Cambridge Ring

Portable 'ring main' system developed at Cambridge

By Peter Hewitt

A PORTABLE minicomputer operating system and processor-speed data ring hardware, developed by a Cambridge University team, could herald a new era of machine independent distributed processing. Both are to be marketed by Toltec Data, a Cambridge-based software house.

A live network using both the operating system, Tripos, and the data ring was demonstrated last week in Cambridge at a one-day user group meeting devoted to BCPL (see Software File, page 7).

BCPL is the high-level system implementation language in which Tripos is predominantly written and from which it derives its portability.

The meeting was arranged by Toltec Data, which is to market Tripos, under the name Trout, and is to begin marketing the ring system in the summer.

The meeting itself provided evidence of the keen interest being shown in BCPL by Siemens, which had four delegates attending from Germany. This could indicate that BCPL is under consideration for the proposed European Systems Language for which the EEC has let the study contract to a joint team from Siemens and CII-Honeywell Bull.

The EEC said last year that the ESL could be an existing language, a development of an existing language, or a completely new language (CW, August 10, 1978).

Designed to take advantage of collapsing processor costs, the Cambridge developments make possible a "ring main" computing system, in which each processor enjoys compatible access to any other device in the network.

Such a system is widely seen as an ideal campus network, providing each student or researcher with a low cost personal computer that can access more powerful hardware and software in the network as required.

An attractive alternative to a large time shared mainframe, it constitutes in effect a local version of such services as Arpanet.

The Cambridge ring system may also have a strong appeal for certain commerical applications, notably where complete portability is required and in distributed database systems.

The operating system component, Tripos, is at present a single user, multi-task operating system that includes facilities for compiling, linking, and debugging BCPL programs, a line editor, and a file handler.

Fortran and Basic are expected to be added by the summer, with Pascal and Algol under consideration for future implementation. There are also plans to adapt the system for multiple users.

The network hardware consists of standard twin-strand cable and a number of Dataring communication ports, one of which is required for each processor, on the ring (CW, September 14, 1978). Data is transferred by the ports at around one Megabit per second and travels round the ring in labelled packets, only interrupting the machine to which it is addressed.

Currently implemented on a printed circuit board and costing about £1,500, the ring port is likely to be available as a hybrid twin chip device in about a year.

In the operational Cambridge ring, Tripos is already running on three processor types: the Computer Automation LSI-4, DEC's PDP-11, and the Data General Nova.

0223 312347

AN INTRODUCTION TO TROUT AND THE DATA RING

The advent of the mini and micro computer has brought with it as many different operating systems and executives as there are manufacturers. In some cases an executive written for one machine is not suitable for another made by the same Company. Any change of operating system (or similar mechanism), and with it the facilities it provides, can be a very big upheaval for a user. The ideal situation is one in which the user can change machines but keep the original operating system. This would avoid the expensive task of modifying programs, reformatting files and retraining staff. It would also provide for complete transportability of programs between users of the same system regardless of the machine it is run on (notwithstanding that one machine is 16 bit and the other 32 bit, although in many situations even this could be allowed to pass justifiably unnoticed).

The TROUT operating system has been designed to provide just such a degree of portability. At present it is a single user multitask system although it will support multi-access for users using shareable re-entrant code such as an editor or a database management system. There are plans to modify the system to multiuser-multitask, but no time scale has been suggested.

The facilities offered by TROUT at the moment are BCPL, a line editor, a file handler and linking and debugging facilities; however, by the summer of 1979 it is hoped that FORTRAN and BASIC will be available. Investigations are being made at the moment into the possibilities of adding ALGOL (60 or W) and PASCAL to the repertoire. Any compiler supported by TROUT is written in BCPL. This means that the compilers as well as the operating system are completely portable.

One very large advantage inherent in the system is that not only is a database immune to exchange of machine, but the database management system is also. If it is possible to remove an entire database system from one machine to another, it is also possible for machines running identical systems to communicate with each other with great ease. This leads to the first step towards a distributed data base. The distribution need not be over just two machines but indeed over many, which may form a ring.

In a system involving several computers interconnected in a closed loop, the situation arises in which a machine may wish to communicate with another that is not adjacent to it. To relieve the machines of the task of handling data that does not concern them, a data ring system has been designed to perform the task for them. Each machine is connected to the next machine using a Dataring communication port. A port not only handles reception and transmission of data, but also contains the address of the machine it serves. Packets of data to be sent around the ring from one machine to another are preceded by a word containing the address of the machine to which they are being sent. The ring communication hardware examines this word. If it does not match the address of that port it transmits it to the port of the next machine in the ring. Throughout the entire procedure the only machine that is ever interrupted is the machine to which the packet is addressed. If it is never accepted by a machine, the transmitting machine is informed of the situation and the packet is checked for errors.

The ring system is due to be marketed in the summer of 1979.

TOLTEC COMPANY PROFILE

The Company's name comes from the Toltecs, a trading nation with a characteristic rigorous geometric architecture, who occupied present-day Mexico before the Mayan civilisation. Their motto was:

"No problem too difficult to solve,
No place too far to trade".

Toltec Data Ltd accepts the obligations this name commands. The Company was formed with the aim of producing bespoke computer systems for people whose computing problems were not satisfied by the major computer manufacturers' off-the-shelf products. In its first three years of operation, the Company's activities have centred on scientific and process control applications. In addition, Toltec takes original ideas from University research and applies to them its technical and financial resources to produce equipment that will hold its own in the competitive world of industry and commerce.

From its small beginnings in the back room of a country cottage Toltec has grown rapidly and is now housed in a modernised 17th century building in the heart of Cambridge. The Company provides an efficient, flexible service for adapting computers to the clients' needs, specialising in the development of innovative software and the matching of novel peripherals to central processors. In its three years Toltec has designed and installed a wide range of sophisticated and cost-effective systems covering a broad cross-section of industry, commerce and scientific laboratories. The fields in which Toltec is active include acoustics, medical physics,

crystal plating and signal correlation. In addition, the Company has undertaken development work in both hardware and software, and has also collaborated with the German Aerospace Research Centre (DFVLR) in a research project into the noise problems associated with the German Federal Railways' high speed train.

Some of the Company's "one-off" projects were so successful that they are now able to be produced rapidly on a large scale. We shall be setting up separate companies in these fields to handle production and marketing. It is our intention to export by setting up separate overseas companies. In this way we feel confident that we can provide a direct and therefore better service to our customers.

Toltec places great emphasis on the efficient maintenance of their systems once they have been designed, built and commissioned, and for this reason the Company has set up its own maintenance organisation to provide customers with a prompt, expert service.

Paul Reeve ABCS - Chairman and Managing Director

Paul has more than 20 years of diverse commercial experience. His career started in company analysis with a firm of stock-brokers, followed by a period of general management during the reorganisation of the company. He then moved into Sales and Marketing with companies concerned with telecommunications, advanced technology and computer hardware. His innovative urge was frustrated by the constraints of large company structures and this led him to set up his own company which was to incorporate the flexibility required in a rapidly changing industry.

Peter Corfield MIL - Director

Peter has experience of a wide variety of commercial environments and an in-depth knowledge of several languages, having lived for many years in Germany. He is responsible for marketing activities world-wide.

Alan Martin MA (Cantab) - Director

Alan is a Cambridge Physics graduate who has specialised in radio astronomy both at Cambridge University and the Max-Planck-Institut in Bonn. Over the past few months, he has carried out major enhancements to the Acoustic Telescope software, and now has responsibility for the continued development of the A-T hardware and software. In addition, he has responsibility for scientific applications software.

John Moughton MA (Cantab) - Director

John, a Cambridge Physics graduate, was principal author of the Interactive-16 system and was responsible for its application to the Acoustic Telescope and to the University Library reader identity card system. His role is overall project supervision from quotation through resource allocation, including any special hardware design, to installation and support.

Valerie Reeve - Director

A principal shareholder, Valerie brings skill, energy and total commitment to the health of the Company. She brought in the range of sophisticated internal accounting and stock control systems, and her experience in both University and commercial fields makes light of the day-to-day demands of a growing company.

Mike Roberts MA (Cantab) - Director

Mike, who has a Cambridge degree in Natural Sciences, is a specialist in crystallography. He has been responsible for developing an on-line system for testing high precision quartz crystal oscillators. Mike is also responsible for all Toltec's present and future graphics and image processing systems.

Adrian Koe - Company Secretary

Adrian's background in O & M, insurance and PR work makes him ideally suited to exercise control of Toltec's administrative systems. In the course of the Company's third financial year, he is introducing further improvements in the Company's organisation.

John Brunney Dip Comp Sci (Cantab), AIDPM, MBCS - Divisional Manager (Toltec Executives and Compilers)

John followed his academic career with extensive and varied programming experience with particular emphasis on data-base work in aviation, medicine and commercial systems. He is in charge of writing a portable operating system for 16-bit mini and micro computers.

Brian Cappell BA (Cantab) - Project Manager

Brian is a Cambridge graduate in physics and computer science, with a background in software research. He is currently producing an improved radiotherapy planning package.

Ginny Fearon AIL - Technical Administrator

Ginny has a combined commercial and technical background, has travelled widely and knows several languages. She looks after the Company's stock control and ordering activities.

Anne Howard BA (Cantab), AIQPS - Assistant to the Directors

Anne is a highly qualified and invaluable member of the administrative team; in addition to the normal secretarial activities, she handles technical documentation, invoicing and media sales.

Matthew Soar - Technician

Matthew has an electronics background. He carries out repairs, puts together prototype boards and provides general support to the technical team.

Debbie Upton - Receptionist

Debbie handles communications with the outside world and looks after our visitors.

John Van Der Heijden - Maintenance Manager

John has an HND in Electrical and Electronic Engineering from the Cambridge College of Arts and Technology, combined with several years in the electronics industry; he provides comprehensive maintenance support to our customers' installations.

Consultants to TOLTEC

Dr Martin Richards - Computer Laboratory,
University of Cambridge

Dr K C A Smith - Engineering Department,
University of Cambridge

David Juett - Physics Department,
Addenbrookes Hospital, Cambridge

Comparison of an Ethernet-like communication system with the Cambridge ring

W.D. Shepherd, B.Sc., G.S. Blair, B.Sc., and D. Hutchison, B.Sc., M. Tech., M.B.C.S.

Indexing terms: Cambridge ring, Ethernet, Local area networks, Simulation

Abstract: This paper describes the simulation and comparison of two types of local-area network. One is a branching broadcasting system, based closely on the standard Ethernet, requiring contention-resolving hardware, and the second is the Cambridge digital communication ring, a point-to-point communication system using the empty-slot principle for deterministic multiplexing.

1 Introduction

One of the most interesting and relevant areas of computer research today is the development of local-area communication networks. As the name implies, these are data-communication networks, typically packet communication networks, confined within a particular site or set of buildings. They aim to provide high-bandwidth communication (greater than 1Mbit/s) over an inexpensive transmission medium and with low error rates.

Local-area networks are also characterised by the wide range of environments in which they are required to work, and the large number of different devices that may be attached to them. This means that the underlying architecture must be flexible enough to support these different applications. A good introduction to local-area networks can be found in the paper by Clark *et al.* [1] and an extensive bibliography is given in Reference 2.

The two dominant architectures used are the broadcasting and the ring or loop system. Two well known examples of these are the Ethernet system developed by Xerox at Palo Alto and the Cambridge ring. Both are being used successfully in a number of different environments.

Much work has been carried out into the performance of broadcast local-area networks, notably by Kleinrock and Tobagi, and Tobagi and Kleinrock [3, 4]. Hopper [5], who was involved in the development of the Cambridge ring, has developed analytic models to compare the performance of different ring systems and, separately, different broadcast systems. A paper by Almes and Lazowska [6] outlines an analytic study of an Ethernet-like network, the results being verified by a simple simulator model. Practical work has been carried out by Schoch and Hupp on measuring the performance of an Ethernet local-area network [7]. However, a question that the earlier work leaves largely unanswered is 'how do broadcast and ring networks compare with one another?' At first sight it might seem tempting to compare well known implementations of the two schemes, e.g. the prototype Cambridge ring and the DEC/Intel/Xerox Ethernet, but the relevance of such a comparison would be limited because the schemes may have been designed with different overall requirements.

Our objective was therefore to compare an Ethernet-like system and a slotted-ring system under conditions as similar as reasonably possible. Departures from the specification of the two systems are highlighted in Section 3. The purpose of the comparison is to throw light on the various factors, e.g. traffic statistics, that could make one scheme preferable to the other.

The paper is divided into three parts. In the first part, we give a brief description of the Cambridge ring and the Ethernet system. In the second, we define the versions of the two systems that have been simulated and describe the construction of the two simulator models. Following this, we give details of the experiments that have been carried out, together with their results in graphical form, and finally we draw a number of conclusions based on these results.

Paper 1808E, first received 22nd December 1980 and in revised form 2nd December 1981

The authors are with the Department of Computer Science, University of Strathclyde, Glasgow, Scotland

2 Cambridge-ring and Ethernet systems

In this Section we give a brief description of the Cambridge-ring and the Ethernet system. Detailed descriptions of the two systems can be found in the papers by Wilkes and Wheeler [8] and Metcalfe and Boggs [9];

2.1 Cambridge ring

The Cambridge ring makes use of the empty-slot system. That is, a number of fixed-length empty packets circulate unidirectionally around the system. If one station on the ring wishes to send a message to another, it enters the source and destination in the appropriate part of an empty packet and the message in the data part of the packet. The structure of a typical ring is shown in Fig. 1 and the packet layout in Fig. 2.

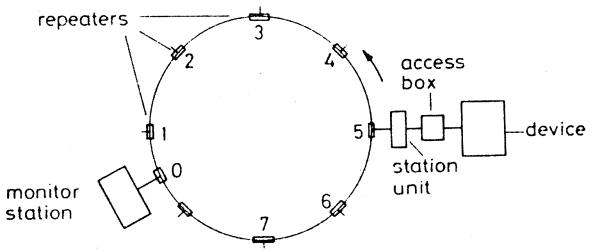


Fig. 1 *Cambridge ring*

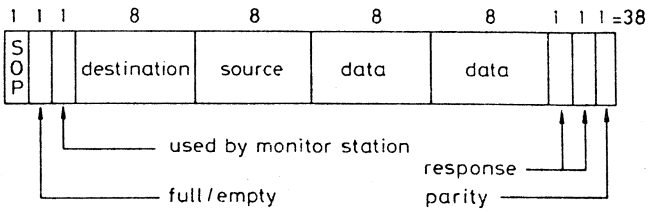


Fig. 2 *Ring packet layout*

The Cambridge ring consists of a number of stations, as shown in Fig. 1. Each station consists of a repeater and a station unit which are identical for all stations, apart from a plug whose internal wiring specifies the station's address. In addition, an access box provides the interface between the device and the station. A unique station called the monitor station is used to create the slot structure on initialisation and to clear corrupted packets.

The packet structure is shown in Fig. 2, and is designed to minimise delay at the transmitter and receiver. The first bit is used for synchronisation and the second indicates whether the packet is full or empty. At present, with the delay at each station being small, the number of packets circulating is only two, and this requires a shift register of about 20 bits in the monitor station to artificially lengthen the ring.

The station unit contains three registers: transmission shift register, receiving shift register and source select register.

0143-7062/82/040147+09 $01.50/0

If the source select register contains all ones, the station will listen to packets from every source, if it is zero it will be deaf to all sources; otherwise, the station will receive only from the source whose address is in the register. On the receiving side, a circuit continuously monitors the data stream for packets addressed to the station. On detecting one, provided the receiving register has been cleared after previous transactions, the digit stream will flow into the receiving shift register. If the register has not been cleared, the control bits in the incoming packet are marked 00, signifying that the station is busy. In the meantime, the source is checked against the source select register. If the station is deaf to the source the control bits are set to 10; otherwise shifting ceases and the control bits are set to 01, signifying that the packet is accepted. On acceptance, the packet remains in the receiving shift register until it is cleared by a signal from the access box.

A simple scheme is used to prevent any station from 'hogging' the system. On receiving back a packet, a source is not allowed to immediately re-use the slot which contained it. Instead, it must mark the slot as empty and pass it downstream. This has the desirable effect on access to the ring, giving all stations an equal share of transmission bandwidth.

2.2 *Higher-level protocols*

It is important not to sacrifice flexibility and efficiency in a local area network by insisting on a particular protocol for all circumstances. A single packet may be sufficient for some tasks. However, protocols for transmitting a quantity of data are required. The following is a description of a basic block protocol currently being used on the Cambridge ring.

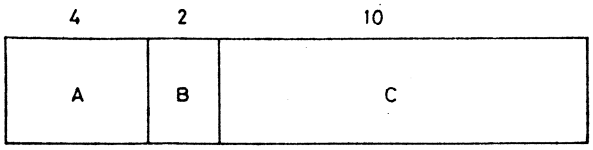


Fig. 3 *Ring basic block header*

A ring basic block commences with a header packet of the form shown in Fig. 3:

Field A is the binary pattern 1001

Field B = 0 long block with checksum
= 1 long block with checksum zero
= 2 this block consists of a single packet carrying data C
= 3 reserved for further expansion (error protem).

A long block consists of route packet, $C + 1$ data packets and a checksum packet. A route packet consists of a port number in the bottom 12 bits, the basic block being notionally directed to that port at the destination station. The remaining four bits are reserved and should be kept at zero.

The $C + 1$ data packets conform to the protocol that is currently agreed to be in use at the port identified in the route packet.

The checksum packet for type 0 blocks consists of a 16-bit end-around-carry checksum over the entire block, including the header packet. In type 1 blocks, the notional checksum packet is sent as zero, and checked to be zero.

We will limit ourselves to describing the mode of transmission. When transmitting the first packet (header) of a block, due allowance must be made for the possibility of the receiving station being busy or unselected owing to its being in the process of receiving a block from another source. Attempts to transmit the header should be maintained for at least as long as the longest possible block can take at the reception station. Any other ring error can be regarded as fatal.

Having transmitted the first packet (header), allowance may have to be made for certain receiving stations to perform certain set up operations for the block; during this time the station will mark itself as 'busy' and reject any further packets addressed to it.

After that, the number of busy rejects that may be expected per packet should be very low, as the receiver is supposed to be concentrating on one source only. It will be necessary for transmitting stations to have a time out or repeat count on a per-packet or per-block basis, in order to recover from a receiving station 'crashing' in the middle of a block. A time out is necessary to recover from certain ring errors, such as 'power off', which result in a packet never returning to its sender.

2.3 *Ethernet*

The following description applies mainly to the DEC/Intel/Xerox specification for an Ethernet system operating as 10 MHz [10].

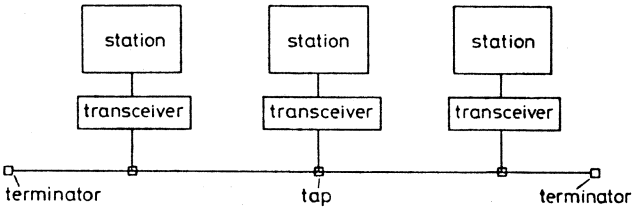


Fig. 4 *Ethernet-type system*

An Ethernet system consists of a number of stations tapped on to a shared coaxial cable, the 'ether', one possible configuration being shown in Fig. 4. Any number of 'ether segments' may be connected through repeaters, allowing a system to grow incrementally throughout a building. However, only one path may exist between any source-destination pair and, therefore, the topology is constrained to be an unrooted tree. Terminators are required at the end of each segment to prevent reflection of signals.

