideal system from the programming aspect mentioned in (d) above. To this end the basic steps involved in transfers to and from the simplest forms of character input/output devices were analysed. For input these are: select device (not required for a spontaneously operating device like an on-line console keyboard); when ready to accept data, wait if none available; otherwise accept character from device. For output all that is involved is: when ready to send data, wait if device is busy; otherwise transfer data to device. This fairly general description covers all the machines involved and ignores details like the way in which the status of an input device is determined and the nature of the interrupt system (which is logically irrelevant). What is evident from this analysis is that, since a 'select device' operation is not obligatory, input must be a passive operation and output the active operation. Accordingly a data-transfer must be initiated by the sending machine; on transfer of a character, the output device must become 'busy' and must remain so until the character has not only reached the other end but has been accepted by the receiving machine. On the receiving end - in the normal way - the 'data-available' condition exists from the time the character is received on the line until the time it is accepted by program. The key point is the way in which the acceptance of the character by the receiving machine is logically associated with the termination of the

'busy' condition on the output interface of the sending machine. The basic innovation in the modified link design was to introduce a mechanism to give effect to this association: this consisted in arranging that when, and only when, a character received on the link was accepted by program (i.e. read from the receiver buffer) an acknowledge pulse would automatically be sent back down the line and that this pulse would clear the 'device busy' status at the transmitting end.

The resulting system, using what will be called the 'Automatic Acknowledge' (AA) protocol, has a number of advantages in addition to achieving the required simplicity of program operation. For data transmission in one direction, only one line is required connecting the transmitter of the sending machine to the receiver of the accepting machine. This line is, strictly, half-duplex, because although data is transferred in one direction only, the control information is sent back on the same line. No interference can arise from this, since until the acknowledge pulse occurs the sending machine is inhibited by protocol from using the line. As well as making it possible to have two-way transmission between a pair of machines using two lines, the fact that one-way transmission does not require both lines leaves open the opportunity to interconnect machines in other ways, for example by daisy-chaining, if this is required.

The fact that the aim of programming simplicity had been successfully achieved was demonstrated by running the standard PAL11 Assembler on the PDP11 with input and output (binary and listing multiplexed: there is sufficient redundancy to unravel them at the other end) over the link to the PDP15. To use the Assembler completely unmodified it was necessary to assign the link to the device addresses reserved for the paper-tape reader and punch; a switch was fitted to the PDP11 to enable this to be selected by push-button.

TECHNICAL DESCRIPTION

The transmitter transmits 11 bit serial asynchronous characters consisting of one start bit, 8 data bits, and two stop bits. The acknowledge pulse from the receiver is defined to be one bit-time in duration and to start at least one bit-time after the middle of the last information bit of the transmitted character.

The transmitter is prevented from transmitting until one half bit following the acknowledge pulse. Thus there is a gap of at least one half bit between the last information bit and the begining of the AA pulse and between the end of the AA pulse and the begining of the start bit of the next character. These half bit gaps are introduced to enable the transmission line to settle before the leading edges of the start bit and AA pulse; these leading edges being the only significant edges in the signal.

A clock running at 8 times the required baud rate is used, and to ensure clock speed independence the synchronization and timing of the acknowledge pulse are accomplished by using a 4 bit binary counter with suitable output gating. The baud rate used for testing was 264,000 baud.

The transmission lines are 75 ohm 'Economy' coaxial cable with BNC connectors. Coax driving and receiving is done by special integrated circuits.

As the purpose of this paper is to describe an interfacing technique rather than a specific interface, no detailed description is given.

The main points of the technique are:-

1) Control of the transmission rate is obtained by the receiver sending an Automatic Acknowledge pulse after each character is read.

2) The same line is used for both data in one direction and control in the other.

3) The timing is as shown in fig 1.

CURRENT STATUS

At present there are three link interfaces in operation, one on each of the PDP8, PDP11 and PDP15. An original version of the link was built for the PDP9 but this has not yet been updated.

