Experience with the use of the British Standard Interface in Computer Peripherals and Communication Systems

D.L.A. BARBER
National Physical Laboratory
Teddington, Middlesex
England

Introduction

The new British Standard Specification[1] 4421 entitled "A Digital Input/Output Interface for Data Collection Systems" is gaining wide recognition in the United Kingdom and has been submitted to the International Standards Organisation for International appraisal. As many of the concepts embodied in the Specification derived from early work by Computer Science Division at the National Physical Laboratory it is now extensively used throughout NPL. There is a range of basic modules which may be interconnected by B.S. Interface to form various systems for collecting and processing data, and for controlling experiments. These modules which are mainly constructed to conform with the NIM (Nuclear Instrumentation Module) standards[2] of the U.S. Atomic Energy Commission are now becoming available commercially, and are used by most Division of NPL. Some Divisions also have small digital computers fitted with the B.S. Interface which allows them to be readily incorporated in monitoring and control systems.

When it was decided to build the comprehensive data communications network described in the companion papers[3][4] there were clear advantages to be gained by using the Interface as extensively as possible for connections to the network. But beyond this, the basic principles of the Interface apply generally to the exchange of information between systems and therefore appear in various guises within the network.

Although the B.S. Interface has been described elsewhere[5][6][7] a brief outline is given here to facilitate the subsequent discussion of its applications to computer peripherals, small computers, and the NPL data network.

The British Standard Interface

The new Standard was prepared by the British Standards Institution in co-operation with users and manufacturers of computers and data processing equipment, taking as a basis a simple interface originally proposed by the National Physical Laboratory. As far as possible the Interface has been made independent of particular devices and is designed to allow the transfer of information in the form of parallel eight-bit characters from any Source to any Acceptor of data. As the transfer of information is in one direction only, two Interfaces are used in opposite directions if bi-directional transfer of data is required. The Specification is in two parts, a Functional Specification and an Electrical Specification. The Functional Specification defines 18 lines and the binary signals which appear on these lines and interact to effect the transfer of information from the Source to the Acceptor. The Electrical Specification defines a particular way of implementing the Functional Specification.

The Interface is illustrated by Figure 1. This shows the Source fitted with a fixed 18-way socket and the Acceptor fitted with a similar 18-way plug. The two devices are joined by a cable fitted with mating connectors. The Interface lines are listed below. The first two lines are passive links between the Source and the Acceptor. The screen or protective earth is intended to join together the chassis of the two devices. The zero voltage reference carries a potential to which the voltage on all other lines is referred.

The remaining lines carry binary signals. The first pair are the Operable signals. These are Logic One when the devices are able to operate and Logic Zero when they are out of action. The next pair of lines are the Control lines. These interact in a hand-shake fashion to transfer characters between the Source and the Acceptor. The hand-shake method is described in more detail later.

The Acceptor Error signal is made Logic One by an Acceptor to indicate when it has detected an error in a previously transferred piece of information. The Source Terminate signal is made Logic One by a Source when it has reached the last character of a block. This enables variable length blocks of information to be

transferred without the need to employ characters in the data stream. It is worth mentioning that Acceptor Error and Source Terminate are conceptually very similar. Acceptor Error enables an Acceptor to indicate one of two responses to previously transferred information, while Source Terminate allows the Source to indicate that one of two types of information is being transferred.

The next pair of lines, Parity Valid and Parity, are closely related. Some Sources provide a parity bit with data to serve as a check of its validity, others do not. Because the Interface is intended to have a wide application the Parity line is included in case a parity digit is used; but it is then necessary to have a Parity Valid line to indicate when the Parity digit is meaningful. The Parity Valid signal may be used to enable or disable Parity checking circuits on an Acceptor, and so avoid the false indication of wrong Parity from a character without a Parity digit. The use of Parity Valid enables a mixed stream of characters, some with parity and some without, to be handled by the Standard Interface.

The remaining 8 lines of the Interface carry the digits of the character in ascending order of significance.

The hand-shake method of controlling data transfer is illustrated by Figure 2. This shows the cycle of signals on the two Control lines required to transfer one character. The Acceptor requests data by setting Acceptor Control to One. Until this occurs the Source must maintain Source Control at Zero, although it may,if it wishes,set a character on the data lines. When Acceptor Control has become One and a data character has been set the Source may set Source Control to One. Once Source Control is set, the Source must hold the data steady and must keep Source Control at One until it detects that Acceptor Control has been set to Zero by the Acceptor. This happens when the data has been accepted, and is an indication to the Source that Source Control may be set to Zero and the next character prepared. When the Acceptor detects Source control at Zero it may set Acceptor Control back to One as soon as it is ready for another character.

It can be seen from Figure 2 that the transition of each Control signal has to be followed by a transition of the other Control Signal. As there are no timing

restrictions on the operation of the Source and Acceptor, a fast and a slow device may be connected together and will operate satisfactorily at a speed governed by the sum of their operating times and the delays in the cable.