As opposed to the active slot structure of the Cambridge ring, the ether is passive in nature. A station wishing to transmit assembles the data into a packet, Fig. 5, and listens to the ether for carrier, deferring to passing traffic.

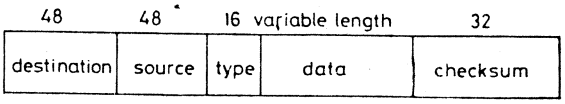


Fig. 5 *DIX packet*

When the ether is thought to be idle, after delaying for a suitable period (interframe spacing) to allow everything to settle, the station broadcasts its packet. For a system operating at 10 MHz, it is essential for synchronisation that the transmission is preceded by a preamble.

This scheme allows the possibility that two or more stations may transmit packets at the same time. Such packets are said to collide. The susceptible period for collision is until all other stations are hearing carrier and hence deferring, and is at most one end-to-end propagation delay of the physical channel. Collisions are detected by hardware in the stations and result in the detecting station transmitting a jamming signal forcing all stations involved to abort and reschedule their transmissions in accordance with the binary exponential back-off algorithm [9]. This algorithm selects a random time to reattempt transmission, this depending on the number of collisions suffered so far by the packet.

A successful transmission will eventually propagate to all stations. Any station can copy a packet into its buffer space, but normally only the station to which the packet is addressed

will do so. Unlike the Cambridge ring, this scheme does not automatically provide the sender with a positive indication that his packet has been received, and so higher-level protocols are needed to allow for this.

The DIX specification defines an Ethernet with the following parameters:

(i) frequency : 10 MHz
(ii) preamble : 64 bits
(iii) interframe spacing: 9.6 — 10.6 μs
(iv) jamming signal : 32 — 48 bits
(v) packet size : 64 — 1518 octets.

3 Simulators

In this Section, we define the versions of the Ethernet and Cambridge ring system which we have simulated, and also describe the method of simulation. It is important in the design of a simulator to avoid emulating the system, i.e. seeking to simulate exactly every small detail. This would be extremely expensive in terms of development and running time. However, too little detail could result in misleading results. We felt that the success of the simulation models was dependent on including just those features that have a significant influence on the results.

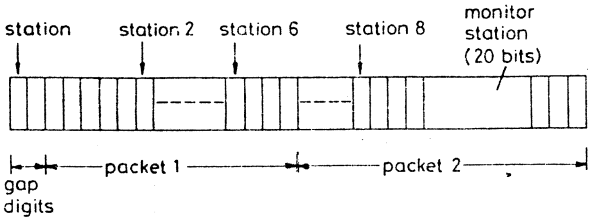


Fig. 6 *Snapshot of ring*

We also decided that, to obtain valid comparisons between the two sets of results, the number of variable factors would have to be restricted. For example, both simulator models operate under the same message transfer protocol, which is not true in the actual implementations. What we are comparing, therefore, are two systems having the main characteristics of the ones on which they are modelled, but not conforming to actual implementations in every detail.

3.1 Environment

Both simulators are designed so that they can be subject to the same environment; i.e., the statistical nature of the workload presented at each work station is parameterised. The individual jobs generated by the workload must be compatible in nature with the highest level of protocol implemented in the simulation models and, therefore, to achieve this the workload generates the arrival of messages which are transmitted by a basic block protocol. The statistical distribution of the messages is determined by two parameters:

(i) the mean waiting time between successive messages
(ii) the length of each message.

This is specified separately for each station.

Each station will then be presented with a stream of messages of the given length whose interarrival time follows a negative exponential distribution with the given mean time. This approach, use of constant message lengths, although a simplification, allows the system to be tested under the various workloads possible in computer application. For example, short bursty traffic from a terminal can be modelled adequately by the negative exponential distribution with a message length of 16 bytes. A disc station could have a message size considerably larger. The benefits of a more sophisticated workload scheme are debatable, as the main requirement is to vary load rather than being concerned about its source.

3.2 Cambridge ring simulator

The system chosen to be simulated is shown in Fig. 1. Eight stations are included, 100 m apart. These stations are given addresses 1 to 8. The monitor station is inserted halfway between stations 8 and 1. These physical parameters were chosen to match, approximately, the prototype ring system constructed in Cambridge and also the Ethernet system being constructed at Strathclyde University. It is assumed that the station logic can drive the ring at 10 MHz.

Two 38-bit packets (as shown in Fig. 2) continuously circulate round the ring with two gap digits at the fore. This requires an additional shift register of 20 bits, as shown in Fig. 6.

This gives the ring a propagation delay of 7.6 μs, in which time 4 data bytes maximum can be carried. Therefore, actual capacity is limited to 4.21 Mbits/s. All stations, on receiving a data packet, become 'busy' for 8 μs.

3.3 Ring protocol

Only one level of protocol is implemented on top of the packet protocol of the ring, this being the basic block protocol (Fig. 8). The packet protocol is as explained in Section 2.1, with the exception that all stations are assumed to be present and, therefore, 'destination absent' is not a possible response. The basic block protocol is based on that outlined in Section 2.1.

To initiate a transfer, a header packet, giving the number of data packets to be expected, is transmitted repeatedly until one arrives back at the source marked 'accepted'. When this happens, the destination will set its source select register to receive from that source only and a one-to-one conversation is set up. When this state is reached, a route packet is transmitted giving a port number for the information. Then the data packets are transmitted followed by a checksum. The checksum sets the source select register to the null value. On transmitting the checksum, the source puts a description of the message into one of eight buffers and seeks another job.

On receiving the checksum, the destination transmits either an ACK or a NACK, the simulated error rate being 1 bit in 10^7. On receiving this reply, the source removes the descriptor from the buffer space and, if it was an ACK, terminates the job. Otherwise, it reschedules the job which must join the queue and recompete for the destination. Discussion of the method of simulation is deferred until later.

3.4 Ethernet simulator

The simulated system follows the DIX standard in most aspects and has the following parameters:

(i) number of stations = 8
(ii) propagation delay between stations = 6 bits (approx. 100 m)
(iii) preamble = 64 bits
(iv) jamming signal = 32 bits
(v) interframe spacing = 9.6 μs.

The configuration is shown in Fig. 7.

However, there are two significant departures from the DIX standard. Firstly, the source and destination fields are each 8 bits long as against 48 bits in the DIX standard. It seemed more valuable to compare systems with similar addressing capabilities and, in any case, for most applications a 48-bit address space is rather excessive.

Secondly, the minimum packet size is not simulated. This limit need only be greater than the largest possible fragment

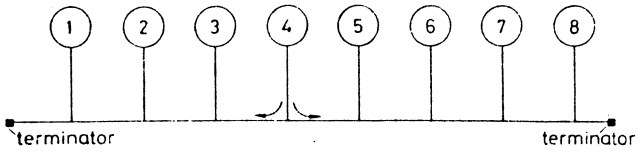


Fig. 7 *Simulated Ethernet system*

resulting from a collision, i.e. one end-to-end propagation delay expressed in bits. In the small system under consideration here, the fixed overhead of message exceeds this limit.

Clearly, if either the above were enforced, performance would suffer and this should be taken into account in the comparison.

3.5 Ethernet protocol

A basic block-transfer protocol is simulated, based on the Cambridge ring. This is done in preference to using the Ethernet file transfer protocol in order to eliminate one variable from the experiments. The format of a basic block is shown in Fig. 8.

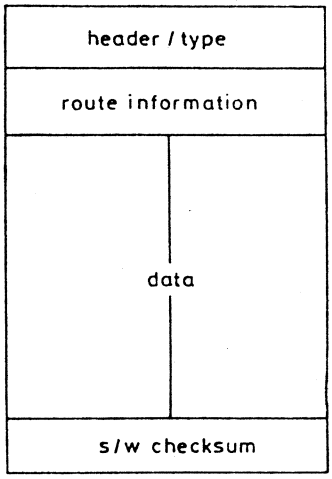


Fig. 8 *Basic block*

A basic block is transmitted in one Ethernet packet with the header included in the type field, the software checksum omitted in favour of the packet checksum and the rest of the block transmitted in the data segment.

The packet structure carries an overhead of 64 bits and the basic block has an overhead of 26 bits. Therefore, excluding the preamble, a transmission has a total overhead of 80 bits. This overhead is fixed and, for longer packet sizes, becomes less significant.

On transmitting a basic block, a station inserts a descriptor into one of a pool of eight buffers and awaits a reply. The destination on receiving a basic block will transmit an ACK or an NACK, the simulated error rate being 1 bit in 10^7.

On receiving a reply, the source station deletes the descriptor. If the reply is an ACK, it terminates the transfer; otherwise, it schedules the message for retransmission. Replies are assumed to be transmitted correctly.

3.6 Method of simulation

Both simulators have the same structure, consisting of the following four phases: namely, (*a*) parameterisation, (*b*) workload generation, (*c*) simulation and (*d*) analysis:

(*a*) *Parameterisation*: During this phase, the run time of the simulator and the statistical characteristics of each station are input. The run time of the system is used by the simulation phase, and a statistical table is passed to the workload generator giving basic block length and interarrival time between basic blocks.

(*b*) *Workload generation*: A basic block is characterised by a source station, destination, length and start time. The generator, knowing the source station and the previous basic block sent by that station, generates a basic block whose destination is chosen randomly from among the other seven stations, its length from the statistical table, and its creation time so as to comply with the negative exponential distribution. This phase runs in parallel with the simulation phase.

(*c*) *Simulation* : Associated with each station there are three queues, namely:

(i) A *waiting* queue – basic blocks waiting for transmission.

(ii) An *analysis* queue – basic blocks which have been successfully transmitted by the station

(iii) An *acknowledgement* queue – ACK/NACKs requiring transmission

An element of any of the above queues consists of five fields: destination, length, start time, select time, and finish time. In the *acknowledgement* queue, fields 2 to 5 have null values. The last three fields in the other queues are used in the analysis stage and are updated as the simulation progresses.

(*d*) *Analysis*: This phase uses the information in the *analysis* queues to calculate performance figures for each basic block, for each station and for the system as a whole. Further details are given in Section 3.9.

3.7 Cambridge ring simulation

The simulation phase consists of advancing the two packets alternately, one station at a time round the ring until the required time period has been simulated. The arrival of a packet at a station may result in the transmission or reception of information by that station and /or a change in its state.

A packet is specified by the following information: (*a*) full/empty, (*b*) destination, (*c*) source, (*d*) data (used to carry information about the type of packet etc.) and (*e*) response bits. Also associated with each packet is its status given by the current time and address.

On the arrival of a packet at a station, the queues for that station are updated so as to reflect the current situation. In particular, the workload generator is invoked to update the *waiting* queue, and new basic blocks added have their start time field set to the creation time of the basic block. The packets, in effect, continuously circulate round the ring servicing the queues, each basic block being transmitted according to the packet-transfer protocol and the basic-block transfer protocol. The *acknowledgment* queue is served with highest priority.

The 'select time' of a basic block is set to the time at which it is selected from its queue for transmission. 'Finish' time is set to the time when an acknowledgment packet, signifying the successful completion of the basic block protocol, is received. At this time, the completed element in the *waiting* queue is transferred to the *analysis* queue.

The *analysis* queue, on completion of simulation, is passed on to the analysis phase.

3.8 Ethernet simulation

At any given time, a station can be considered to have a current job to perform. This may be null if both queues are empty; otherwise the current job is defined by:

(i) type – ACK/NACK or message

(ii) a count of the number of collisions

(iii) time for the next attempt at transmission.

At the end of each attempted transmission, the simulation progresses by choosing the station which will be the next to attempt. This is a complex task, and, to avoid simulating idle periods, involves the look ahead by one element in the workload. This is examined if the current job is null. Having chosen the first attempt to transmit, queues and current job status for the different stages are updated to the current time. If new messages are added, the start time is set from the workload information. If a message is selected from a *waiting* queue to be the current job, the select time is also set.

The simulator then attempts to acquire the ether for the chosen transmission by checking for collision. This is simulated by checking stations to left and right, one step at a time, to see if the next attempt by that station will occur before the first bit of the message reaches that station. If this condition is encountered, a collision is deemed to have occurred. This check is carried out for all stations, the first station detecting a carrier being deemed the jamming station. As each station detects a collision, a retransmission algorithm simulating the binary exponential backoff is used to alter the 'next attempt' field defining that job, and the collision count is incremented.

If no collision is detected, the ether has been acquired and a transmission period is entered; otherwise the jamming station jams the line and the transmitting station calls the retransmission algorithm.

At the end of a transmission period, or, if a collision occurred, at the end of the jamming signal, a deference check must be carried out. This is done by going left and right one step at a time from the transmitting/jamming station simulating the end bit of the message/jamming signal. At each step the propagation time of 6.0×10^{-7} is added to the time and, if at a station, the next attempt time is found to be less than the current time, deference has occurred. The attempt time must therefore be altered to the current time.

The above procedure is repeated until the required time has been simulated. At this stage, the *analysis* queue is made available to the analysis stage.

3.9 Analysis

The performance figures are derived from the information held in the *analysis* queue. This data structure is the same for both simulators, and so the analysis phase can be common. The fields of interest from the analysis phase are:

(i) start time of message, T_q
(ii) select time of message, T_s
(iii) finish time of message, T_f
(iv) length of message in bits, L

From this information, performance figures are collected for three levels.

(a) *Message level*

Queueing time Q, where $Q = T_s - T_q$
Transmission time T, where $T = T_f - T_s$
Estimate of capacity for that message C, where $C = 8 \times L/T$ (bits/s).

(b) *Station level*

To derive performance measurements for each station, certain figures are accumulated for every message transmitted from that station. These are:

$$S_q = \Sigma Q \quad S_t = \Sigma T \quad S_1 = \Sigma L$$

Also, a count of the number of messages transmitted from that station is kept in N.

The above figures give the following measurements:

Mean queue time $= S_q/N$ (s)

Mean capacity $= 8 \times S_1/S_t$ (bits/s).

(c) *Global level*

The accumulated figures above are then summed for all eight stations as follows:

$$G_q = \Sigma S_q \quad G_t = \Sigma S_t$$

$$G_1 = \Sigma S_1 \quad M = \Sigma N$$

The following global measurements can then be derived:

overall average queue delay $= Gq/M$ (s) . . . . . A

overall average capacity $= 8 \times G_1/G_t$ (bits/s) . . . . . B

average message length $= G_1/M$ (bytes) C

delay for average message $= A + 8 \times C/B$

In addition to the above figures, both simulators provide an estimate for line utilisation.

In the Ethernet simulator, an accumulator sums the amount of time over which meaningful data is being transmitted, the estimate of line utilisation being this figure divided by total simulated time.

In the Cambridge ring simulator, the monitor station records two values:

(i) the total number of packets passing during a simulation, TOT
(ii) the total number of packets which pass marked full, FULL

An estimate of line utilisation is given by FULL/TOT.

4 Results and conclusions

In this Section, the various experiments carried out using the simulators are outlined. Comparisons are made between the Ethernet and Cambridge ring modes of operation and conclusions are drawn.

Graphs of mean wait time against delay show the direct effect on service of varying the workload. Graphs of line utilisation against delay, the standard way of representing performance, are useful to show breakdown points in the system. It should be noted that all graphs have a logarithmic scale for the dependent variable.

4.1 Experiments

Experiment 1

In this experiment, we investigate the performance of the two systems in transmitting messages of length 2 bytes.

All stations transmit 2-byte messages, the mean wait time being varied to obtain a spectrum of results for different workloads.

The delay for a 2-byte message is plotted against mean wait time (Fig. 9*a*) and against line utilisation (Fig. 9*b*).

From Fig. 9*a*, it can be seen that both systems have very similar delay figures with the Cambridge ring slightly better at all levels of load. Both systems degrade steadily as line utilisation increases (Fig. 9*b*).

Experiment 2

In this experiment, a message of length 16 bytes is transmitted. This is the sort of message pattern to be expected from a terminal concentrator, buffered commands being transmitted to a processing unit. Results are collected for varying workloads, delay being plotted against mean wait time (Fig. 10*a*) and line utilisation (Fig. 10*b*).

In this case, Ethernet proves to give consistently lower delay figures. This can be explained by two factors:

(i) In the Cambridge ring, the amount of control information is proportional to the length of the message, whereas in the Ethernet it is constant.
(ii) In Ethernet, longer messages imply that the time wasted by collisions becomes shorter relative to total transmission time.

From Fig. 10*b*, we again see that, in the Cambridge ring, the

delay increases steadily; however, in Ethernet the curve is flatter than in experiment 1 below 30% line utilisation.

Experiment 3

In this experiment, we investigate the performance of the two systems in transmitting messages of length 64 bytes. This could represent the transfer of small files. Delay is plotted against mean wait time (Fig. 11*a*) and against line utilisation (Fig. 11*b*).

The trend emerging in the previous experiments continues. Ethernet now gives substantially lower delay figures, being better by a factor of four at low loading levels. From Fig. 11*b*, it is seen that both systems degrade slowly as line utilisation increases.

Experiment 4

The performance of the two systems in transferring messages of length 128 bytes is shown in Figs. 12*a* and *b*.

There is now a factor of six between the two systems at low loads. One other trend has now emerged. With longer message lengths, delay times for Ethernet are seen to degrade less sharply as line utilisation increases. The Cambridge ring, however, degrades in a similar way, independent of message length. In Ethernet, the lessening impact of collisions with longer messages reduces the snowballing effect of retransmissions.

Experiment 5

Finally, we investigated the claim that the stations transmitting longer messages in Ethernet can hog the system. This obviously cannot happen in the Cambridge ring system, since all messages are transferred in units of 2 bytes.

Stations are deemed to belong to one of two classes:-

class A: transmitting short bursty traffic
class B: transmitting long messages.

It was decided arbitrarily to place stations 3 and 7 in class B, and the rest in class A.

If delay time is independent of message-length distribution, delay times for heterogeneous and homogeneous traffic should be the same for a given total offered load, where total offered load is defined to be the overall arrival rate of messages into the system.