These links are currently being used to give the PDP11 and PDP8 access to the peripheral power of the PDP15 (Dectape and high speed printer). For example, the PDP11 absolute loader is loaded from Dectape on the PDP15 over the link to the PDP11 and is then used to load the PAL11 assembler from Dectape. This assembler is then run with input and output over the link using Dectape for source input and binary output and the high speed printer for listing. The binary file can then be loaded over the link. All this is achieved without any alteration to the original PDP11 software.

The PDP8 PAL III assembler is similarly run using the Dectape and line printer of the PDP15 but in this case some half dozen input/output instructions are patched to use different device codes.

FUTURE DEVELOPMENTS

In the immediate future it is planned to have one link interface on each of the Department's computers so that any pair can be connected by patching the necessary cables at a central distribution panel. In addition it is hoped that all slow to medium speed devices will in future be interfaced through link controllers so that, once a device has been 'interfaced', it can be plugged into any of our computers or indeed into another compatible device.

Work is currently in progress to produce link controllers for a line printer, a paper tape reader and a card reader. When these are ready it should be possible to connect the paper tape reader or the card reader directly to the line printer and so do a certain amount of off-line testing.

The reader may have noticed that in defining the timing of the AA pulse the term 'last information bit' was used, rather than 'last data bit'. This distinction is made because it is most likely that instead of using 8 data bits and 2 stop bits, future links will use 8 data bits, 1 parity bit, and 1 stop bit. (The concept of a stop bit is really redundant in this application).

The parity bit, which is an information bit but not a data bit, will be generated and detected by hardware. Parity errors will be signalled to the receiver program by a 'parity error' flag and a double length AA pulse will be returned to the transmitter. The double length AA pulse will be detected by the transmitter and signalled to the transmitter program by a 'transmission line error' flag.

The advantages of using a separate parity bit are obvious. The disadvantages are: added cost and complexity, less compatibility with existing hardware (Teletype transmitter/receiver modules).

CONCLUSION

The Automatic Acknowledge link interfacing technique offers great flexibility, which will allow the reconfiguration of computing systems thought so necessary in a teaching department.

Advantages

1) Very simple and inexpensive
2) Easily connected and disconnected
3) Requires only simple software
4) Reasonably high speed
5) Transmitter and receiver completely independent.
6) Can be used as a general purpose interface

Disadvantages

1) Not fast enough for all applications
2) Performance poor over long distances (due to propagation delays and the need to acknowledge each character).

ACKNOWLEDGEMENT

The author wishes to acknowledge the assistance of his departmental colleagues, especially Mr. Hamish Dewar with whom all aspects of the work were discussed in detail, Dr. David Mills, and Professor Sidney Michaelson who allowed the author the freedom to work on this project.

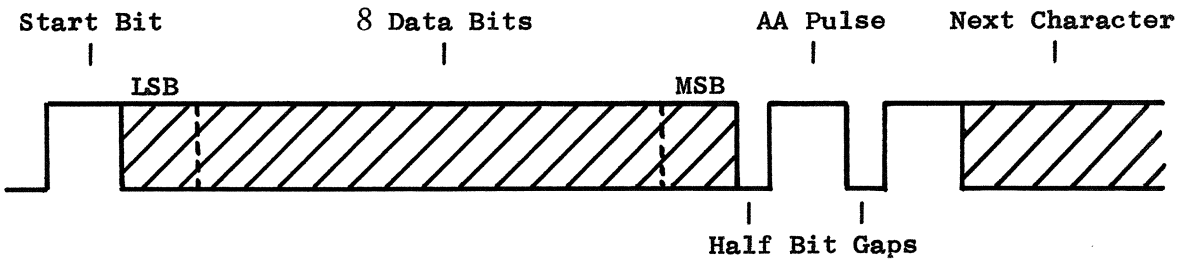

Fig. 1 Timing.