The Electrical Specification given in the British Standard was based on three criteria. It should conform if possible with any existing International Standards; it should be tolerant of electrical interference induced in long connecting cables; and it should be reasonably cheap to implement.

The only Internationally accepted standards are the Commité Consultatif International de Télégraphique et Téléphonique (C.C.I.T.T.) [8] recommendations for use with data transmission modem equipment, and a modified form of these Recommendations was adopted with an improved noise performance. This does not give the cheapest possible interface, but rather gives an economical way of satisfying the first two criteria. The Electrical Specification is illustrated by Figure 3. When a transmitter is connected to a detector, Logic Zero must lie between -5v and -11v, and Logic One between +5v and +11v, and when the transmitter is open circuit its output voltage must not exceed +12.5v or -12.5v. The input characteristics of a Control signal detector are shown at the centre of Figure 3. The threshold at which a Control signal is detected as changing from Logic Zero to Logic One must lie between +1½v and +3v, and the threshold for detecting changes from Logic One to Logic Zero must lie between -1½v and -3v. The action of this circuit in reducing the effects of noise can be appreciated by supposing that the input signal is at Logic One. This voltage will be at least +5v and a negative noise-pulse superimposed on this voltage must carry the input to the control signal detector to at least -1.5v before it can cause a change in the output. Noise spikes must therefore be at least 6.5v at the input to the detector. The B.S. Specification calls for the insertion of simple noise rejection circuits in each input line, and if these circuits are smoothing filters it is always possible to increase their time constants to such a value that noise spikes are reduced below this critical amplitude of 6.5v.

The characteristic specified for detectors for all other signals is shown at the bottom of Figure 3. These detectors are more sensitive than the control signal detectors and will operate earlier with signals of a given rise time. So all signals

of the Interface will have been detected correctly before a control signal is detected. This overcomes the effects of skew due to variations in the transmission times of parallel signals along the coupling cable.

Unfortunately, the data transfer rate allowed by the noise rejection filters may be too low for some applications, and if this is so, the Electrical Specification cannot be used and some other signalling method must be employed. However, the published Specification forms a valuable reference standard because the effects of noise in cables can usually be eliminated when determining if two devices interact in accordance with the Functional Specification.

In practice, the Functional Specification has been used by different organisations outside NPL for interfaces with various electrical signalling methods. For example, Transistor-Transistor Logic levels have been used for cheapness, twisted pair cables with balanced transmitters and receivers for high speed transfers while, in situations with excessive common-mode noise, opto-electronic convertors have proved invaluable. Clearly, it is easy to convert from one signalling method to another, so the Electrical Specification is much less important than the Functional Specification. Adoption of the latter by the various organisations has enabled fruitful exchanges of ideas and equipment to occur, and has provided a common language and understanding for the discussion of mutual interface problems.

In this respect the B.S. Interface is like the interface used for connections between data terminal equipment and modems supplied by a common carrier. As the components of teleprocessing systems made by all manufacturers must be compatible with this modem interface it could, in principle, be used for a direct link between systems made by two different manufacturers. For this to be possible it is necessary to agree on common information exchange procedures between the two systems. The large number of control lines (up to 28) needed to control the serial transfer of data is rather uneconomic, so the modem interface is not attractive for general use.

With the B.S. Interface the agreement of common procedures is also necessary - often they can be the same as those used in teleprocessing systems, but the transfer of information in parallel bytes using just a few control signals is much more suitable for the majority of devices which handle information internally in a parallel fas-

Applications to Peripheral Devices

In the experimental environment at NPL the B.S. Interface is particularly important because it enables different systems to be made from a set of basic modules, and allows the reconfiguration of systems to meet changing requirements. The availability of digital computers fitted with the Interface greatly extends the scope of this building block approach to system design, and the use of the Interface for new peripheral devices makes them immediately useable with existing computers and systems.

An important factor encouraging the wider use of the Interface has been its adoption by International Computers Ltd. for the data communications equipment available with the 1900 series computers[9]. The ICL 7020 multiplexing terminal equipment which is fitted with two Source and two Acceptor British Standard Interfaces connects to any standard 1200, 2400 or 4800 baud modem and permits the customer to join peripheral equipment fitted with B.S. Interfaces to a telephone line. ICL supply interfaced card readers, line printers and teletypes, but the 7020 terminal is already being used by some customers to connect their own equipment to remote ICL 1900 processors.

Besides ICL, other manufacturers like Elliott Automation, Honeywell and the Digital Equipment Corporation also offer the Interface on certain of their computers marketed in the UK. Quite often this is an expensive extra item but as the demand grows the price will no doubt decrease, and even the current prices can be justified by a research establishment, where the use of standard connections for all computers can result in considerable economies in the time taken to try out new ideas. The availability of the Interface on different types of computer has encouraged several manufacturers to adopt it for new peripheral devices, and a rapidly increasing amount of equipment incorporating the Interface can be obtained in the U.K.

The adaptation of peripheral devices to operate from the Standard Interface is usually straightforward so no discussion of the subject will be given here. But it is worth stressing that, for the benefit of users, devices performing similar functions should use the Interface and associated procedures in a similar manner, and it is in this area that further standardisation agreements are urgently required.