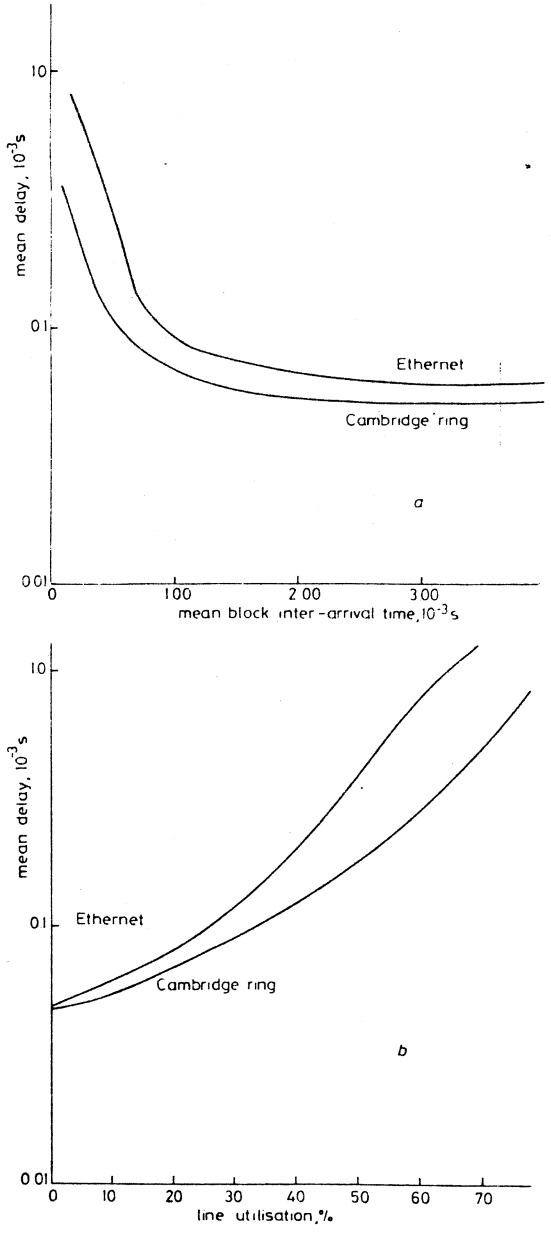


Fig. 9 *Experiment 1*

Packet size = 2 data bytes

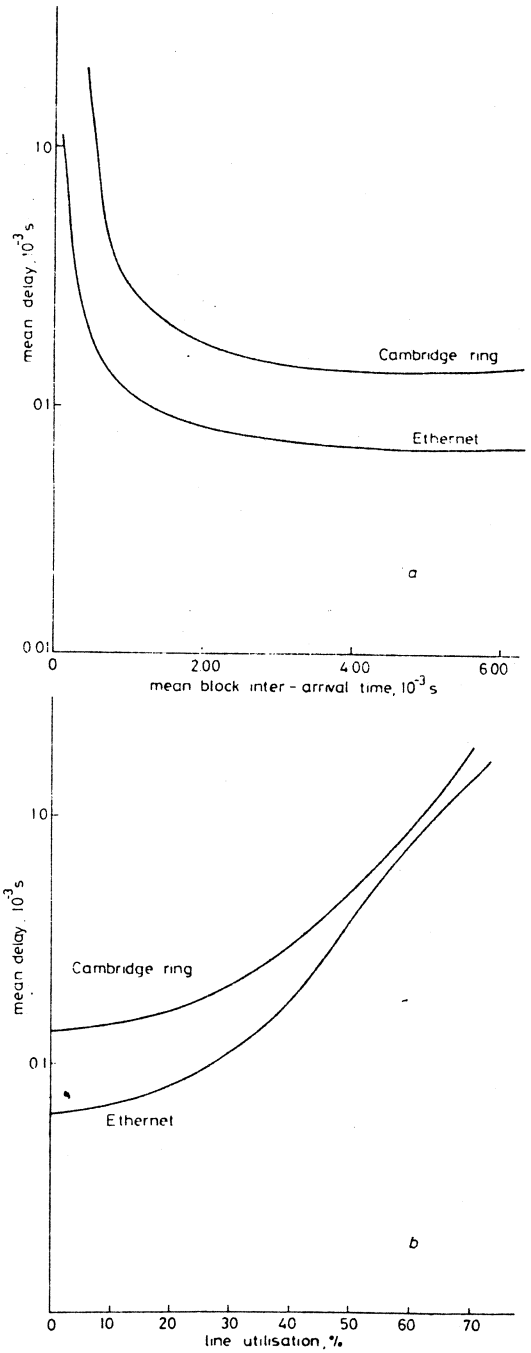


Fig. 10 *Experiment 2*

Packet size = 16 data bytes

Two runs of the simulator were made. Run 1 had class A stations transmitting 16-byte messages with a mean wait time of 8×10^{-4} s, and class B stations transmitting 128-byte messages with a mean wait time of 3×10^{-3} s. This gives a total offered load of 1.64 Mbits/s. The simulated time was 0.1 s.

We are interested in whether the average delays for class A and B are significantly different from that expected for the same total offered load for homogenous traffic. The following figures were obtained:

average delay for class A = 1.6×10^{-4} s
average delay for class B = 2.8×10^{-4} s

For homogeneous traffic to achieve the same total offered load would require mean message interarrival times of 6.2×10^{-4} and 5×10^{-3} s for 16-byte and 128-byte messages, respectively. Reading from Figs. 10*a* and 12*a*, this gives the following

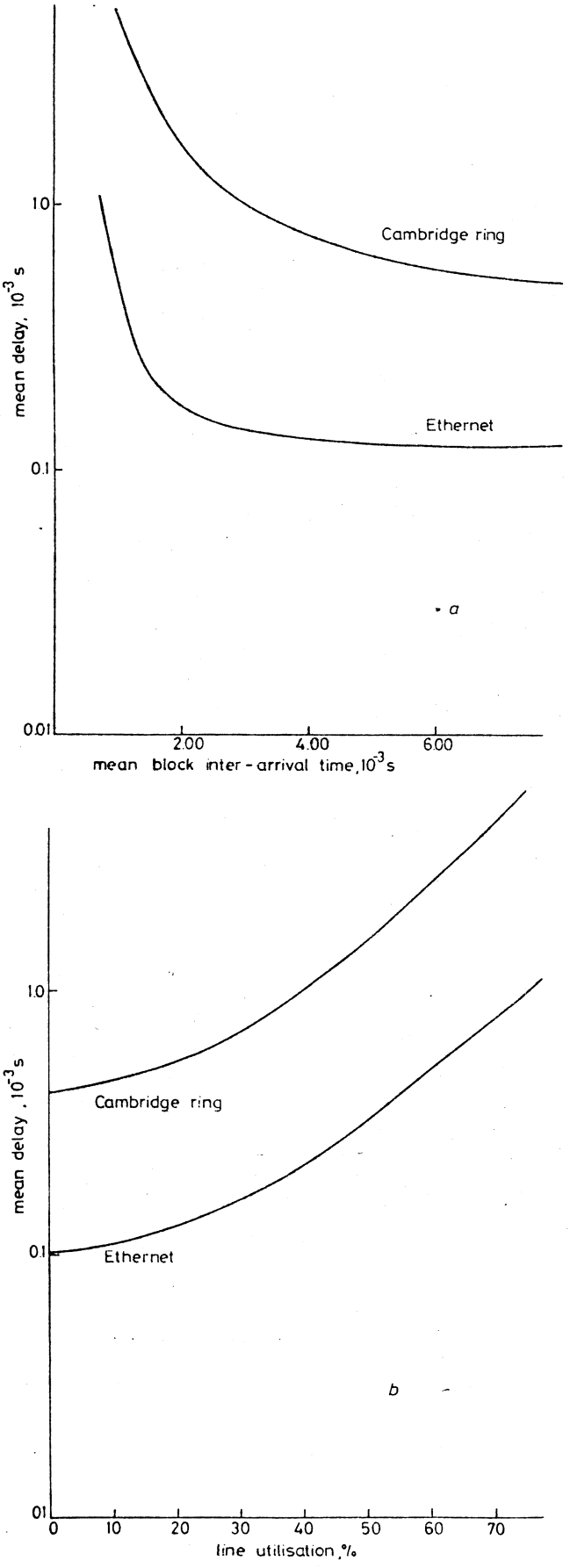


Fig. 11 *Experiment 3*
Packet size = 64 data bytes

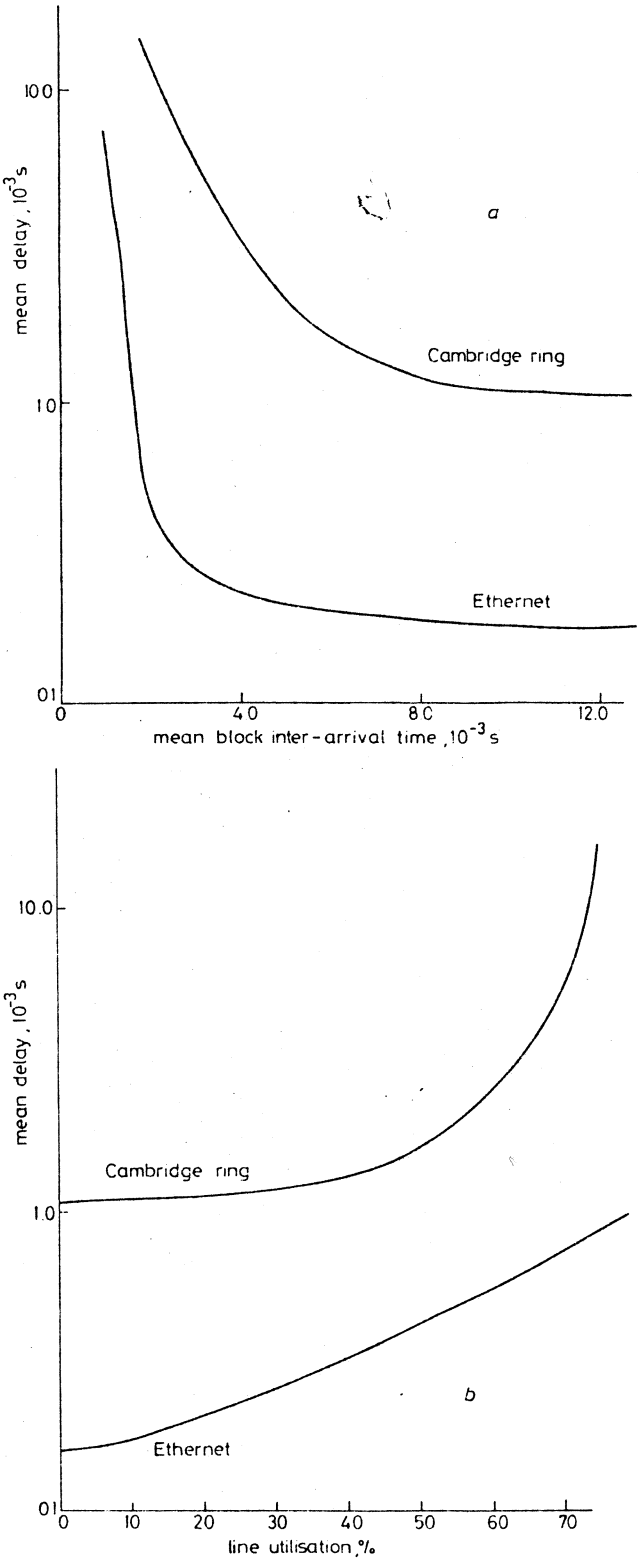


Fig. 12 *Experiment 4*
Packet size = 128 data bytes

values:

average delay for class A = 7.0×10^{-5} s
average delay for class B = 2.2×10^{-4} s

Clearly, both classes of message are transmitted less efficiently in the heterogeneous case. This can be explained by class B messages hogging the ether for long periods, increasing the queueing delays for class A messages. This build-up increases contention for the ether, and therefore as a secondary effect class B messages suffer.

In a second simulation run, class A stations transmitted 16-byte messages with mean wait time of 8.0×10^{-4} s as before, but class B stations transmitted 512-byte messages with mean wait time of 1.5×10^{-2} s. This gives a total offered load of 1.5 Mbits/s. Again, the simulated time is 0.1 s.

The following results were obtained:

average delay for class A = 1.6×10^{-4} s
average delay for class B = 1.6×10^{-4} s

A similar calculation to that above gives the expected delay for class A messages in a homogeneous environment as 7×10^{-5} s.

Therefore, we again see that the longer messages of class B stations have affected the performance of class A stations; therefore the delay depends on the distribution of message lengths and hence total offered load, and not solely on mean values.

4.2 Other factors

The comparisons have been made for systems running at 10 Mbit/s. However, because of the additional complexity in resolving conflict, an Ethernet running at 10 Mbit/s will be more expensive than the equivalent Cambridge ring. It is estimated that, for comparable levels of technology and cost, a 10 Mbit/s Cambridge ring is equivalent to a 3 Mbit/s Ethernet. It has been shown by Blair and Shepherd [11] that if the 3 Mbit/s Ethernet has the same overheads as the standard Ethernet, then the Cambridge ring gives much better delay figures in that case.

Another important property of a local-area network is its reliability. In this respect, Ethernet has several advantages:

(i) The ether is a passive medium as opposed to the active slot structure of the Cambridge ring. The latter relies heavily on the monitor station to create the slot structure and to clear corrupted slots.

(ii) The ring topology of the Cambridge ring means that a line breakage disables the entire network, whereas only a branch of the network would be affected. The Cambridge ring design has included several features to combat this inherent reliability problem [12].

To reduce costs and increase application possibilities, it is essential that LSI-chip based local-area networks can be built. The simple architecture of the Cambridge ring lends itself more readily to this type of development, as has been demonstrated by the implementation at Cambridge University.

One problem with the Cambridge ring is the high overheads of the minipacket protocol: only 16 bits out of 38 carry data. The effect on delay has already been seen. However, it is possible to alter the number of data bytes per minipacket, and investigation has shown that this can be beneficial [13].

Finally, the Cambridge ring, because it can guarantee that a station will get an empty slot within a known maximum time, lends itself more readily to real-time applications, e.g. voice. This has been demonstrated by the implementation of a telephone system on the ring at Cambridge University and is described in the paper by Leslie [14].

4.3 Conclusions

It is clear, from the experimental work carried out, that except for very short message lengths, there is a significant order of magnitude between the performance of the Ethernet system and the Cambridge ring. The difference becomes greater for longer message lengths. The behaviour of the Cambridge ring system under increasing workloads gave a characteristic shape of graph for all message lengths, showing a steady degradation of service. The effect of longer messages, because they are transmitted in uniform-sized packets, did not alter the behaviour but served to increase workload. This had the effect of shifting graphs of mean wait time against delay upwards and to the right as message lengths increased.

The behaviour of the Ethernet under changing workload, however, is dependent on message length. With longer messages, delay times are less sensitive to changing load.

It has also been demonstrated that, in a heterogeneous environment, longer messages can hog the system and cause an overall degradation in performance.

5 References

1 CLARK, D.D., POGRAN, K.J., and REED D.P.: 'An introduction to local area networks', *Proc. IEEE*, 1978, **66**, pp. 1497–1517
2 FREEMAN, H.A., and THURBER, H.J.: 'Updated bibliography on local computer networks', *Comput. Commun. Rev.*, 1980, **10**, (3), pp. 10–18
3 KLEINROCK, L., and TOBAGI, F.A.: 'Packet switching in radio channels: Pt. I', *IEE Trans.*, 1975, **COM-23**, pp. 1400–1416
4 TOBAGI, F.A., and KLEINROCK, L.: 'Packet switching in radio channels: Pts. 2–4', *ibid.*, 1975, **COM-23**, pp. 1417–1433; 1976, **COM-24**, pp. 832–45; 1977, **COM-25**, pp. 1103–1119
5 HOPPER, A.: 'Local area computer communication network'. Ph.D. thesis, Computer Laboratory, University of Cambridge, April 1978
6 ALMES, G.T., and LAZOWSKA, E.D.: 'The behaviour of Ethernet-like computer communication networks', Proceedings of the Seventh Symposium on Operating System Principles, Dec. 1979, pp. 66–81
7 SCHOCH, J.T., and HUPP, J.A.: 'Measured performance of an Ethernet local area network', *Commun. ACM*, 1980, **23**, pp. 711–721
8 WILKES, M.V., and WHEELER, D.J.: 'The Cambridge digital communication ring'. Local area communications network symposium, Mitre Corp. and National Bureau of Standards, Boston, May 1979
9 METCALFE, R.M., and BOGGS, D.R.: 'Ethernet: distributed packet switching for local computer networks', *Commun. ACM*, 1976, **19**, pp. 395–404
10 'The Ethernet, a local area network, data link layer and physical layer specifications', (Digital, Intel and Xerox, 1980)
11 BLAIR, G.S., and SHEPHERD, W.D.: 'A performance comparison of Ethernet and the Cambridge digital communication ring', Internal Report, Dept. of Computer Science, University of Strathclyde
12 HOPPER, A., and WHEELER, D.J.: 'Maintenance of ring communication systems', *IEEE Trans*, 1979, **COM–27**, pp. 760–761
13 BLAIR, G.S.: 'A performance study of the Cambridge ring', to be published in *Comput. Networks*
14 LESLIE, I.M., BANERJEE, R. and LOVE, S.J.: 'Organisation of voice communication on the Cambridge ring' (Local Networks and Distributed Office Systems; Online Publications Ltd, 1981)

Doug. Shepherd received the B.Sc. degree in mathematics from London University in 1965. From 1965 to 1967 he was employed as a technologist at the BP Research Centre, where he was engaged in computer simulation of oil plants. From 1967 to 1971 he was lecturer, then senior lecturer, in computer science at Portsmouth Polytechnic. Since 1971 he has been a lecturer in the Department of Computer Science, University of Strathclyde. He spent a sabbatical at the University of Cambridge from 1980 to 1981, working on the Cambridge ring. His main interests are local area networks and distributed operating systems.

David Hutchison received the B.Sc. degree in physics from Edinburgh University in 1971 and the M.Tech in computing from Brunel in 1976. From 1971 to 1976 he was employed as a senior analyst and programmer with Plessey Radar and Plessey Telecommunication Research, where he worked on the development of operating system for high reliability and multiprocessor applications using the systems 250 capability machine. Since 1976, he has been a lecturer in the Department of Computer Science at Strathclyde University. His main interests are local area networks and VLSI design. He has published a book on logic design.

Gordon Blair received the B.Sc. degree in computer science from the University of Strathclyde in 1981. Since then he has been a postgraduate student in the department of Computer Science, working on his Ph.D. He has published a number of papers on the performance of local area networks. His main interest is in the design of network operating systems which to a large extent have decentralised control.

Contents of *Software & Microsystems*

The contents are given below of the June 1982 issue of *Software & Microsystems.* *

**Software & Microsystems*, published bimonthly by the IEE. *Annual subscription prices*: £27.00 UK, £29.50 overseas, $65.00 USA. IEE members may take out a personal subscription for £10, provided that payment accompanies each order and IEE membership number is clearly stated. All subscription inquiries and orders should be sent to: IEE Publications Sales Dept., PO Box 26, Hitchin, Herts. SG5 1SA, England.

ARJ

Ringing the changes at college

Peter Collinson describes how Unix has grown around a network of Cambridge Rings at Kent.

The Computing Laboratory at the University of Kent has been using Unix since 1978. In 1976, a decision was taken to support a part of the main computer service on a Unix system running on a Vax-11/780. Unix is now run on a number of processors in the university, and is the major part of the computing facilities. Central to the growth of Unix had been the development of a local area network based on the Cambridge ring. The network was the first ring to be installed outside Cambridge University and was also the first to be used in a non-research environment. This article gives an idea of the current state of the network and Unix developments at Kent.

The computing laboratory has two responsibilities: first, the provision of general computing power to the University for teaching and research. Secondly, the laboratory is an academic department, supplying the main teaching input for two undergraduate degree courses and also teaching components of several other courses. The laboratory also undertakes research and has a number of postgraduate students doing work in various areas of computer science.

To supply these needs, the laboratory has three mainframe host processors connected together by a local area network formed from Cambridge rings. One of these processors is an ICL 2960 and the other two are Vax machines running Unix (Version 4.1 BSD from the University of California, Berkeley). The network joins the laboratory to other parts of the campus and there are currently two other Vax processors which run Unix. Two further Vax processors will be installed later this year, one to run Unix and the other VMS.

Current Unix hardware on the Kent Campus

The main Unix system is a Vax-11/780 with 4 Mbytes of memory, a floating point accelerator, one RM80 disc and three RM03 discs. Yes, we are woefully short of disc space. The system also has a TU77 tape drive. This machine supports a maximum of 50 simultaneous users with a very mixed work load; this includes CPU intensive simulations for physicists and chemists, document preparation and production for a wide number of users, project work for 2nd and 3rd year undergraduates, cross assembly of programs for various micro processors—the list could go on and on. The system has a user population of about 700.

Most of the teaching done by the University is supported by a Vax-11/750. This system has 1.5 Mbytes of memory and two RM03 discs. The processor supports about 30 users and has a user population of 650. It is used mostly for teaching Pascal but several other systems and packages are used to service other teaching needs.

The electronics department has a Vax-11/750. The processor has 2 Mbytes of memory, an RM03 and a recently acquired Fujitsu Eagle Winchester drive. This system has a much smaller user population (about 100) and is mostly used for teaching.

The registry owns a Vax-11/750 which is being used to do the payroll and other types of processing relating to administrative functions within the university. This machine has 1 Mybte of memory, two RM03 discs and a TS-11 tape drive.

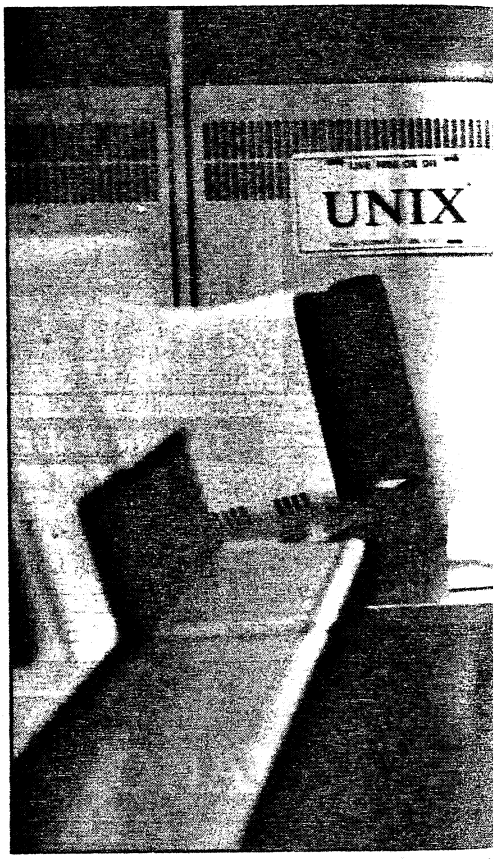
More machines

Finally, two further machines will be installed later this year, a Vax-11/750 will go into the physics laboratory to run Unix and another Vax-11/750 will be installed in the personal social services reseach unit (PSSRU) and will run VMS.

None of the machines functions in isolation, they all use the Cambridge ring for transfer of files. More importantly, most users access the machines using terminals connected via a processor to the ring. The network is so central to the running of the service that when it goes down, which thankfully is seldom, the entire service is knocked out.

The Kent Network—UKCnet

UKCnet consists of four interconnected Cambridge rings. The primary ring is situated in the computing laboratory building, it is connected via a Z80 bridge to the Campus ring. This ring runs through devious routes all over the campus: in all, it is 2.5 kilometres long. The campus ring is seen as a distribution network and in the long term will have several other nets hanging off it. One of these already exists in the registry, which has its own ring and bridge for security purposes. The final ring is used for hardware development and the testing of ring components. The laboratory is also considering the purchase of an Ethernet which will eventually be bridged into the current network.



Peter Collinson at his terminal. The University is a bi

The Vaxs are attached to the ring either by a programmed interrupt interface giving an interrupt every ring mini-packet (16 bits) or by a KMC-11 processor which has been programmed to give an interrupt for blocks of several mini-packets. The ICL 2960 is attached to the ring via a PDP-11 front end processor, the Vaxs use the same protocols and have the same access to the network as the 2960.

Special servers

The ring also has many special purpose processors (or servers) which do particular jobs. Terminals are connected to a set of servers called terminal control processors (TCP). There are currently two types of processor in use for this work. PDP-11s and Z80s. The original TCP software was based on the PDP-11 but

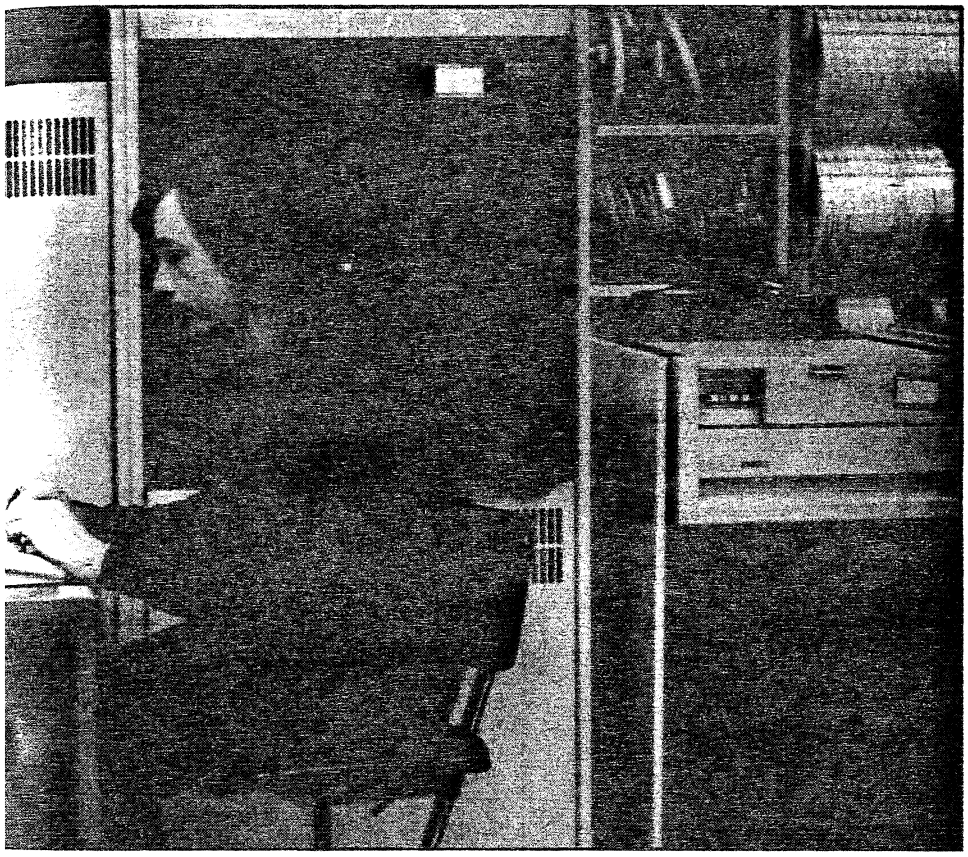
this is being phased out on cost grounds in preference for a Z80 based system. A typical Z80 TCP has a maximum of 16 terminals attached to it and also optionally supports a printer.

For a single terminal, the job of the TCP is to collect a line of data from the user, deal with line reconstruction (deleting characters, words, lines etc.) and then send the clean line up to the selected host. The host has control over the behaviour of the terminal; for instance, the protocol allows the echoing of characters to be turned off for the input of pass words. The protocol also allows the program to select single character input; so, for example, visual editors can be supported.

So, a TCP does most of the work which in normally performed by the lower levels of the terminal handling software in a host.

The effects of this approach are:

- The user can select one of several hosts at login time and then converse with the host as if they were on a directly connected terminal.
- Adding a new host into the system is simply a matter of adding a name into the centralised ring name server.
- We can buy processors with just disc drives and have them working on the network with no recabling of terminals. For instance, the computing laboratory Vax-11/750 was supporting a user service within 24 hours of its release by Dec engineers.
- The host is relieved of the burden of processing characters from terminal lines a byte at a time, and can deal with a single line of input by processing a single interrupt. This only applies to the systems interfaced by the KMC-11. However, the other systems perform less character processing than is normal for directly connected terminals because the TCP processor does all the work of echoing and liner reconstruction.



ax user, with four in place and two on order

TCPs also support printers, and as a general principle, we try to put as many peripherals as possible on servers on the ring. Then, rather than being owned exclusively by one host, the peripherals become sharable resources which may be used by any host on the ring. This approach is used for all printers and also applies to specialised devices such as the Calcomp plotter and the Canon laser printer. To make things easier for software in the Unix hosts, the software is generally fooled into believing that peripheral devices are attached to the host. This means that standard software can be used to drive these devices.

Installation

The installation of a network is not just the joining of processors by bits of wire and a few chips. It is after that has been done that the fun begins, the protocols which are used around the network cost more in man-effort than the actual physical connection.

Protocols

Most connections are done using the standard virtual circuit protocol for Cambridge rings embodied in the Cambridge Ring 82 Protocol Specifications. This provides for a transport service on the ring which is called Transport Service Byte Stream Protocol. (TSBSP). TSBSP uses Basic Block Protocol (BBP) to send blocks of several mini-packets round the ring. BBP can also be used to provide a datagram service. The KMC11 processors on two computing laboratory Vaxs talk BBP to the outside world and use direct memory access to communicate basic blocks into their hosts.

Terminal handling

Terminal handling is done using ITP, which was designed by Edinburgh regional computer centre to support the Emas operating system which is run on the ICL 2960. The Laboratory is planning a switch to the more standard TS29 terminal handling protocol and has a prototype TCP which is designed to use this. The change to the new protocol is expected to happen fairly soon.

The software in the Unix systems consists of three parts: (1) a ring interface driver which communicates using basic blocks to (2) a centralised kernel resident protocol handler, which in turn communicates with (3) a number of different kernel interfaces which are available to user programs.

Perhaps the most important user interface provides terminal access. This mimics the standard Unix terminal interface, so the user software designed to drive terminals is unaware of the network.

Stream device

Another interface provides a standard Unix stream device, i.e. the name of the device is part of the file system, it can be opened and a stream of bytes read or written to it. This interface tends to be used by spoolers and other programs which are normal Unix devices and means that the programs can be naive about the network.

There is an interface which provides access to BBP, this is often used to allow a program to interrogate a server. For instance, one server tunes into the Rugby atomic clock and provides a really accurate time. Another program is used to poll all the machines on the ring and outputs to a TV screen so the stage of the network can be seen at a glance.

Finally, there is a TSBSP interface for

▷

really knowledgeable programs which require access to the protocol primitives —for instance, this interface is used by the programs supporting File Transfer Protocol (FTP).

All of this is achieved by the installation of a protocol handler into the kernel of the Unix operating system. The writing of such a handler requires a good knowledge of the internals of the Unix kernel and is a non-trivial exercise. The author has had a great deal of Unix experience and wrote the initial version in about 2 months. A recent total re-write took somewhat longer because the machines were being used as a service environment and stand-alone access was restricted.

Local communication between machines and servers is the main reason for the installation of the Kent Network. However, communication with the rest of the world must not be neglected. A proportion of the computing needs of the University is met by a direct connection to both the University of London computer centre and an ICL 2988 at Oxford University; this link is performed by a server on the ring.

List of Devices on UKCNET at the University of Kent

Service ring	
Hosts	
ICL 2960	connected by PDP11/34
VAX11/780	
VAX11/750	
PSS Gateway	PDP11/34
ULCC/SERCNET Gateway	PDP11/40
ULCC/Oxford link	PDP11/10 + printers – Printer server
Terminal Booking server	PDP11/24
Other research processors	
Pascal Micro Engine	Compiler Server project
Sage microcomputer	connected by a Z80 – Compiler Server project
Hex 29 processor	
Vector Graphics	
Testing machine	PDP11/34
Testing machine	Z80
Servers	
TCPA	PDP11/10 – 32 terminals + plotter + paper tape
TCPC	PDP11/34 – 32 terminals + 2 printers
TCPD	Z80 – 16 terminals
TCPE	Z80 – 16 terminals supporting micros + printer
TCPF	Z80 – 16 terminals + printer
TCPG	Z80 – 16 terminals + printer
TCPH	Z80 – 16 terminals
REMPHA	Z80 drives a serial line to a TCP – 8 terminals
Campus ring bridge	Z80
Rugby clock driver	Z80
Teletext decoder	Z80
Canon Laser printer	M68000
Name Server	Pronto Z80
Campus ring	
Hosts	
VAX11/750	Electronics
VAX11/750	Physics
VAX11/750	VMS for the PSSRU – connection late 1983
Servers	
TCPRCA	Z80 – 14 terminals + printer
TCPELA	Z80 – 16 terminals + printer
TCPPHA	Z80 – 16 terminals
TCPELB	Z80 – 8 terminals
TCPCHA	Z80 – 12 terminals + printer
TCPMAA	Z80 – 4 terminals
Bridge to Registry ring	Z80
Bridge to Service ring	Z80
Registry ring	
Hosts	
VAX11/750	Registry VAX
Servers	
TCPRGA	Z80 – 16 terminals + printer
Bridge to Campus ring	Z80
Test ring	
Servers	
Bridge to Service ring	Z80
TCPZ	Z80 – test TCP

Communication with the world

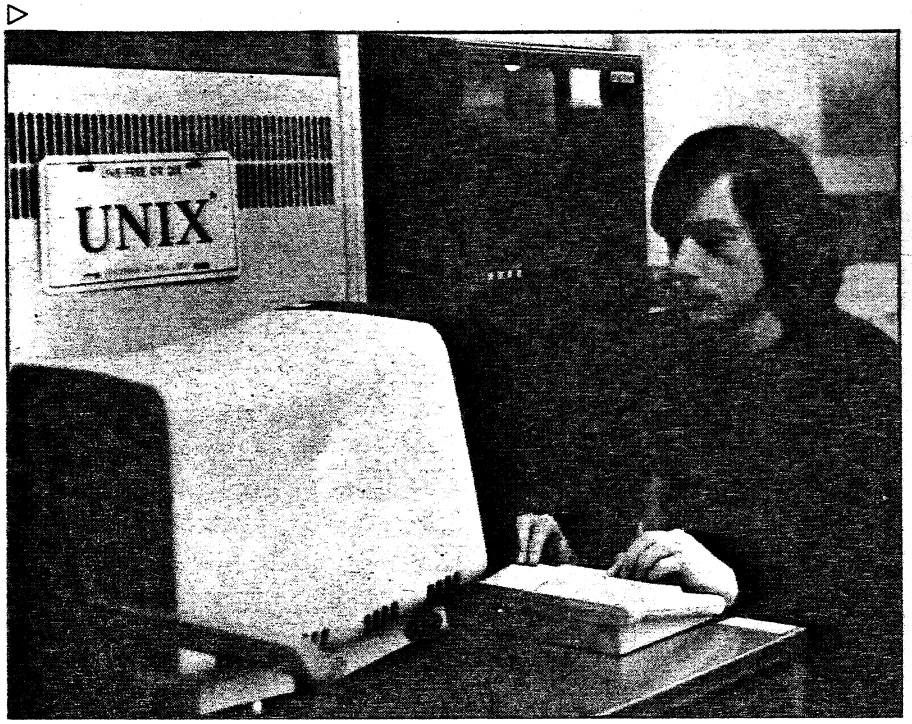
There are also two other areas of communication with the world. First, the laboratory is connected to the British Telecom PSS network by a PDP-11 on the ring. Users log into the gateway using ITP and the gateway converts this to X.25 in order to talk to several sites in the UK and the USA. In addition, FTP is increasingly being used to move files about the UK using PSS. Another PDP-11 is used to connect to Sercnet, the Science and Engineering Council's packet switched network.

The second method of connection to the world uses some standard Unix software called unix-to-unix copy (uucp). The connection is based on telephone calls to the mathematics centre in Amsterdam who in turn call the US Usenet (the US Unix user group is called Usenix hence Usenet), which is an ad-hoc network of over 1000 Unix sites worldwide. The network carries news, bug fixes, information requests and various discussions. The ability to have discussions with the enormous Unix user community in a sort of global village is something which it is not clear you need until you get it—but then it seems impossible to do without it. Usenet has spread to Europe relatively recently and the Laboratory is very keen to fan it out over PSS and phone lines. This fan-out will be helped by the person taking up the new post of Unix support officer, which has been funded from external sources and is intended to help the growth of Unix in British Universities.

The future

Well, that is where we are; where are we going? In the very near future, the Laboratory is expecting the delivery of a file server from Logica VTS. This consists of an Intel 8086 processor and 96 Mbytes of disc storage. The processor will be connected to the network and will allow the installation of disc-less servers for a number of applications. Users will be able to place files on the file server and access them easily from all hosts.

The laboratory is also interested in the



Collinson can talk to some 1000 Unix sites world-wide

Joint Network Team's Mace project. The Mace sits between a host and the ring containing all the code required to perform ring transport service protocols. It consists of a Motorola 6809 processor with DMA interfaces to the ring and also to the host machine. It means that the large code which is required to drive the complicated ring protocols can be placed outboard from the hosts, leaving them free to do 'real' work. An interface for Vax/VMS is being written and will be used to connect the Vax for PSSRU. The laboratory is interested in using the mace for its Unix machines.

New Berkeley system

The author's main interest is the installation of the new Unix system coming from Berkeley (4.2BSD). This system has a set of network support primitives which have grown out of research in distributed computer systems—Inter-Process Communication primitive (IPC). The Laboratory has a pre-release of the new system and will be seeking to use it on the computing laboratory Vaxs in the very near future. The system supports the standard US protocols and we will need to install UK protocols into the kernal. In addition, we can see the need for more sophisticated methods of joining machines on a local area network together. IPC provides a basis for this work and perhaps the Newcastle Connection is one way which this might go. □

Peter Collinson is a lecturer in computer science at the University of Kent.

BARCLAYS MICROPROCESSOR UNIT,
UNIVERSITY OF MANCHESTER.

Presentation of
THE PLEXUS LOCAL AREA NETWORK SYSTEM.

ABSTRACT JUNE, 1984.

The PLEXUS system provides a set of versatile hardware and software modules which can be interconnected to provide a high performance Local Area Network service, suitable for microcomputer systems (including distributed database environments) as well as for systems based on larger machines.

A 'full-configuration' network based on the Plexus system will be divided into *two* logical levels: the *Access* (local) level and the *Hyper* level. Whilst the Access level concentrates on flexibility and convenience, the Hyper level offers high bandwidth, 'trunk' facilities through the use of a star node hierarchy. Efficient broadcast facilities are included.

Rather than restricting the end user to a uniform topology, the Plexus philosophy takes into account tradeoffs between cost, performance and physical convenience, and, therefore, a flexible 'local' link system has been introduced, which allows for various local topologies, including a slot-highway configuration. As well as the basic local link system, there exists a set of *accessory* modules which enhance the performance of the network system, providing facilities such as an efficient, 'built-in' logical channel management system, and a message service which is designed to simplify inter-host, Inter-Process Communication.

s0- sc/class/w/env.

DISTRIBUTION OF AUTOMATED PROCESSES

I. 'GENERAL-PROCESSOR-DEPENDENT' COMMUNICATION.

FULL, CENTRALISED MULTI-PROCESSING/ FILE/ DATABASE SYSTEM; SIMPLE TERMINAL ACCESS.

*LOW-BANDWIDTH, FULL/HALF DUPLEX SERIAL LINKS (V24, 110-9600 BAUD).

*IN-CORE MESSAGE SWITCHING.