Applications to Computers

The flexibility of a digital computer is a problem when deciding precisely how it should be adapted to match the standard interface. From the user's viewpoint there are obvious advantages if the Interface is fitted in a similar way to all computers, and some recommendations which may help to achieve this end are given below. They are based on arguments discussed in a report available from the NPL[7].

The method of attachment depends on whether it is connected to the Input-Output busbar, or to the direct store access equipment of the computer. In each case, however, the Operable and Control signals are treated in the same manner.

The outgoing Operable signals should be derived from the 'power-on' signal of the computer or perhaps from a permanent or switchable logic one signal. The use of a program instruction to set and clear the Operable signal is a possible, but not essential, refinement.

The incoming Operable signals should normally cause a program interrupt when they change from logic one to logic zero. But it is essential that a means is provided to mask out, or otherwise ignore, such interrupts as required by the circumstances of any particular application.

An outgoing Control signal will be derived from the 'data ready' signal in a Source and the 'ready for data' and 'data accepted' signals in an Acceptor.

An incoming Control signal may, however, be treated in two ways. It may be arranged to give a program interrupt when a change of logic state occurs, or it may be 'polled' by a program loop to detect if a change has taken place. It should be possible for a user to operate the computer in either of these two ways.

Connection to I-O Bus

When the British Standard Interface is connected to the input-output busbar of a computer it is recommended that the scheme shown in Fig. 4 should be adopted. In this scheme, all the lines of the Interface are allocated to digits in the computer word and software is employed to make the Interface suitable for use with different peripherals.

Unfortunately there are two basically different ways in which a computer may be required to operate, depending on whether or not the Acceptor Error facility is in use:

(1) When Acceptor Error is used, a Source computer must wait until Acceptor Control becomes logic one so that it may interrogate Acceptor Error to decide if information may be changed or must be repeated. Similarly, an Acceptor computer must delay the setting of Acceptor Control until it has checked the data and is able to set Acceptor Error to logic one if this is necessary.

(2) When Acceptor Error is not in use, a Source computer can prepare new information as soon as Acceptor Control changes to logic zero, while an Acceptor computer may set Acceptor Control to logic one as soon as data has been transferred into the computer.

For both Source and Acceptor, the use of Acceptor Error clearly slows down the rate at which information may be transferred. It might be argued that parity checking hardware in the Acceptor with a hardware buffer from which information may be resent in the Source would be a remedy. But apart from the considerable extra expense and the restriction that only characters may be exchanged, some loss in speed is inevitable. This is because the hardware buffer at the Source may not be reloaded by the Source computer until Acceptor Control has changed to logic one, indicating that Acceptor Error is valid. (Otherwise the buffer could have been loaded earlier when Acceptor Control became logic zero.)

This situation has a parallel in the use of the Standard Interface in data collection systems. In any system, the fastest rate of data transfer is obtained if the Source prepares new information when Acceptor Control changes to logic zero. On the other hand, the information may be out of date when Acceptor Control becomes logic one again, (for example, if the Source is an analogue-to-digital convertor sampling an analogue signal). If this is the case, there is no alternative but to delay the preparation of information by the Source until Acceptor Control becomes logic one.

It is desirable, therefore, that two modes of operation of the Interface are made available to the user. It is necessary to provide some means of switching between these two modes, and it is suggested that this should be done by a program

instruction rather than a mechanical switch, so that the instruction may be included in the software relevant to a particular peripheral equipment.

Connection direct to store

When the British Standard Interface is connected to a direct store access system it is recommended that the scheme shown in Fig. 5 should be adopted. Although there may be a need to transfer single characters in and out of the store it is much more likely that the direct store access system will be used for blocks of characters and that transfers will take place at high speed. As users may wish to pack data in the store in a variety of ways, it is suggested that a full computer word be provided by using a large connector capable of accommodating the extra data lines. (Admittedly this does not conform strictly with the B.S. Interface Specification but it is a trivial matter to select eight of the bits when a true B.S. Interface is required.)

With regard to Parity and Parity Valid, it is suggested that where an internal parity signal is available it might as well be brought out and Parity Valid be set to logic one, but that it is not worth while adding extra hardware in order to generate and check parity. In general, error checks will be carried out over blocks of characters which will be transferred and repeated as required using Source Terminate and Acceptor Error. Source Terminate is used in the Acceptor computer where it causes an interrupt to call for program to deal with the new block of information; Acceptor Error is used in the Source computer to cause an interrupt to request the retransmission of the previous block of information. But it is important to ensure that both Source Terminate and Acceptor Error are interlocked with the control signals so that the correct procedure occurs at the end of each block.

When a computer acts as an Acceptor it will normally set Acceptor Control to logic one after each character has been put into the store. However, when Source Terminate is received, the computer must not set Acceptor Control to logic one until the block has been checked for errors. This allows it to set Acceptor Error appropriately before setting Acceptor Control to logic one to request the first character of a new, or repeated, block.