-WITH GENERAL TERMINAL ACCESS (GDU, µCOMPUTER, PROCESS CONTROL, ROBOT, ETC.), THEREFORE, POSSIBLE LOCAL PROCESS ACTIVITY.

*HIGH BANDWIDTH (>9600 BAUD) POINT-TO-POINT SERIAL LINKS

*POSSIBLE INTER-PROCESS COOPERATION BETWEEN TERMINAL PROCESSES AND CPU.

-WITH FRONT-END PROCESSING OR 'INTELLIGENT' TERMINAL PROCESSORS.

*HIGH BANDWIDTH SERIAL OR PARALLEL P-T-P LINKS

*INTER-PROCESS COOPERATION OVER LINKS

*IN-CORE MESSAGE SWITCHING

TWO-LEVEL PROCESSOR HIERARCHY- CENTRALISED FILE/DATABASE SYSTEM WITH MANY SATELITE 'DEPARTMENTAL' OR LOCAL PROCESSORS (MINICOMPUTERS)

*HIGH BANDWIDTH SERIAL P-T-P LINKS IN HIERARCHY

*VARIOUS HIGH/LOW BANDWIDTH SERIAL LINKS TO TERMINALS

*IPC ACTIVITY COOPERATION OVER LINKS

*USUALLY IN-CORE MESSAGE SWITCHING

sl sc/class/w/env

II. 'GENERAL-PROCESSOR-INDEPENDENT' COMMUNICATION

DISTRIBUTED MULTIPROCESSOR/FILE ENVIRONMENT - e.g. SINGLE-LEVEL RESOURCE SHARING SYSTEM (DEPARTMENTAL MINIS, FILE SERVERS, NUMBER-CRUNCHERS, PRINTERS, ETC.

*HIGH BANDWIDTH SIMPLE/HIGHWAY ROUTED LINKS (INHERENT NETWORK MESSAGE SWITCHING/ROUTEING).

*IPC ACTIVITY OVER NETWORK.

DISTRIBUTED SYSTEMS OF SINGLE-USER PROCESSORS, SINGLE-LEVEL PROCESSOR ENVIRONMENT, POSSIBLY INCLUDING RESOURCE SHARING AS ABOVE, OR FULLY DISTRIBUTED FILE/DATABASE; TASKING SYSTEM; COMMON PROCESSING INTERESTS.

*HIGH BANDWIDTH SIMPLE/HIGHWAY ROUTED LINKS (HIGH LEVELS OF INHERENT NETWORK MESSAGE SWITCHING/ROUTEING).

*IPC ACTIVITY OVER NETWORK

COMBINATIONS OF THE ABOVE CATEGORIES.

s2/prac/req

PRACTICAL REQUIREMENTS

1 PROVISION FOR REASONABLE NETWORK-ADDRESS SPACE (CAPACITY OF THE ORDER OF 64K TERMINAL NODES BY 'NETWORK-INHERENT' ROUTEING)

2 PROVISION OF TOPOLOGICAL FLEXIBILITY (HIGHWAY, SIMPLE MULTI-POINT, POINT-TO-POINT) AT A 'LOCAL' LEVEL

3 ADEQUATE BANDWIDTH AND TRANSMIT LATENCY TIME CHARACTERISTICS, BEARING IN MIND N^{O} OF POTENTIAL SUBSCRIBERS AND TYPES OF MESSAGE TO BE TRANSFERED (RECORD/TRANSACTION LEVEL)

4 EFFICIENT LOGICAL CHANNEL MANAGEMENT TO BE PROVIDED, FACILITATING HIGH LEVELS OF MESSAGE-CONCURRENCY BETWEEN A USER-HOST AND THE REMAINDER OF THE NETWORK PROCESS CONTEXT.

5 EFFICIENT BROADCAST FACILITIES.

6 MAXIMAL UTILISATION DURING >100% OFFERED HIGHWAY LOAD

7 NETWORK EQUIPMENT SHOULD BE LARGELY 'SELF-CONTAINED'; SHOULD NOT IMPOSE SIGNIFICANT OVERHEADS ON HOSTS.

8 INTERFACES TO HOSTS SHOULD BE CLEAN AND MAINTAINABLE BY USERS.

9 HARDWARE AND SOFTWARE IS TO BE MODULAR, SYSTEMATIC, ROBUST AND EASILY UNDERSTOOD BY USERS, AT A MODULE LEVEL.

10 FLEXIBLE LOCAL TOPOLOGY AND LINK CONFIGURATIONS TO SUIT A VARIETY OF PHYSICAL ENVIRONMENTS.

s3/netov/

PLEXUS NETWORK OVERVIEW

1 'FULL CONFIGURATION' PLEXUS NETWORK IS DIVIDED INTO *TWO* LOGICAL LEVELS: THE *ACCESS* AND THE *HYPER* LEVEL.

2 ACCESS LEVEL PROVIDES 'CLEAN' INTERFACE TO USER HOSTS, AND FLEXIBLE INTERCONNECTION TOPOLOGY, WHILST MAINTAINING PERFORMANCE. THEREFORE -

3 - NO UNIFORM TOPOLOGY SPECIFICATION AT THE LOCAL LEVEL.

4 HYPER LEVEL PROVIDES HIGH-BANDWIDTH TRUNKING BETWEEN LOCAL, ACCESS LEVEL DEPARTMENTS. ORGANISED AS A STAR HIERARCHY.

5 ACCESS LEVEL EQUIPMENT INCLUDES 'INTELLIGENCE' OR CONTROL MODULES, AND A LOGICAL CHANNEL MANAGEMENT SYSTEM.

6 ONE AVAILABLE TOPOLOGY CONFIGURATION AT THE ACCESS LEVEL IS A SLOTTED HIGHWAY SYSTEM.

7 NORMALLY, INTERCONNECTION SYSTEM USES ELECTRICAL LINKS AT THE ACCESS LEVEL, AND OPTICAL LINKS ARE PROJECTED FOR THE HYPER LEVEL.

s4/archov/.

LOCAL (ACCESS-) LEVEL ARCHITECTURAL OVERVIEW.

BASED UPON A VERSATILE, 'INTELLIGENT' LINK MODULE, PLUS A SET OF ACCESSORY MODULES.

LINK MODULE (VSL10H) PROVIDES A 'CLEAN', STREAM-BUFFERED INTERFACE BETWEEN A USER-HOST AND A 10MBIT/SEC FULL DUPLEX SERIAL LINK SYSTEM (ELECTRICAL OR OPTICAL).

LINK MODULE IS RECONFIGURABLE, FACILITATING:

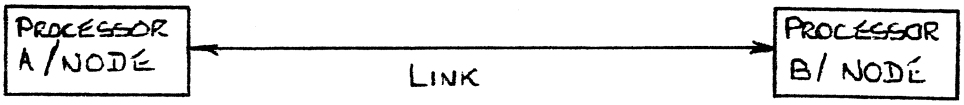
SIMPLE, POINT-TO-POINT;
SIMPLE, WITH MULTI-POINT PROTOCOL;
HIGHWAY (OBVIOUSLY MULTI-POINT).

IN MULTI-POINT CONFIGURATIONS, THE LINK-LEVEL PROTOCOL IS BASED UPON A FIXED LENGTH SLOT OR FRAME, OF LENGTH 160 BITS, 55% UTILISATION.

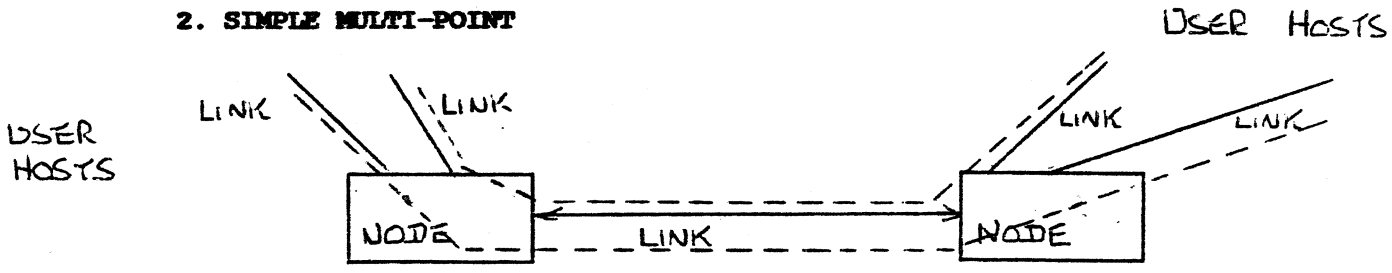
CONFIGURATION DOES NOT INVOLVE ANY HARDWARE CHANGES; OCCURS AUTOMATICALLY, OR BY THE OPERATION OF SOFT SWITCHES.

BASIC CONFIGURATIONS:

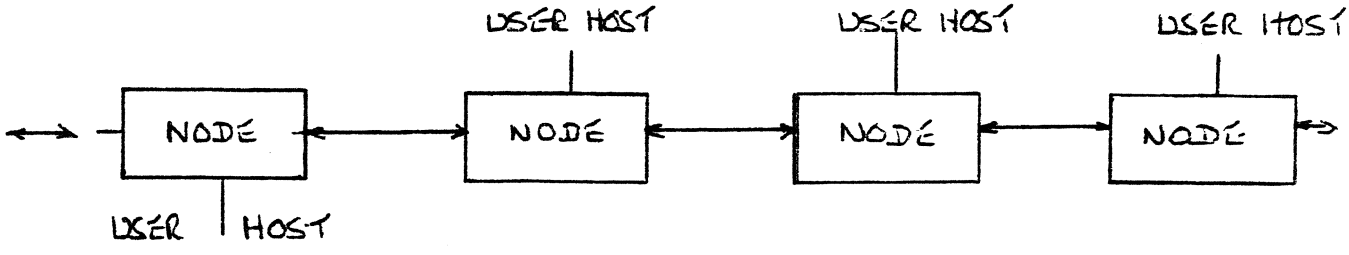
1. SIMPLE POINT-TO POINT



2. SIMPLE MULTI-POINT



3. HIGHWAY



s4a/archov

VSL10H LINK MODULE CAN OPERATE IN CONJUNCTION WITH A USER-HOST, WITHOUT THE AID OF FURTHER MAJOR EQUIPMENT. HOWEVER, TASKS SUCH AS LOGICAL CHANNEL MANAGEMENT/MULTIPLE CHANNEL DMA, BUILT IN IPC MESSAGE SERVICE ARE PROVIDED BY 'ACCESSORY' MODULES.

THE ACCESS-LEVEL ACCESSORIES CONSIST OF:

1. ACCESS CONTROL PROCESSOR - HOSTS AN IPC MESSAGE SERVICE WHICH IS OFFERED UP TO THE USER HOST, OR A GROUP OF USER HOSTS, DEPENDING ON CONFIGURATION. HOSTS, ALSO, A LOGICAL CHANNEL MANAGEMENT SYSTEM WHICH OPERATES IN CONJUNCTION WITH:

2. MULTIPLE CHANNEL BUFFER UNIT - ESSENTIALLY A HIGH-SPEED, 1024 CHANNEL DMA BUFFER UNIT. COLLATES FRAMES RECEIVED FROM THE NETWORK LINK, ACCORDING TO THE INTER-PROCESS MESSAGE OF WHICH THEY ARE A PART. BUFFERING OPERATION IS BY TABLE-DRIVEN DMA.

ALL MODULES ARE INTERCONNECTED BY A LOCAL BUS SYSTEM - THE *NETWORK EQUIPMENT HIGHWAY*. THIS IS A 16 OR 32 BIT (DATA), SYNCHRONOUS HIGHWAY, HAVING SIMILAR COMPLEXITY TO THE IEEE 796 SYSTEM, BUT WITH A SLIGHTLY SPECIALISED TRANSFER PROTOCOL.

BECAUSE COMPLEX NODE CONFIGURATIONS CAN EXIST (E.G. SEVERAL LINKS CONVERGING ON A SINGLE NODE (STAR), POSSIBLY WITH ASSOCIATED ACCESSORY EQUIPMENT), SEPERATE *NEH ARBITRATION MODULES* ARE PROVIDED.

FINALLY, THEIR REMAINS THE PROBLEM OF INTERFACING A NODE TO THE OUTSIDE WORLD. *NEH INTERFACE MODULES* PROVIDE INTERFACES BETWEEN THE NETWORK EQUIPMENT HIGHWAY AND STANDARD MICROCOMPUTER AND INSTRUMENTATION HIGHWAYS (IEEE 488, 796, Z80, ETC.).

s4b/hyp/lev/star

TRUNK (HYPER-) LEVEL OVERVIEW.

BESIDES THE 'INTELLIGENT' VSL10H LINK MODULE, OTHER LINK SYSTEMS ARE PROJECTED -

VSL20S, VSL30S.

THESE ARE TO HAVE OPERATING RATES OF 50+50 & 140+140 MBIT/SEC (GROSS) RESPECTIVELY.

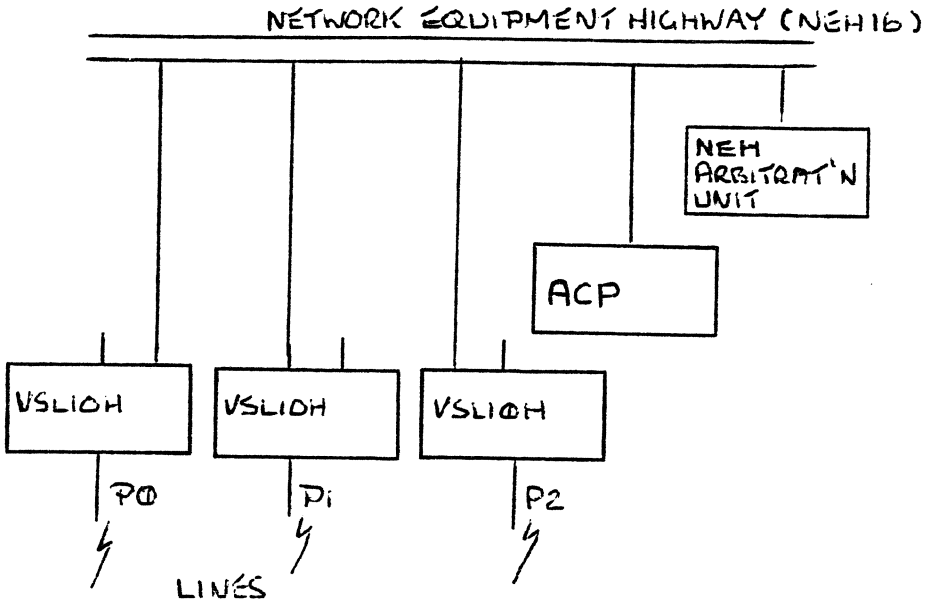
HOWEVER, "20S AND "30S SYSTEMS WILL NOT HAVE THE SAME BUFFERING AND INTELLIGENT FACILITIES AS THE "10H. FURTHERMORE, THESE WILL ONLY OPERATE IN A SIMPLE (NON-HIGHWAY) CONFIGURATION.

IF A COMBINATION OF THE VARIOUS LINK MODULES (UP TO 16) ARE CONNECTED TO A 32 BIT VERSION OF THE NETWORK EQUIPMENT HIGHWAY (NEH32), A HIGH-THROUGHPUT *STAR* NODE CONFIGURATION IS FORMED - THIS IS THE HEART OF THE *HYPER-LEVEL* SYSTEM.

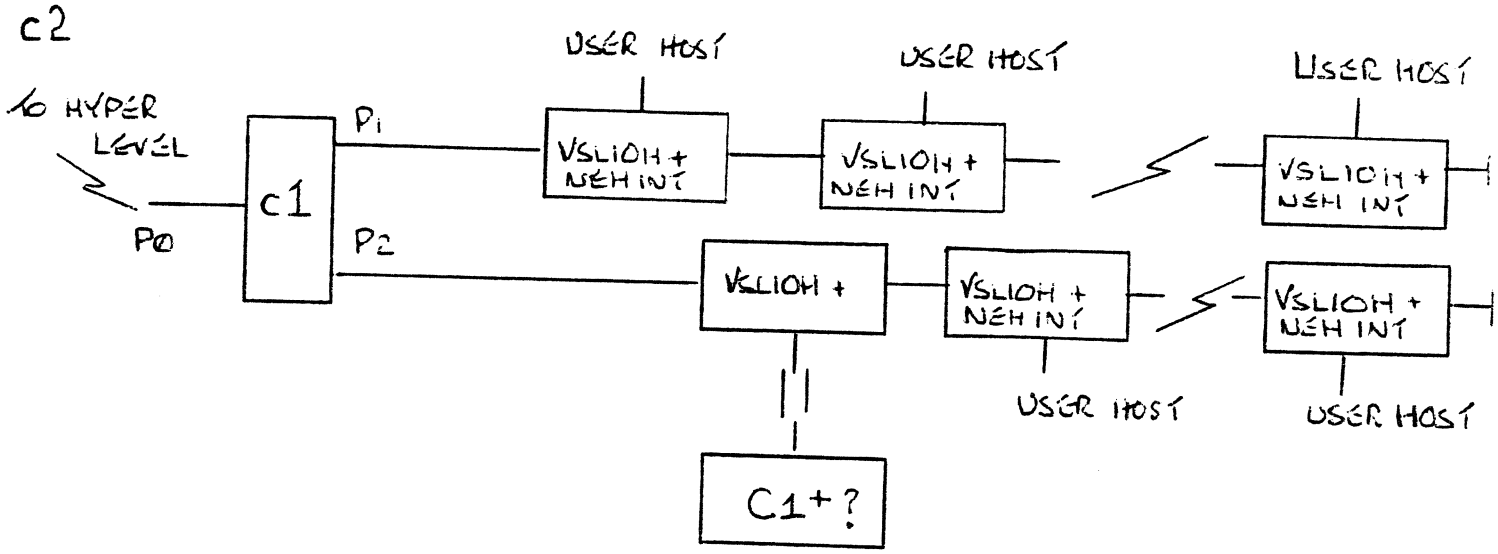
s4c/archov/

ACCESS-LEVEL CONFIGURATION EXAMPLES

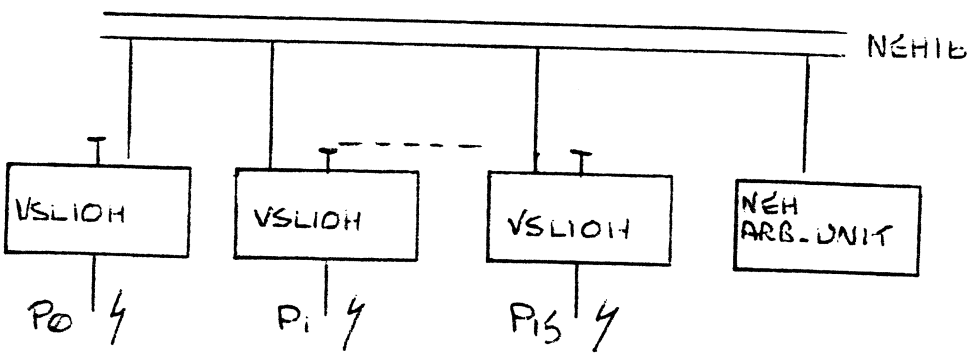
C1. STAR (3-PORT) WITH MCBU AND ACP



C2



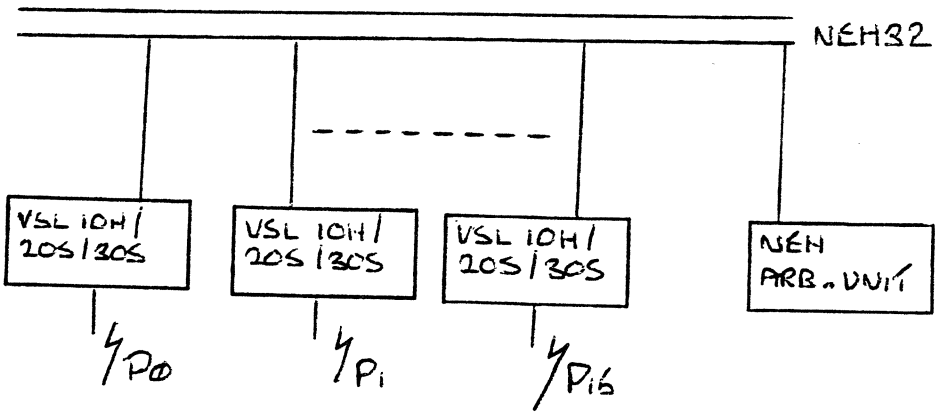
C3 SIMPLE STAR (16 PORT)



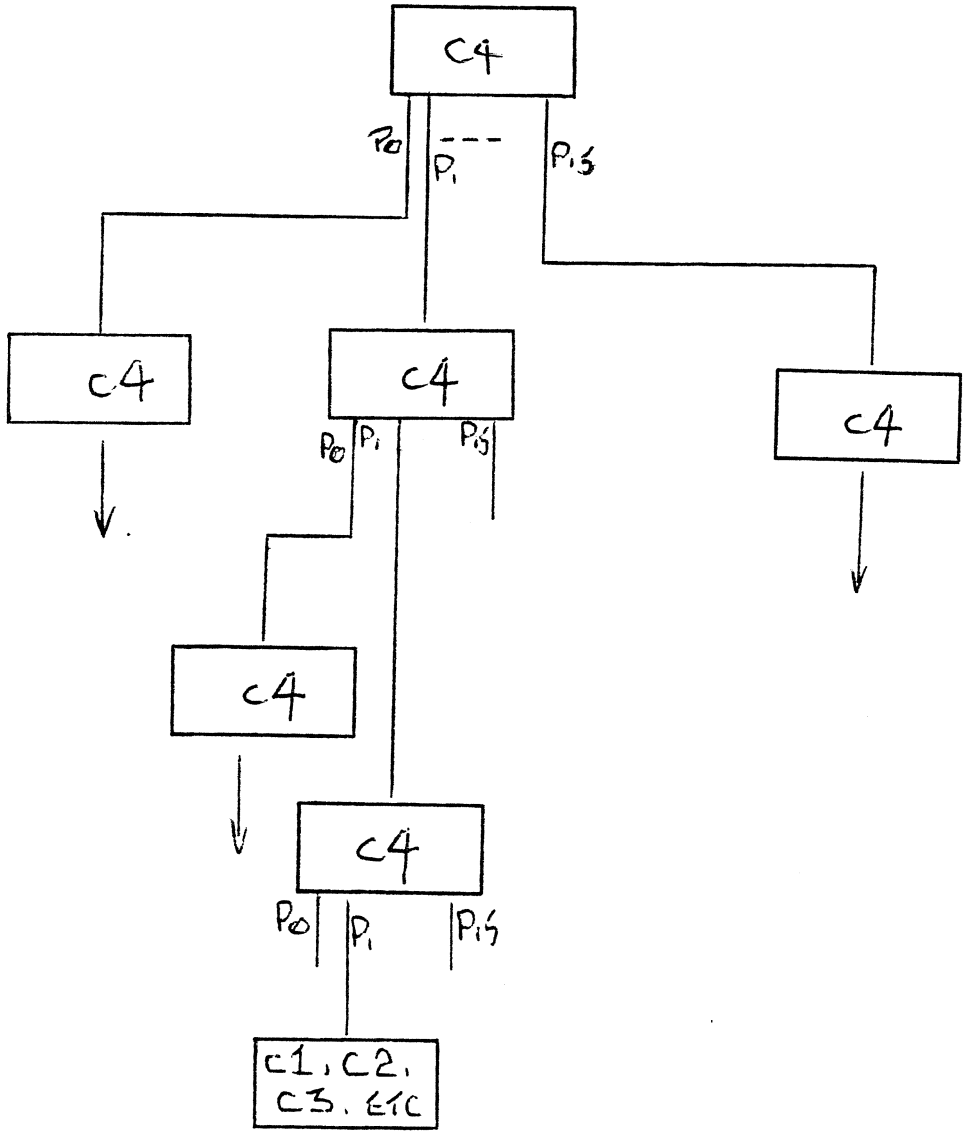
s4d/archov/

HYPER-LEVEL INTERCONNECTION

c4



c5



s5/proov/

PROTOCOL OVERVIEW
(NO REFERENCE TO OSI 7-LAYER REF. MODEL!)

LINK LEVEL PROTOCOL IS FRAME-BASED.

FRAMES ARE INDIVIDUALLY ROUTED, THEREFORE SOURCE AND DESTINATION ADDRESSES ARE REQUIRED FOR EACH FRAME.

FOR EACH HIGHWAY OR SIMPLE LINK, AN INDEPENDENT SYSTEM OF ERROR CHECK/RECOVERY AND HANDSHAKING EXISTS. EACH FRAME CARRIES A CHECKWORD.

ACP HOSTS A HIGH-LEVEL MESSAGE SERVICE, WHICH CONCEALS LINK PECULIARITIES. THIS PROVIDES INTER-PROCESS COMMUNICATION FACILITIES ACROSS THE NETWORK. (FACILITATES SIMPLER IMPLEMENTATION OF A *NOS* WHILST SPARING THE PROCESSING RESOURCES OF A USER-HOST PROCESSOR).

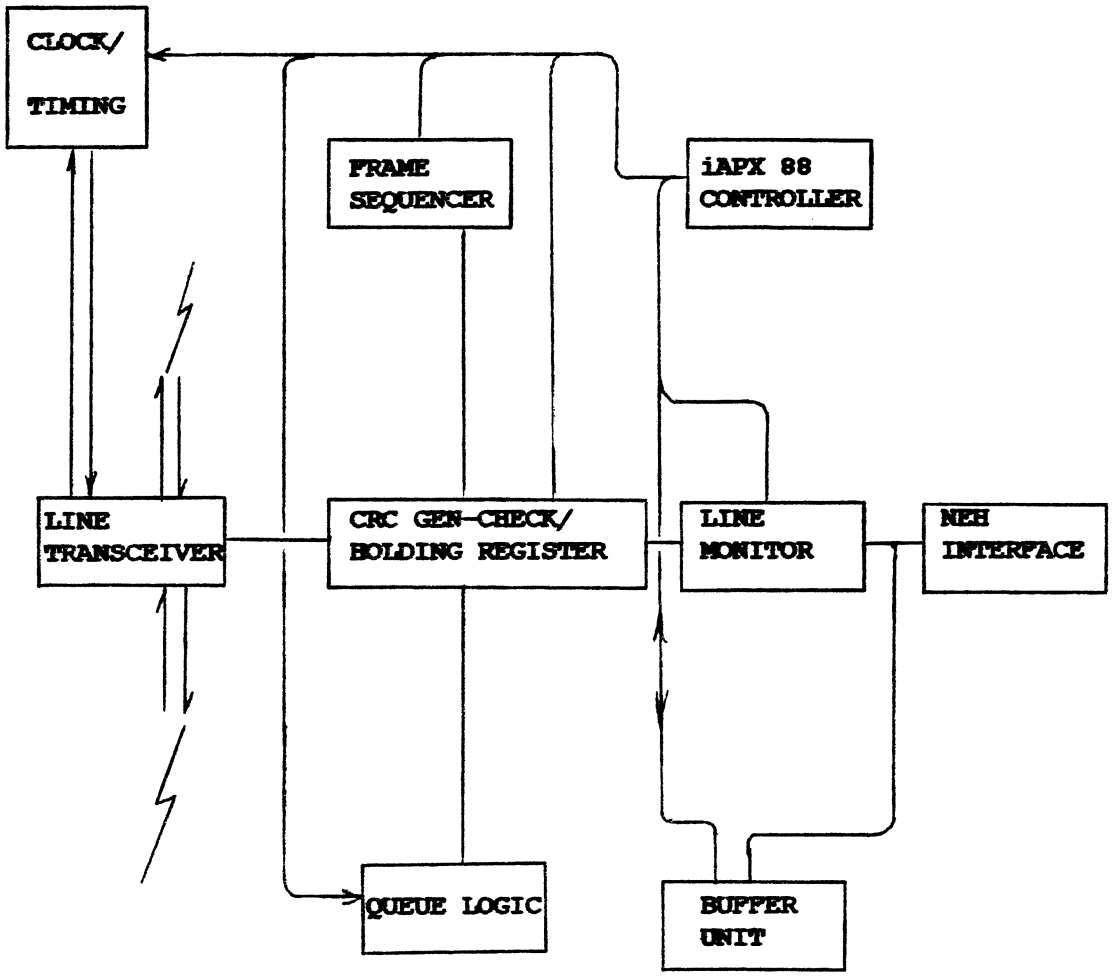
EACH MESSAGE IS ALLOCATED A *LOGICAL CHANNEL* (BY THE *LCMS*), WHICH IS A UNIT OF NETWORK RESOURCE.

A LOGICAL CHANNEL MAY BE CONFIGURED TO CLOSE DOWN IMMEDIATELY AFTER THE MESSAGE TRANSFER IS COMPLETE, OR IT MAY EXIST AS A LONG-TERM ENTITY BETWEEN PROCESSES.

CONVENTIONAL DATAGRAM FACILITIES WILL BE IMPLEMENTED.

s6/vsl10h/sch

VSL10H LINK MODULE - SCHEMATIC



DESCRIPTION

iAPX88 BASED CONTROLLER
HOSTS THE VSL10H LINK MANAGEMENT SYSTEM. THIS INCLUDES AUTOMATIC INITIALISATION, CONFIGURATION, LINK EVENT MONITORING, DIAGNOSTIC MANAGEMENT AND ERROR RECOVERY. NOTE THAT CONFIGURATION IS LARGELY AN INVISIBLE OPERATION IN THAT THE USER IS NOT AWARE THAT IT IS TAKING PLACE. CONSEQUENTLY, IN THE HIGHWAY CONFIGURATION, ALL NODES APPEAR TO SERVE THE SAME FUNCTION REGARDLESS OF THEIR POSITION ON THE HIGHWAY.

s7/vsll0hdes/

LINE TRANSCEIVER -
BASIC OPTICAL/ELECTRICAL CABLE DRIVER + RECEIVER, MANCHESTER ENCODER/DECODER.

CLOCK/TIMING -
GENERATES ALL TIMING SIGNALS, EITHER FROM INTERNAL MASTER CLOCK, OR FROM RECOVERED LINE CLOCK.

SEQUENCER -
ACTIVE ONLY IN THE NODE AT THE 'HEAD' OF A HIGHWAY CONFIGURATION. ORIGINATES THE LINK LINE SEQUENCE (FRAMING, LINK UTILITY FIELD, ETC.). AS WELL AS THE NORMAL 160 BIT, MULTI-POINT FRAME SEQUENCE, OTHER SEQUENCES EXIST, SUCH AS POINT-TO-POINT, DIAGNOSTIC/INITIALISE SEQUENCES, ERROR (AUTO-RETRANSMIT) SEQUENCES.

QUEUE LOGIC -
ACTIVE ONLY IN THE NODE AT THE 'TAIL' OF A HIGHWAY CONFIGURATION. ESSENTIALLY THE 'HEART' OF THE DISTRIBUTED QUEUEING SYSTEM WHICH IS FORMED WHEN A SERIES OF LINKS ARE CONFIGURED AS A HIGHWAY.

LINE MONITOR -
RESPONSIBLE FOR ERROR AND EVENT SIGNALLING BETWEEN THE LINE AND THE CONTROLLER, AND FOR GENERAL, LOW LEVEL LINK CONTROL (QUEUE REQUESTS, INITIALISATION, ETC.).

CRC GEN-CHECK/HOLDING REGISTER -

PROVIDES THE CYCLIC REDUNDENCY CHECK FUNCTION, AND THE REGISTERS INTO WHICH WORDS TO BE ENCODED FOR TRANSMISSION ARE ASSEMBLED, AND INTO WHICH NEWLY RECEIVED/DECODED WORDS ARE RECEIVED.

BUFFER UNIT -

A LARGE SCALE STREAM BUFFER TO WHICH TRANSMIT FRAMES ARE WRITTEN, AND FROM WHICH RECEIVED FRAMES ARE READ. SERVES AS A SMOOTHING BUFFER BETWEEN THE LINK AND A HOST (MINIMUM CONFIGURATION) OR THE MAIN BUFFER SYSTEM. ALSO RETAINS FRAMES AFTER TRANSMISSION, FOR RETRANSMISSION IN THE EVENT OF A LINK ERROR.

NEH INTERFACE -

PROVIDES THE STANDARD INTERFACE TO THE LOCAL NETWORK EQUIPMENT HIGHWAY.

s8/vsl10/prot

VSL10H LINK SYSTEM - LINK FRAME EXAMPLE (MULTI-POINT)

DESTINATION ADDRESS:	**16 BITS;**
NETWORK UTILITY FIELD:	**16 BITS;**
SOURCE ADDRESS:	**16 BITS;**
AUXHILIARY CONTROL:	**8 BITS;**
USER DATA FIELD:	**88 BITS;**
CRC FIELD:	**16 BITS.**
FRAME SIZE:	**160 BITS**

s9/neh/sig

NETWORK EQUIPMENT HIGHWAY - SIGNAL DESCRIPTION

HCK	HIGHWAY CLOCK;
INIT	INITIALISE;
AUXRES	AUXHILIARY RESET;
RD	READ/$\overline{\text{WRITE}}$;
RR	READ RETURN QUALIFIER;
AM[0-1]	ADDRESSING MODE (2);
RT[0-9]	READ TAG (10);
RRT[0-9]	READ RETURN TAG (10);
RQ[0-2]	REQUEST CODE (3);
GR[0-2]	GRANT CODE (3);
EQAD[0-4]	EQUIPMENT ADDRESS (5);
AD[0-19]	ADDRESS (20);
D[0-15/31]	DATA (16/32);

s9a/nar/

NETWORK EQUIPMENT HIGHWAY - SIGNAL DESCRIPTION

HIGHWAY CLOCK;
10 MHZ CLOCK (BALANCED) TO WHICH ALL OTHER SIGNALS MUST BE SYNCHRONISED.

INITIALISE;
INITIALISES ALL ATTATCHED MODULES TO A PREDETERMINED STATE.

AUXHILIARY RESET;
EXTERNAL (MANUAL) RESET.

READ/WRITE;
HIGHWAY ACCESS QUALIFIER.

READ RETURN QUALIFIER;
DATA RETURNED AS A RESULT OF A READ OPERATION.

ADDRESSING MODE (2);
REAL/././ NAME+CURRENT POINTER.

READ TAG (10);
'NAME' OF READ OPERATION.

READ RETURN TAG (10);
'NAME' OF A DATA RETURN OPERATION (RESPONSE TO 'READ TAG'.

REQUEST CODE (3);
REQUESTS ADDRESS(READ)/DATA(RETURN)/BOTH(WRITE)/BOTH(PRIORITY-INTERRUPT).

GRANT CODE (3);
GRANT COUNTERPART TO THE ABOVE.

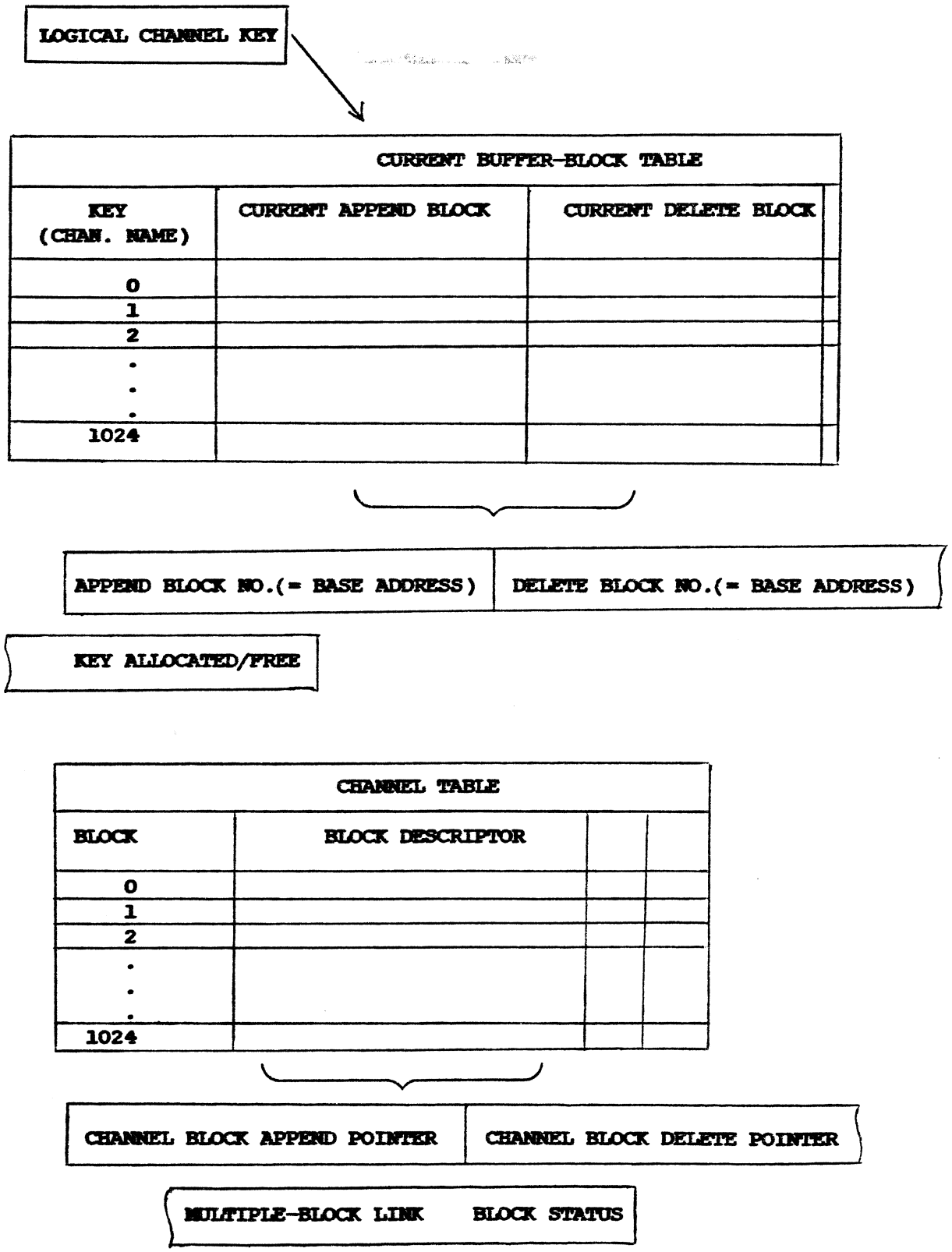
EQUIPMENT ADDRESS (5);
ATTATCHED MODULE IDENTIFIER.