When a computer acts as a Source, and generates Source Terminate from an internal 'end of block' signal, it must interrogate Acceptor Error before starting a new block in case a block has to be repeated.

Use in the NPL Network

The NPL Network[3] comprises a small computer connected by serial data links to terminals distributed about the Laboratory. A novel method of concentrating traffic from clustered terminals using a byte multiplexing technique economises on cables and the need for storage at terminals.

The computer handles all data traffic between terminals in a short message store-and-forward manner and dynamically allocates blocks of store as required by currently active terminals.

The B.S. Interface is used for connections between the network terminals and equipment provided by the 'subscriber'; a modified form of the Interface is used within the network whenever parallel data transfer is required; while a 'character' handshake technique is used where the flow of data occurs serially.

The basic building blocks of the NPL data network are shown in Fig. 6 which illustrates the various interfaces between units. These are:

1. The Message Switching Computer (MSC)
2. The Input/Output Controller (IOC)
3. The Multiplexer (MXR)
4. The communications units Master and Slave Line Terminals (MLT, SLT)
5. The Peripheral Control Unit (PCU)

The first two units - the message switching computer and the input/output controller - really form a single entity because, amongst other things, the input/output controller has to match the direct memory access interface of the computer to a pair of Network Standard Interfaces. The Network Standard Interface is based on the British Standard Interface and is used for all parallel interface connections throughout the network. The Network Standard Interface uses TTL (Transistor Transistor Logic) signal levels and is functionally similar to the true B.S. Interface except that the Acceptor Error signal is not used, and there is a total of 17 data bits - the eight data bits of the true B.S. Interface plus up to nine 'address' bits. These address bits are required by the input/output unit to control the transfer of the eight data bits into or out of areas of the message switching computer store.

Briefly, this is accomplished by using 'pointers' containing a free store address dynamically allocated by the operating system. The pointers - one for each terminal - are, themselves, held in the main store and are autonomously accessed using the nine bit address. The system may therefore accommodate up to 512 terminals.

The Source Terminate signal of the Interface distinguishes data from status or control characters in the normal manner. For example, when Source Terminate is ONE inward transfers cause a program interrupt to interrogate, decode and respond to the accompanying character; but when Source Terminate is ZERO the character is stored without software intervention.

The use of a common interface allows the remaining building blocks of the system to be used in any desired configuration, and with any message switching computer which has an attendant specialised input/output unit. The functions of these building blocks are briefly as follows:

The multiplexer (MXR) contains a demand sorter and switch which transfers characters between the pair of interfaces nearest the computer and the remaining eight input/output pairs. In so doing it adds three bits of address for inward transfers and strips off three bits for outward transfers. If desired 73 multiplexers could be stacked together in three levels (1 + 8 + 64) to give the 512 input/output interfaces corresponding with a nine bit address field.

The communications units comprise the Master and Slave Line Terminals connected by a serial duplex data link. A continuous interchange of serial characters occurs between the line terminal in a handshake fashion analogous with that occurring between the Acceptor and Source Control signals of the parallel Interface. As a result, parallel eight-bit characters presented to the Acceptor interface of either terminal are transferred to the Source interface of the other. The communications units allow the other units of the system to be distributed, as required, about a site such as NPL. The serial transmission of data allows coaxial or twisted-pair cables to be used instead of multiple core cables for all long connections.

The peripheral control unit PCU matches the true B.S. Interface (used for connections to subscriber's peripheral devices) to the Network Standard Interfaces used

for the other units: it also inserts status characters which are used to communicate with the message switching computer - these cause a program interrupt through the input/output unit and are used, for example, when establishing connections between terminals. The data characters passing across the true B.S. Interfaces between the network and the subscriber's equipment are, of course, passed right through the network to the subscriber's equipment at the chosen remote terminal. They do not affect, and are not affected by, the network.

By using the Network Standard Interface in the manner described, all engineers engaged on the development of the network units have a common understanding of the interconnection problems, particularly as similar internal logic can be used for some parts of each unit. In addition, voltage level conversion circuits may be fitted temporarily so that standard B.S. Interfaced readers, punches and computers can be used for testing the individual network units during development and production. This is proving an extremely powerful method of checking the operation of the units and collections of units of a relatively complex network of data links and computers.

Acknowledgement

The work described above has been carried out at the National Physical Laboratory.

References

1. B.S.4421 : 1969 A digital input/output interface for data collection systems. British Standards Institution, Sales Branch, 101 Pentonville Road, London, N.1.
2. Costrell, Louis. Standard nuclear instrument modules. United States Atomic Energy Commission, Washington, D.C. 20545
3. Scantlebury, R.A. A model for the local area of a data communication network - objectives and hardware organization. ACM Data Communications Symposium, Pine Mountain, Georgia, October 1969.
4. Wilkinson, P.T. A model for the local area of a data communication network - software organization. ACM Data Communications Symposium, Pine Mountain, Georgia, October 1969

5. Barber, D.L.A., Blake, D.V. and Scantlebury, R.A. Implementation of the British Standard Interface for data exchange between Sources and Acceptors of digital data. Colloque International sur la Téléinformatique, Paris, March 1969.