ADDRESS (20);

DATA (16/32);

s10/mcbu/

MULTIPLE CHANNEL BUFFER UNIT - BLOCK SCHEMATIC & CHANNEL TABLE DESCRIPTION.



s11/mcbudes/

LOGICAL CHANNEL KEY - BUFFER ADDRESS TRANSLATION

DESCRIPTION

LOGICAL CHANNEL KEY
(SEE DEFINITIONS) THE 'NAME' OF A LOGICAL CHANNEL, UNIQUELY ASSIGNED (WITHIN THE SCOPE OF A LCMS) TO A TERMINAL OR NETWORK PROCESS. INDEXES THE -

CURRENT BUFFER BLOCK TABLE
WHICH CONTAINS, FOR EVERY ALLOCATED CHANNEL, AN -

APPEND BLOCK NO.
AND A -

DELETE BLOCK NO.

FOR EVERY BUFFER ACCESS, DEPENDING ON WHETHER IT IS AN APPENDIX OR A DELETION, ONE OF THESE ITEMS, BEING THE BASE ADDRESS OF THE CURRENT APPEND/DELETE BLOCK FOR A PARTICULAR CHANNEL, IS PASSED ON TO ACCESS A BLOCK ENTRY IN THE -

CHANNEL TABLE
I.E., FOR EVERY BLOCK OF 256 BYTES IN THE BUFFER SPACE, THERE EXISTS AN ENTRY CONTAINING A

BLOCK DESCRIPTOR
OR, MORE SPECIFICALLY

BLOCK APPEND POINTER
BLOCK DELETE POINTER
MULTIPLE BLOCK LINK
BLOCK STATUS.

THE APPEND AND DELETE POINTERS INDEX THE CURRENT WRITE/READ OFFSETS WITHIN THE BLOCK. IN CASES WHERE A CHANNEL IS ALLOCATED A BUFFER AREA CONSISTING OF SEVERAL BLOCKS, IT MAY NOT BE POSSIBLE TO ALLOCATE THESE BLOCKS AS A CONTIGUOUS AREA OF BUFFER, SO FRAGMENTED BLOCKS ARE LINKED INTO A LOGICAL CHAIN USING THE MULTIPLE BLOCK LINKER, I.E., THE LINKER POINTS TO THE DESCRIPTOR FOR THE BLOCK WHICH IS NEXT IN THE LOGICAL CHAIN. THE CHAIN IS ESTABLISHED BY THE ACP WHEN THE CHANNEL IS ESTABLISHED, AND THE DESCRIPTOR OF THE FIRST APPEND/DELETE BLOCK IN THE CHAIN IS INDEXED BY THE APPROPRIATE 'APPEND/DELETE BLOCK NO.' ENTRY IN THE CURRENT BUFFER BLOCK TABLE. WHEN A READ OR WRITE ACCESS IS MADE TO THE BUFFER, THE DELETE OR APPEND POINTER IS INCREMENTED. IF THE POINTER REACHES ITS MAXIMUM VALUE, THEN, IF ANOTHER BLOCK IS TO FOLLOW WITHIN THE SAME LOGICAL BUFFER AREA, THE CONTENTS OF THE MULTIPLE BLOCK LINKER ARE PLACED IN THE APPROPRIATE HALF OF THE CURRENT BUFFER BLOCK TABLE, HENCE THE NEW BLOCK IS NOW 'CURRENT'.

THE BLOCK STATUS REGISTER RECORDS INFORMATION SUCH AS WHETHER OR NOT THE BLOCK IS CURRENTLY FREE, WHETHER OR NOT THE MULTIPLE BLOCK LINKER IS VALID (USED), ETC.

sl2/psu/

POWER SUPPLY

POINT OF INTEREST - POWER SUPPLY TO ACCESS-LEVEL VSL10H MODULES WHEN OPERATED IN HIGHWAY MODE.

POSSIBILITIES:

1 POWER 'BUS'
  A IN PARALLEL WITH, OR,
  B SUPERIMPOSED UPON LINK LINES;

2 INDEPENDENT PSU IN EACH VSL10 MODULE.

1B ABOVE WAS ADOPTED FOR THE CAMBRIDGE RING SYSTEM - DOUBLE-BALANCED PAIR CABLE USED TO CARRY DATA & POWER SUPPLY.

PROBLEMS (1) -

POWER NECESSARY FOR SEVERAL VSL10H NODES MAY NECESSITATE EXCESSIVELY HIGH LINE VOLTAGES. A PRACTICAL LIMIT WILL BE IMPOSED WHICH MAY BE RESTRICTIVE.

1B IMPOSSIBLE WITH REGARDS TO OPTICAL SYSTEM.

GREATER PHYSICAL ORGANISATION THAN 2.

(2) -

UNLESS SPECIAL PRECAUTIONS ARE TAKEN, REMOVAL OF POWER SUPPLY (MAINS) BY A USER, TO ANY NODE, RENDERS THE HIGHWAY USELESS.

SOLUTION ADOPTED - 2

TEMPORARY BATTERY BACKUP PROVIDED WITHIN EACH VSL10H MODULE, WHICH, IN THE EVENT OF A LOCAL POWER FAILURE, MAINTAINS LINK UNTIL IT IS 'CLOSED', OR UNTIL POWER IS RESTORED. SIMPLE ALARM SYSTEM WILL INDICATE ANY LOSS OF 'PSU' POWER DURING OPERATION (I.E. ACTIVATED AT THE SWITCHOVER FROM PSU TO BATTERY).

PLEXUS LOCAL AREA COMMUNICATION NETWORK:
DEFINITIONS.

ACCESS CONTROL PROCESSOR
Access-level network 'intelligence' accessory. 8086 microprocessor based module which hosts the *Logical Channel Management System*, as well as the *Message Service*. Operates in conjunction with the *Multiple-Channel Buffer Unit*: the pair may exist one per host or one per group of hosts.

ACP
Access Control Processor.

ACCESS LEVEL
The part of the network equipment to which user equipment (terminal processors, etc.) are attatched. Access-Level equipment includes the *Access Control Processor, VSL10H highway system, Multiple-Channel Buffer Unit, Network Equipment Highway System (NEH16) + Arbitration unit*. NOTE that the VSL10H system, along with a version of the Network Equipment Highway (NEH32), are also compatible with the *Hyper-level* system.

A-LEVEL
Access level.

CHANNEL DE-ESTABLISHMENT (TIME)
See *'CHANNEL ESTABLISHMENT TIME.'*

CHANNEL ESTABLISHMENT (TIME)
When a user/terminal process has an imminent message stream to transfer over the network, a message request is submitted to the *Access Control Processor*, which responds by transmitting a request, over the network, to the ACP which is responsible for the destination process (remote ACP). The message request is queued at the remote ACP. When this is serviced, the remote ACP creates an entry in the Multiple Channel Buffer Unit's LCK tables, which allocates an amount of buffer space large enough to receive the whole of the imminent message. Once this operation is complete, a *receive* channel can be considered granted; information to this effect is returned to the sending unit which subsequently releases the message. (NOTE that at the sending unit, the message will have been established within the local MCBU, and a *transmit* channel will have been established in a similar way to that in which the receive channel was created. In fact, the two may be created simultaneously under most circumstances. In the above example, the definition of channel establishment *time* is: the duration from the time at which the message-request is submitted by the user process, to the time at which either the channel is granted by the receive process, or the transmit channel is established by the transmit process, whichever occurs later. Channel *De*-establishment time is that period during which the message is acknowledged, and the logical channels are closed down, again by accessing the channel tables. The above operation is a request to transmit a message to a remote process: similarly, data may be requested from the remote point, in which case the roles of the local and remote buffers are reversed.

FRAME
In a *routed link (multi-point)* context, (one in which many destinations are possible) a frame is a unit or packet of information, to which addressing, control and error checking data are tagged for the purpose of transmission over the network. In the PLEXUS system, the frame length is deliberately short and fixed, as in the case of the Cambridge Ring, Centrenet, etc. This is for performance reasons (Transmit-Latency remains predictable). Obviously, a *message* may consist of many, uncontiguous frames.

If the link is *unrouted* (a private entity between only two logical points), there need not be the same emphasis on latency problems, since the link is not shared. Therefore, the frame, in this context, may exclude addressing, and it can be of variable length, if not continuous. Clearly, the disadvantage in the case of shorter frames is the degradation in useful bandwidth due to address/control overheads.

The format of a normal PLEXUS frame is as follows (in order of transmission):

NETWORK UTILITY FIELD: *16 BITS;*
DESTINATION ADDRESS: *16 BITS;*
AUXHILIARY CONTROL: *8 BITS;*
SOURCE ADDRESS/LCK: *16 BITS;*
USER DATA FIELD: *88 BITS;*
CRC FIELD: *16 BITS;*

FULL DUPLEX, SLOTTED PSEUDO-HIGHWAY
The tedious, but official term for the VSL10H link system when in highway configuration.

GBS
General Broadcast System.

GENERAL BROADCAST SYSTEM
The system of facilities throughout the access- and hyper- levels, which provide for efficient broadcasting at the *frame* level.

GENERAL TERMINAL
In contrast to the *simple teminal*, the term 'General Terminal' is used to describe, for example, a graphic terminal, a microcomputer hosting a *terminal process*, or a physical process-control system. More generally, the term refers to a terminal which may demand high link bandwidth (>>19.2KBaud) and inter-process communication (IPC) activity.

GROSS BANDWIDTH
(Link context) The rate (in bits/sec) at which binary data (not only user-data, but address, control and other overheads) are transfered across a link.

HIGHWAY ARBITRATION UNIT

An *accessory unit* which is used in both the *A-* and the *H-*level contexts for co-ordinating access to the *Network Equipment Highway (NEH16, NEH32)* system. This unit provides a circular-priority highway access mechanism.

HIGHWAY LINK
In the PLEXUS context, synonymous with '*full duplex, slotted pseudo-highway*'.

H-LEVEL
Hyper-Level.

HOST PROCESSOR
Usually a user-processor (micro'), Hosting a *terminal process*.

HYPER LEVEL
The part of a PLEXUS network body which normally exhibits very high bandwidths (50-140 Mbit Gross.), is of a *star hierarchy* topology, and which provides a 'trunk' service interconnecting local, *Access-level* systems (highways, etc). see *Access-level*.

INHERENT LINK ACCESS TIME
The lapsed time incurred by a link or link interface in commencing transmission of a *frame* once it has been established in a local stream buffer. Often caused by link-queueing mechanisms.

INHERENT LINK-DELAY TIME
The delay offered by a frame's journey across a link or a series of links, between injecting a *frame* into one end, and the frame appearing at the other, assuming no other obstacles- e.g. queueing.

INTER-PROCESS COMMUNICATION

In the context of a truely distributed Network Operating System (*NOS*), operating-system/user processes communicate within either a single *host* or between hosts, via a uniform message service. This level of communication is termed a 'uniform IPC mechanism.'

IPC
Inter-Process Communication.

LCK
Logical Channel Key.

LCMS
Logical Channel Management System.

LINK CONFIGURATION
Usually used in reference to the reconfigurable *link systems*. Currently, the only system of this type is the PLEXUS VSL10H. Examples of configurations are: *'Highway' (multi-point), 'Unrouted, simple (point-to-point)', 'Routed, simple' (multi-point).*

LINK MODULE
The actual piece of equipment which provides VSL10/20/30 link access, interfacing between the optical or electrical cable, and the *Network Equipment Highway.*

LINK SYSTEM
This term refers to the practical *link module*/protocol combination used in constructing a link, i.e. VSL10/20/30.

LOGICAL CHANNEL

A temporarily or permanently established unit of network resource, which exists as a one-to-one or many-to-one routeing system, giving the appearance of a dedicated link amongst the hosts concerned.
See 'Logical Channel Establishment,''Logical Channel key'.

LOGICAL CHANNEL KEY
In a *routed link* system, an unscheduled frame (one which is *not* a part of a message on a properly established *Logical Channel*) bears a 16-bit *source* address field. This is used, at a destination, simply to determine the frame source. However, if a frame is a part of an MCBU established message on a *logical channel*, the destination buffering operation must occur in a high-speed, 'DMA' mode, as the frame arrives. Effectively, the frame data will be written to a 'segment' of buffer, the address of which will depend on the frame-source value. Rather than using the full source address to perform the mapping (which would involve too large a decode-set to be practical at the present), a 10-bit *Logical Channel Key* is arbitarily assigned to each logical channel. The 10-bit value reduces the decode-set.

During *Logical Channel Establishment*, the *ACP* responsible for the receiving process transmits the assigned LCK to the ACP responsible for the transmitting process. This key is transmitted in the *source* address field of every frame transmitted over the logical channel.

LOGICAL CHANNEL MANAGEMENT SYSTEM
An *ACP* resident software/firmware system which manages the allocation and establishment/de-establishment of *logical channels*.

MCBU

Multiple Channel Buffer Unit.

MESSAGE
The basic unit of communication between processes, either within a single host, or between hosts (across the network)

MESSAGE CONCURRENCY
Delineates the quantity of *logical channels* converging on a single host or host group.

MESSAGE SERVICE
The 'built-in' *IPC* facilities provided by an *ACP*.

MULTIPLE CHANNEL BUFFER UNIT

An accessory unit which is capable of simultaneous, multi-channel 'DMA' type buffering at high speed. *Frame* data are transmitted from, or received into a 'segment' of the buffer, the address of which is determined from the *Logical Channel Key*.

NEH
Network Equipment Highway.

NEH INTERFACE MODULE.
Any module which interfaces between the *NEH* and a standard microcomputer or instrumentation highway (IEEE 796, 488).

NETWORK ACCESSORY MODULE
Any *NEH* compatible equipment which enhances an *Access-level* link above the minimum configuration (VSL10 plus host).

NETWORK EQUIPMENT HIGHWAY
A parallel, synchronous, high speed highway system which provides communication between modules of a network node.

NETWORK OPERATING SYSTEM

A distributed operating system in which processes cooperate in a network-wide context, *between* host machines, as well as within the context of a single host machine.

NODE

(Physical) A unit of hardware providing a network terminating/access, or a link 'exchange' function.

NOS
Network Operating System.

PLEXUS
The name of the network project, borrowed from the Latin language.

PNOS
Portable Network Operating System.

POINT-TO-POINT LINK
A link which, at the *network* level, exists as a private connection between two entities (hosts). Obviously, data on this type of link need not be tagged with addressing (routeing) fields.

PORTABLE NETWORK OPERATING SYSTEM
An actual implementation of a *NOS*. (Marsden, B.W., Mimica, O., Ananda, A.A., Barclay's Microprocessor Unit, University of Manchester.)

ROUTED LINK
(conventionally *Multi-point* link) A single link (*highway* or *simple*) which communicates information pertaining to more than two network addressible entities (hosts).

SIMPLE LINK
In contrast with a *highway link*, a simple link is a normal, node-to-node link, which may be multi-point or point-to-point.

SIMPLE TERMINAL
In contrast with a *general terminal*, the simple terminal is a character display plus keyboard (VDU) requiring relatively low link bandwidth between itself and the processing resource (<19.2Kbaud). Furthermore, this type of terminal link will not involve *IPC* activity as described in this glossary.

STAR NODE
A 'link junction' or node which provides an inter-link exchange function.

STAR HIERARCHY
(See *'STAR NODE'*) A set of *star nodes* which are interconnected in a hierarchical manner. The normal method of *PLEXUS Hyper-level* node interconnection.

TERMINAL PROCESS
In a network context, a *terminal process* is one which does not contribute directly to the communication resources of the network. A *host* process usually falls into this category. The *Access Control Processor* system is not an example of a terminal process, because it hosts an IPC service. Similarly, an attatched process providing an 'in-core' exchange or node function would not be termed a terminal process.

TRANSMISSION LATENCY TIME
The combination of *inherent link access time* and any other local or remote overheads (usually *channel establishment time*) which delay the transmission of a message.

USABLE BANDWIDTH
The portion of a link-system's bandwidth which is directly usable by the application, that is, excluding the network address and control information, and other overheads. Refer, also, to *Gross Bandwidth*.

VSL10H
VSL20H
VSL20S
VSL30S
The actual set of PLEXUS *link systems* descriptors:

< link system type descriptor > ::=
V< parallel / serial designator >L< bandwidth designator >< highway / simple designator >;

< parallel / serial designator > ::= P|S;

< bandwidth designator > ::= 10|20|30; (10/50/140 Mbit/sec gross)

< highway / simple designator > ::= H|S.

SPECIFICATION FOR

CAMBRIDGE RING INTEGRATED CIRCUITS

ULA5C026

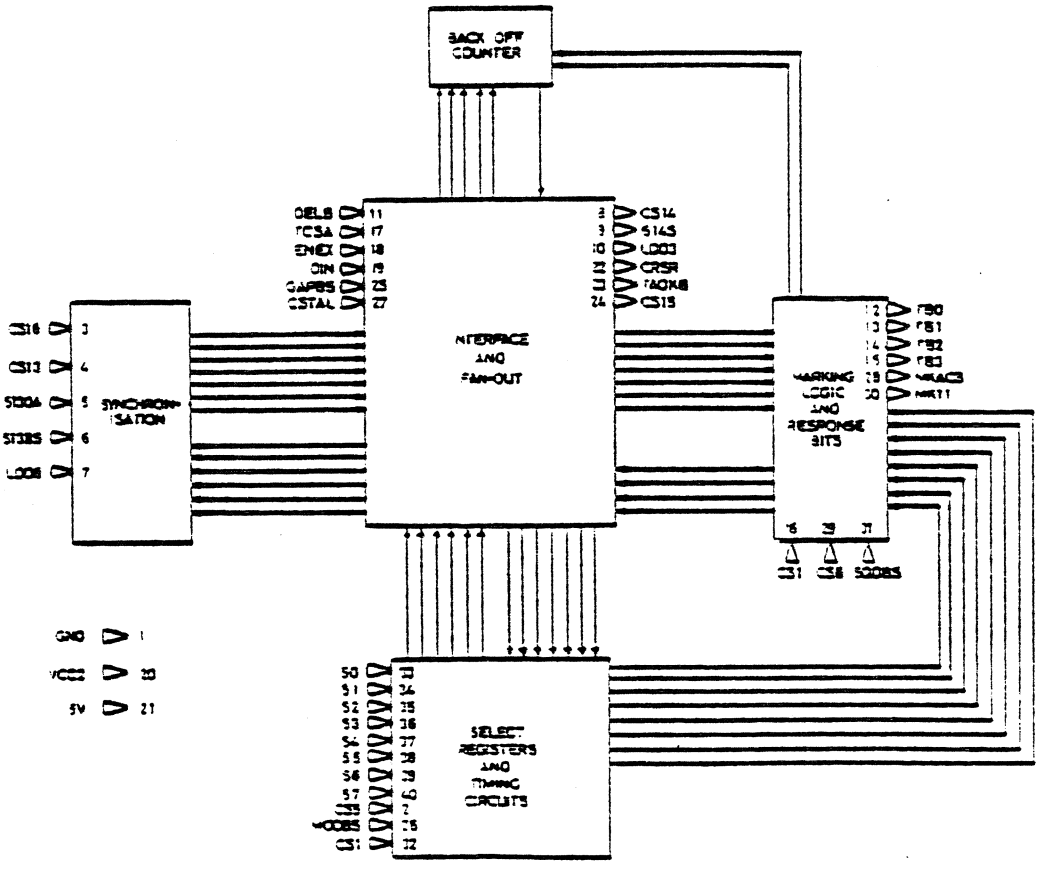
ULA5C027

ISSUE 1.0 MAY 1982.