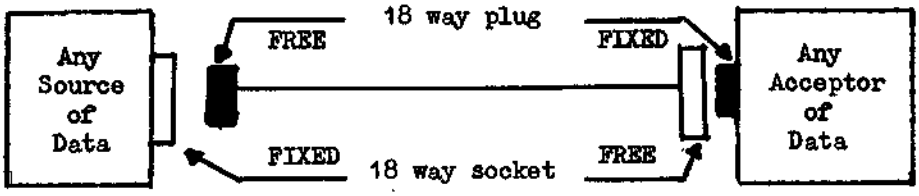
6. Barber, D.L.A. and Blake, D.V. The implementation of the British Standard Interface at NPL. NPL Com.Sci. Report 38, May 1968.

7. Barber, D.L.A., Scantlebury, R.A., and Bartlett, K.A. The use of the British Standard Interface with digital computers. NPL Com.Sci. Report T.M. 20, January 1969.

8. Commité Consultatif International de Télégraphique et Téléphonique, Blue Book Vol. VIII, Data Transmission

9. Data Communications and Interrogation. Technical Publication Service, I.C.L. Ltd., I.C.L. House, Putney, London, S.W.15.

THE BRITISH STANDARD INTERFACE



THE INTERFACE LINES

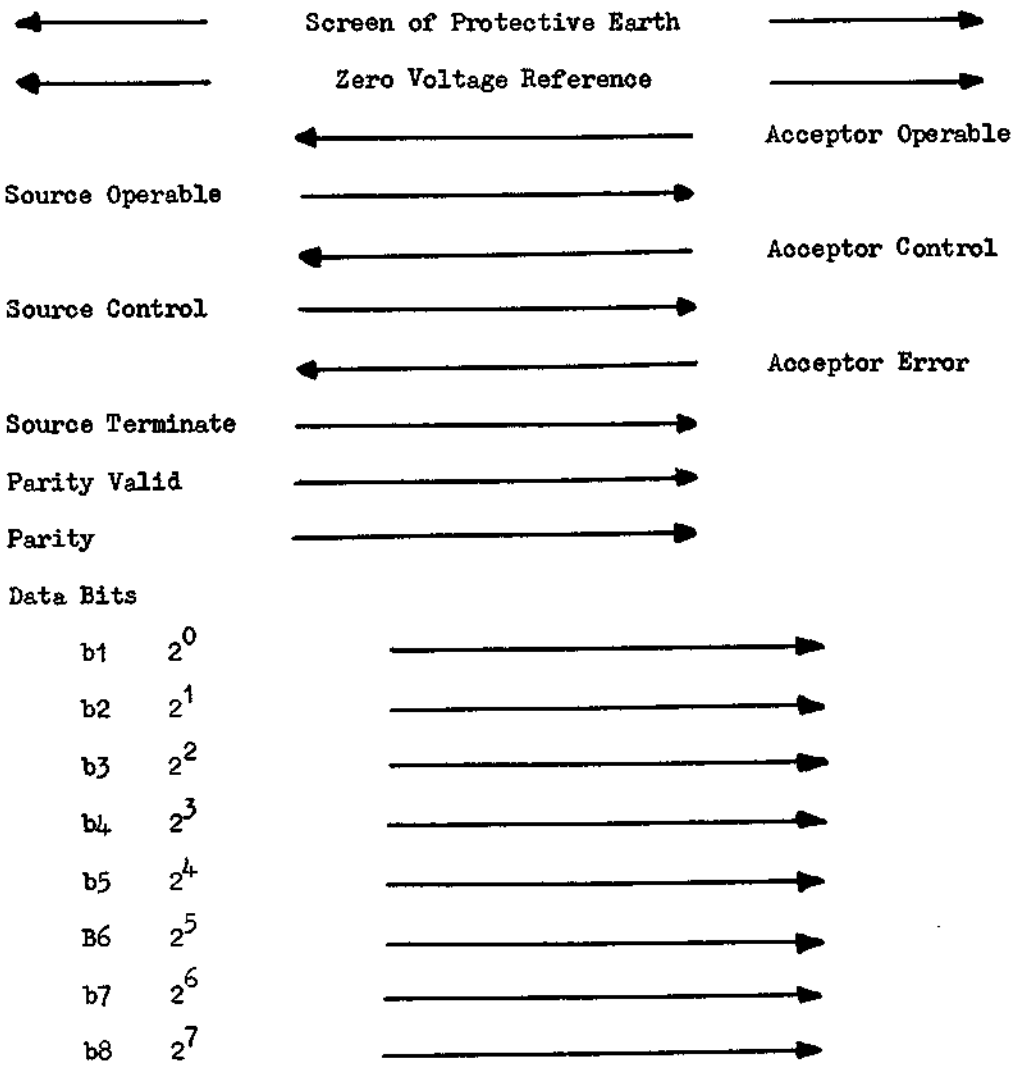


FIGURE 1

THE HANDSHAKE METHOD OF CONTROLLING DATA TRANSFER

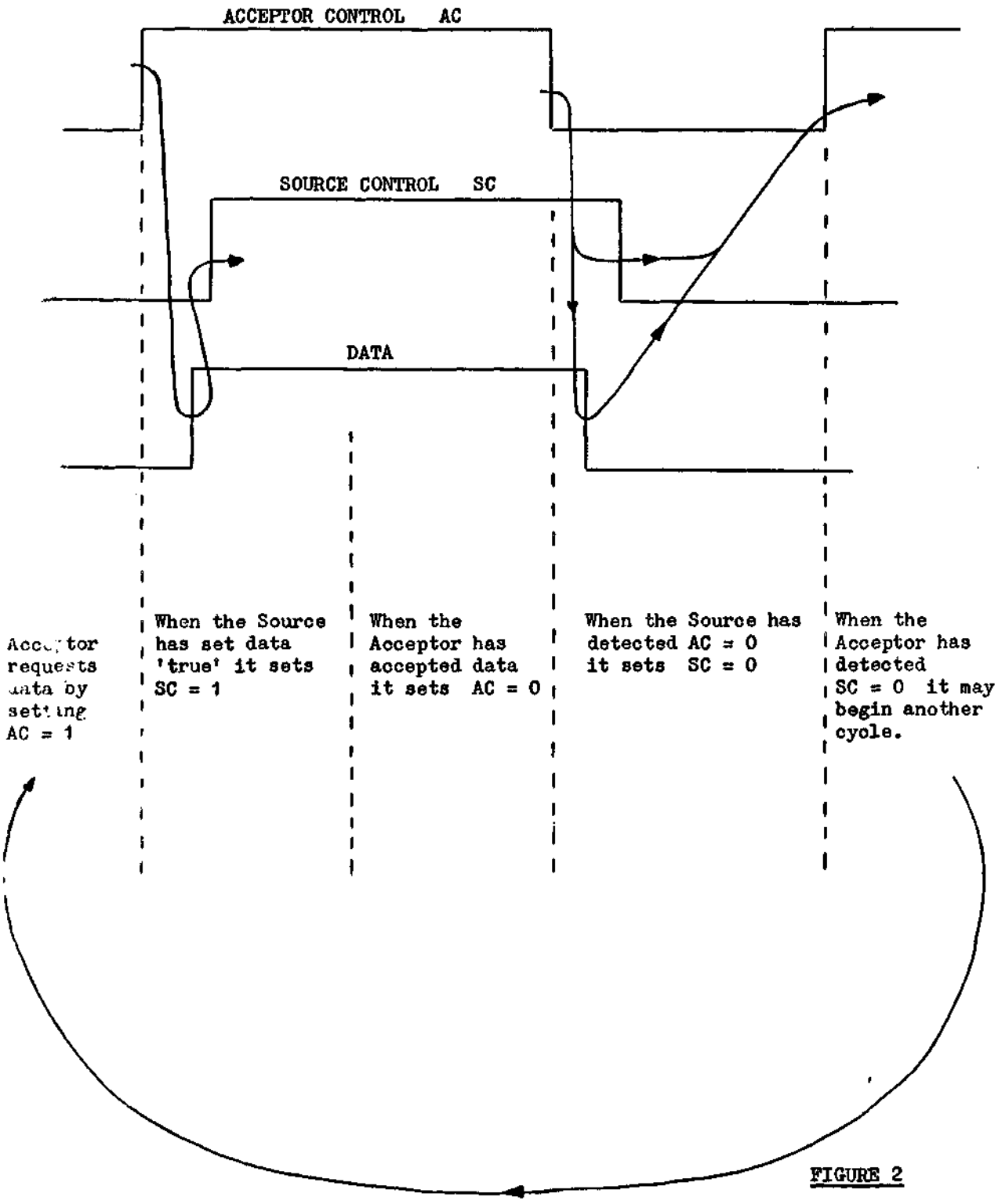
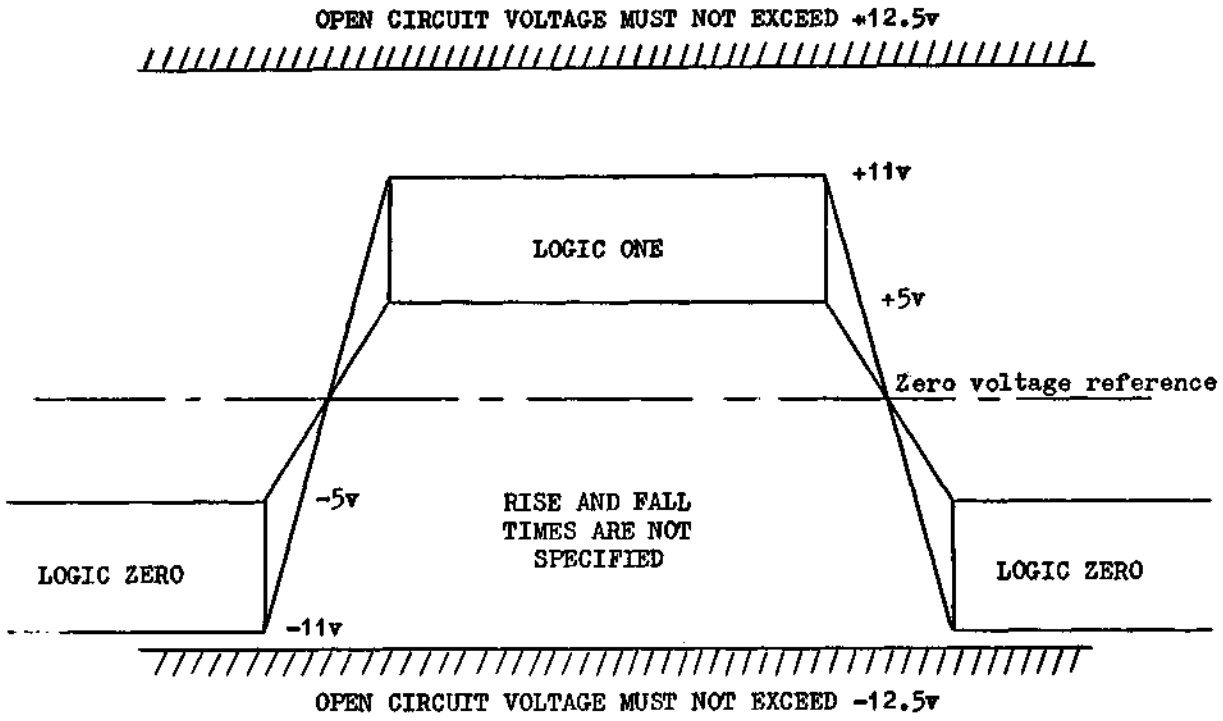


FIGURE 2

ELECTRICAL PARAMETERS

OUTPUT SIGNALS



INPUT CHARACTERISTICS

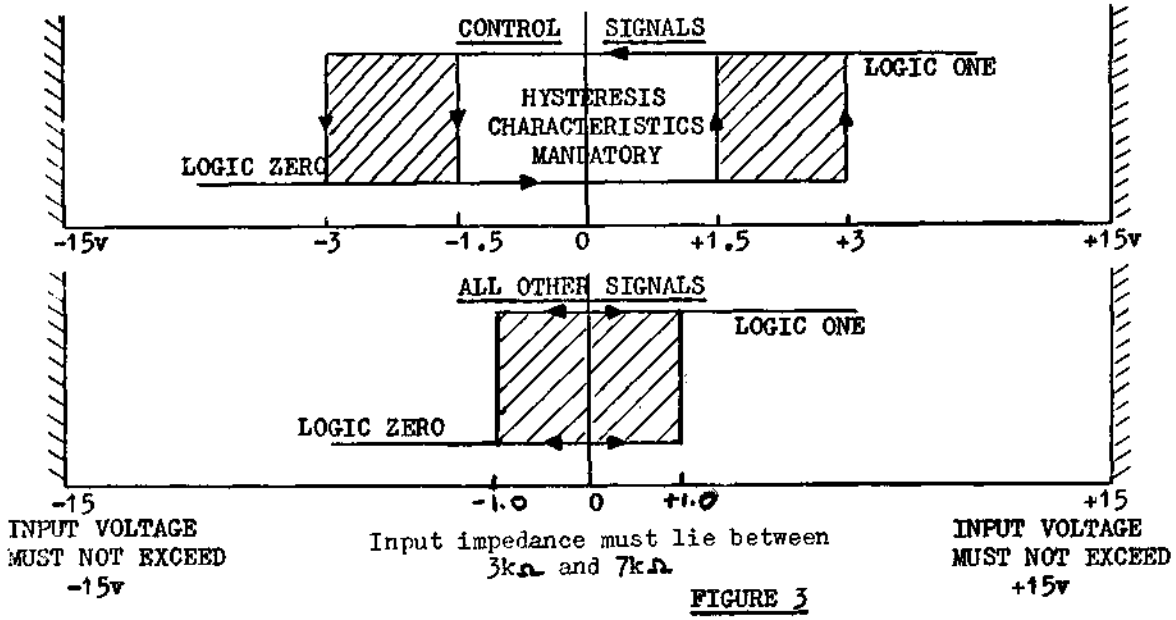


FIGURE 3

Connection to Input/Output bus via specified connectors

Source	Line	Acceptor
Interrupt on failure (mask if not required)	← AO	Power on or logic one
Power on or logic one	→ SO	Interrupt on failure (mask if not required)
Suitable control logic circuits	← AC → SC	Suitable control logic circuits
When AE → 1 inspect AE before fetching data and setting SC → 1	← AE	Set by program after error check failure, before setting AC → 1; clear by SC → 1
When AC → 0 data may be fetched. When AC → 1 set SC → 1	AE NOT USED	AC may be set to 1 without waiting for a parity check
From eleven bits of data busbar: 2^{10}	→ ST	2^{10}
2^9	→ PV	2^9
2^8	→ P	2^8
2^0	2^0	2^0 — To eleven bits of data busbar
.	.	.
.	.	.
.	.	.
.	.	.
2^7	2^7	2^7

FIGURE 4

Connection to Direct Access Store Controller
via Extended data width connectors

Source	Line	Acceptor
Interrupt on failure (mask if not required	← AO	Power on or logic one
Power on or logic one	→ SO	Interrupt on failure (mask if not required)
Suitable control logic circuits	← AC → SC	Suitable control logic circuits
When AE → 1 Inhibit next transfer. Interrupt to repeat block (or character) before setting SC → 1	← AE	Set by program after block (or character) error check failure; clear by SC → 1
Use 'End of range' to set ST → 1	→ ST	When ST → 1 Interrupt to initiate error check and set AE before setting AC → 1 to call next block
Use optional i.e. when store parity available present P and set PV → 1	→ PV → P	Use optional i.e. when store parity can be used read P if PV → 1
Complete word presented on one connector	2^0 ... 2^7 ... 2^n	Complete word received on one connector

FIGURE 5

THE DATA NETWORK MODULES

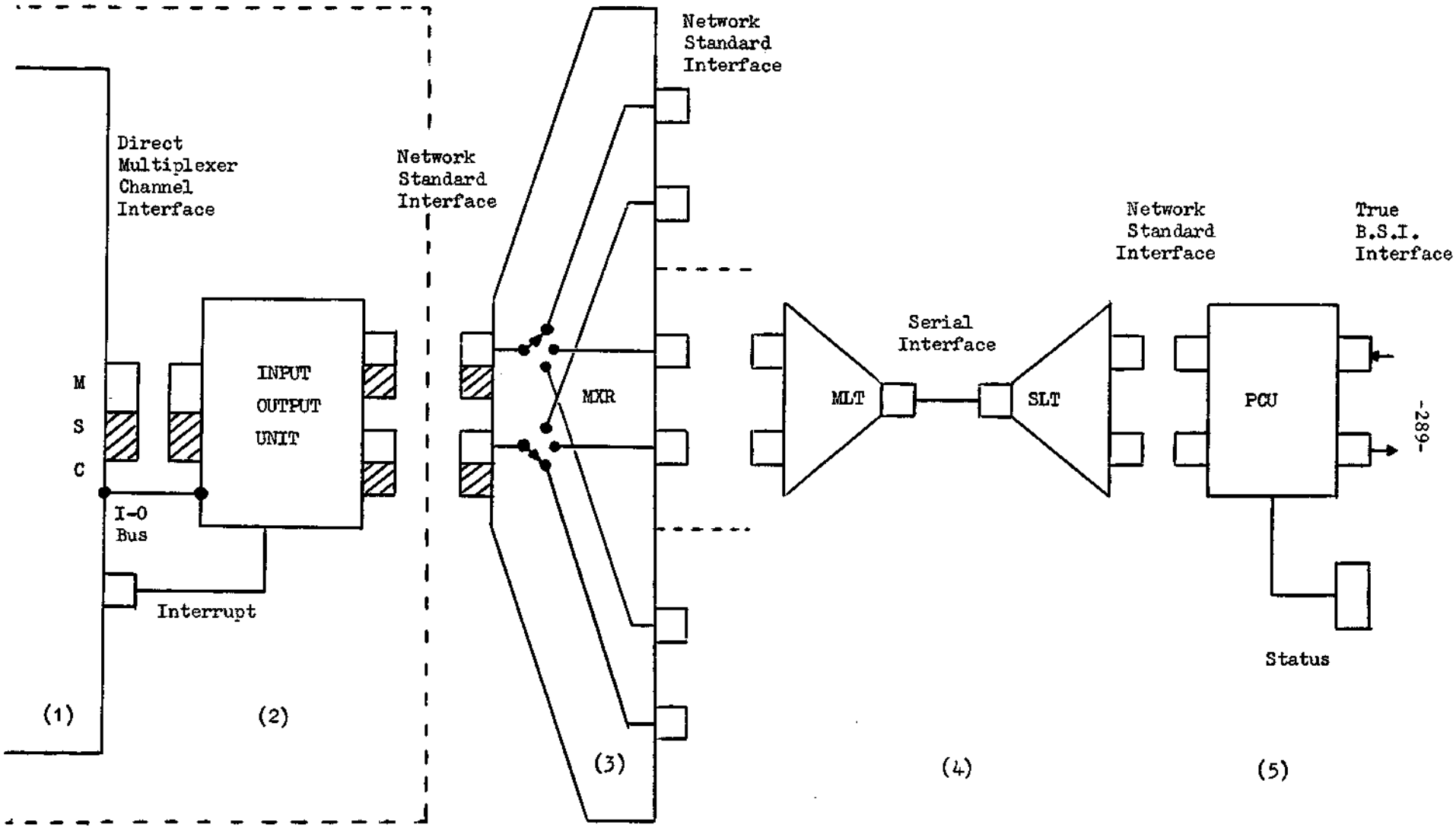


FIGURE 6