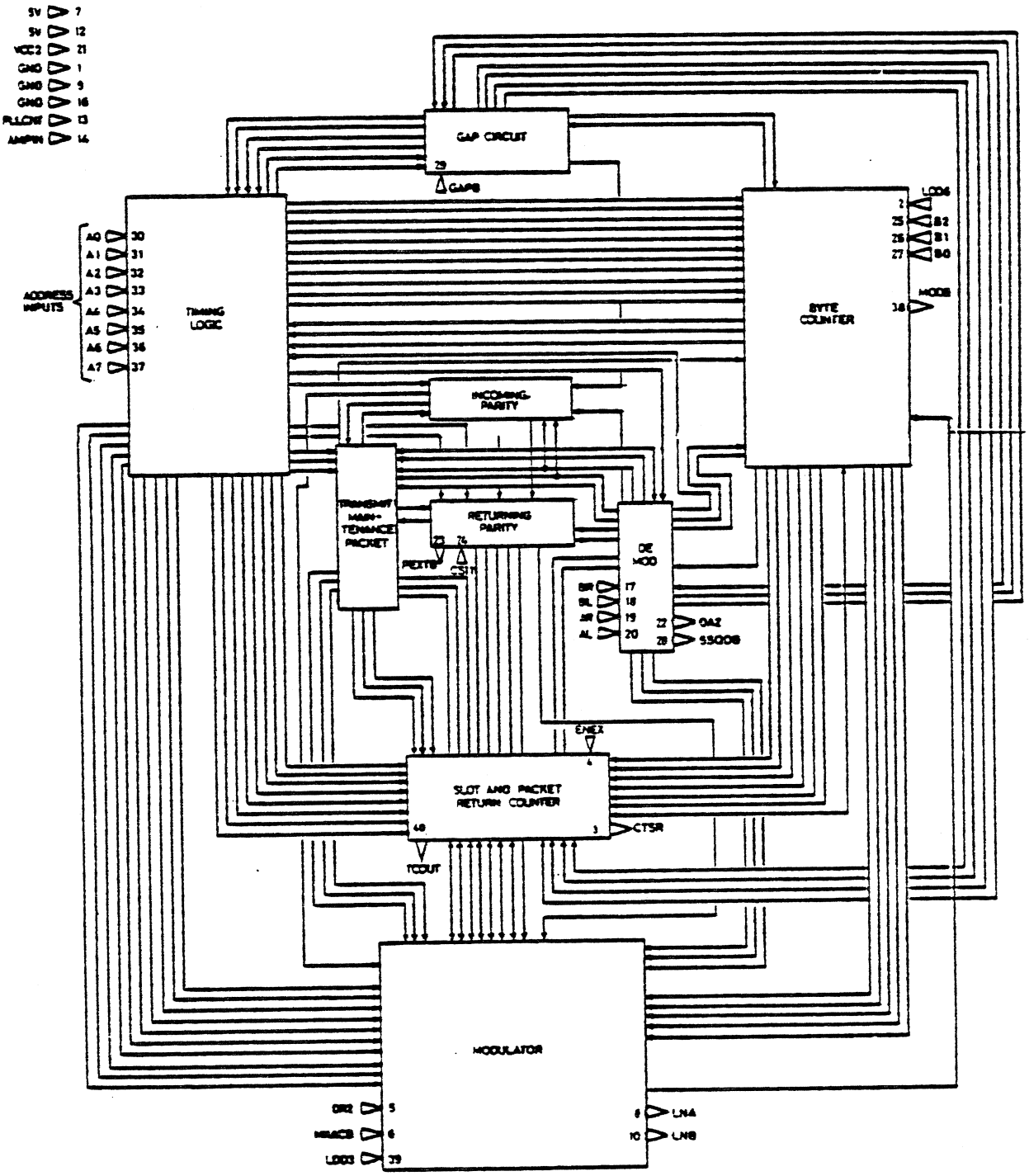
C O N T E N T S

1. DESCRIPTION

The ULA5C026 and ULA5C027 chip set is designed for the implementation of the Repeater and Station Functions on a Cambridge Ring installation. The use of these devices considerably reduces the number of components required for each node; a full description of the system design is given in the Applications Section of this specification.



SCHEMATIC OF STATION CHIP
ULA SC027



SCHEMATIC OF REPEATER CHIP
ULA SC026

2. ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

Supply Voltage.....7.0 volts
Operating Temperature.....0°C to +70°C
Storage Temperature.....-55°C to +125°C

ULA5C026

D.C. Characteristics over operating temperature range (V_{cc} = 5V)

Parameter		Test Conditions	Min.	Typ.	Max.	
Supply Current				180	240	mA
Low level input voltage	All Inputs				0.8	V
Low level input current	All E.F. type	V_{in} = 0V			-20	uA
	All TTL (TP&Tri)	V_{in} = 0V			-1.6	mA
	ENEX	V_{in} = 0V	150		450	uA
High level input voltage	All Inputs		2.4			V
High level input current	All E.F. type	V_{in} = 2.4V			600	uA
	All TTL (TP&Tri)	V_{in} = 2.4V			-750	uA
	ENEX	V_{in} = 2.4V	40		150	uA
Low level output voltage	LNA, LNB	I_{sink} = 20mA			0.5	V
	All other op's	I_{sink} = 8mA			0.5	V
High level output voltage	LNA, LNB	1mA	2.4			V
	All other op's	400uA	2.4			V
Tristate Disable Current	PEX TB O/P				±10	uA

2. cont/d.....

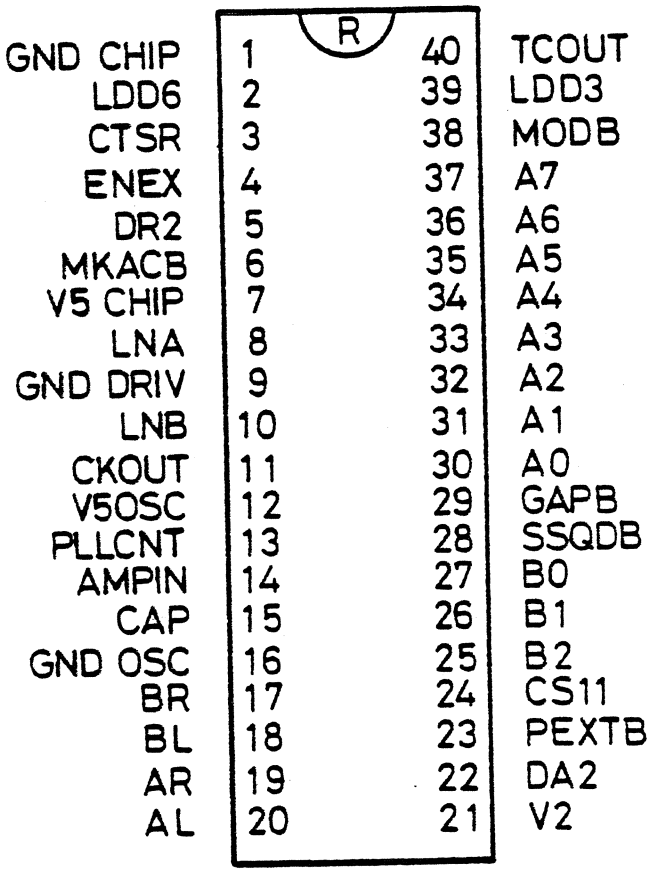
ULA5C027

D.C. Characteristics over operating temperature range (V_{cc} = 5V)

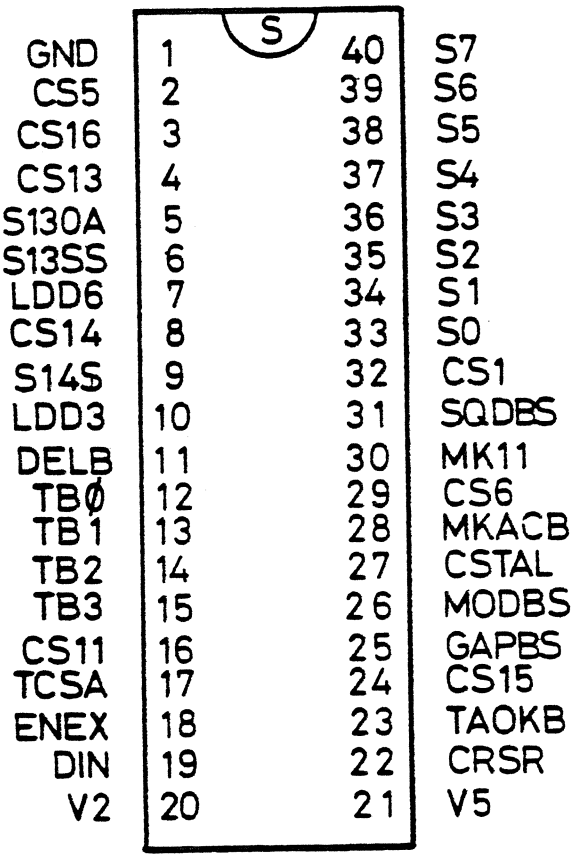
Parameter		Test Conditions	Min.	Typ.	Max.	
Supply Current				170	220	mA
Low level input voltage	All inputs inc. I/O pins				0.8	V
Low level input current	All E.F. inc. I/O pins	V_{in} = 0V			-20	uA
	All TTL	V_{in} = 0V			-1.6	mA
High level input voltage	All inputs inc. I/O pins		2.4			V
High level input current	E.F.:-CS5,CS11 CS6	V_{in} = 2.4V			20	uA
	E.F. All others inc. I/O pins	V_{in} = 2.4V			600	uA
	All TTL	V_{in} = 2.4V			-750	uA
Low level output voltage	All outputs inc. I/O pins	I_{sink} = 8mA			0.5	V
High level output voltage	All outputs and I/O pins	I_{out} = 400uA	2.4			V
Tristate Disable Current	All tristates				$\pm$10	uA

3. PINNING DIAGRAMS

3.1. ULA5C026 (Repeater)

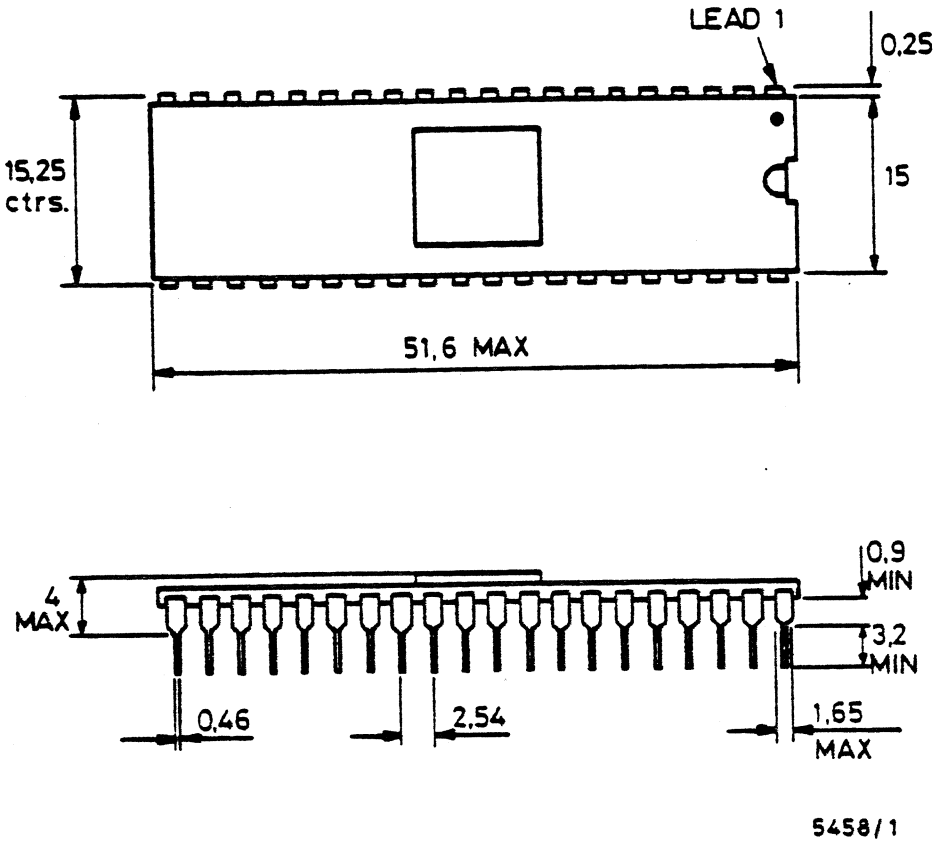


3.2. ULA5C027 (Station)

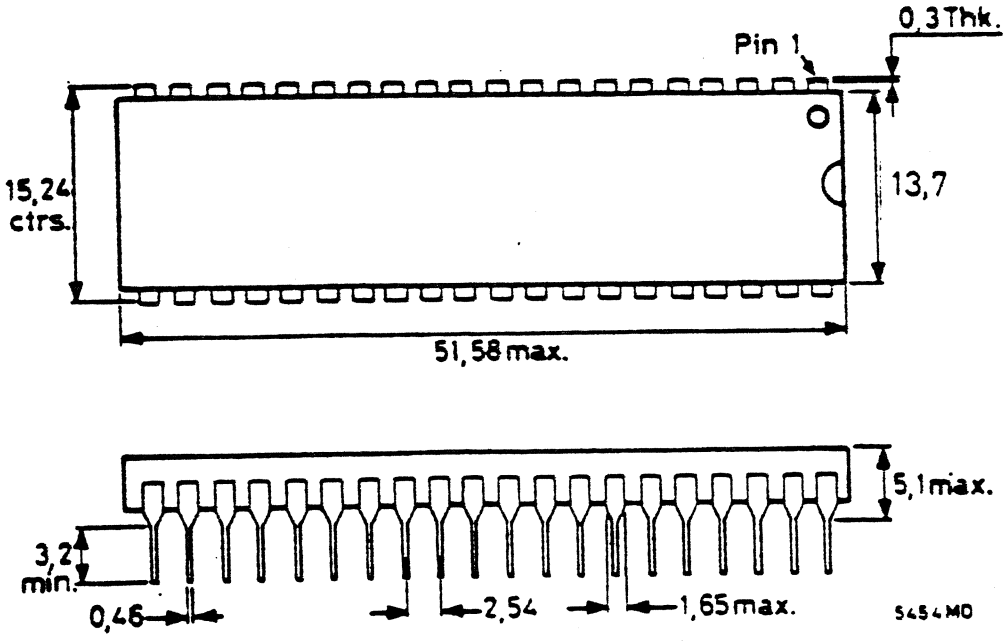


4. PACKAGE OUTLINES

All dimensions in millimetres



40 LEAD CERAMIC DIL SO119F



40 LEAD MOULDED DIL SO119F

5. APPLICATIONS INFORMATION

5.1. General

The two ULA chips implement most of the repeater and station logic. Connected to them are the transmit and receive shift registers. The combined unit will be referred to as the tap. The shift registers are used to buffer data entering and leaving the network and their length varies according to the number of data bytes in a minipacket. In general the system will use a parallel bus for communication with the attached device. Echo signals and bus amplifiers may be added if the connecting cable to the access box is long.

5.2. System Structure

An outline of a typical tap configuration is shown in Fig.1. The repeater contains the analogue circuits for driving and receiving from the ring transmission lines and for clock generation and phase-locking. It is connected to the station chip by a number of signals. These are used to clock the station chip and to interact during transmissions. The repeater chip also receives data from the transmission shift register which is clocked when a minipacket is launched. Each repeater is identified by a repeater number which is supplied by eight signals.

The repeater can function in repeater mode, maintenance mode, and normal mode. In repeater mode the logic works serially and performs the functions of demodulating data from the line, optionally gating new data, and modulating it again. In maintenance mode the repeater can also transmit maintenance minipackets which are identified by a repeater number and are transmitted under various conditions. The sending of maintenance information is a repeater function and thus all links are continuously monitored irrespective of whether the attached devices are switched on. In normal mode the repeater chip is used in conjunction with the station chip and transmits and receives minipackets for the user.

The station chip is sequenced by timing waveforms from the repeater. It is primarily used to synchronise external control signals and to provide responses to the access logic. The response signals can be divided into two categories: those which indicate the state of the tap logic and those which refer to the state of the returned minipacket. The latter are three-state and can thus be used on a bus. The station implements the select register which is loaded and read using eight bidirectional signals. It is also connected to the receive shift register and generates a clock for receiving minipackets.

The system bus can be implemented in a number of ways. Because the shift registers are external it is easy to arrange that they are loaded in parallel. However for most applications an 8 or 16 bit bus is likely to be sufficient. The bus can be duplicated if full duplex operation is required. Power to the ULA chips consist of a regulated 5V supply and optionally an intermediate unregulated power supply or at least 2.0V to reduce the package dissipation. All control signals are active in the low state.

5.2.1. Options

The LSI implementation of the Cambridge Ring provides a number of options for the designer. These are directly controlled by pin signals and there is no concept of programming internal registers. The most important options are listed below.

1. Number of data bytes in a minipacket: This option is specified using three pins on the repater chip and must be the same for all repeaters in one ring. Minpackets are loaded and read using the data bus and a minipacket can contain between 1 and 8 bytes of data. With care it is possible to mix minipacket sizes on a single ring.

2. Repeater number: This number identifies repeaters and is 8 bits wide. Normally this is not changed while the ring is working, although there are times - during the gap for example - when its value may be changed safely.

3. Control bits: Two additional control bits data fields have been incorporated in the minipacket format. They can be implemented to appear anywhere between the source address and the response bits and are not used by the internal tap hardware. In standard applications the control bits should be implemented immediately in front of the response bits. If the control bits are not used the minipacket format remains the same and the corresponding shift registers can now be simpler as they are not connected to the bus.

4. Broadcast address: Although the broadcast may lead to protocol difficulties it is provided because the ring lends itself to this mode of operation. A special address (255) is used and the response bits have special meanings. They indicate whether any tap has marked the minipacket as either accepted or unselected. In this mode the busy response is disabled and if it appears at the interface this means that the broeadcast minipacket has been accepted by some taps and marked unselected by others.

5. Delayed responses: If a returned minipacket is not marked accepted an additional progressive delay is added before the access logic is notified. This delay can be disabled to make the response available immediately.

6. Transmit on accepted: In this transmission mode the destination address and data of a new minipacket can be placed in the transmit shift register while the previous is still making its way round the ring. The new minipacket is launched if the previous was marked accepted. This makes it possible for the slot following the one marked empty to be used and also makes a fast double buffer scheme feasible.

7. Monitor Station: The design of the monitor station can be more or less complex according to the number of features that are required. In its simplest form a monitor station forces a single frame onto the ring. This facility can be implemented using a repeater chip, however, a complete monitor station will require some additional circuits.

8. Slot retention: With this option a tap may retain a slot it has found empty for any number of further revolutions. The responses indicate whether all the transmitted minipackets were either accepted or marked unselected. Responses indicating busy have no meaning in this mode and indicate that only some of the minipackets were accepted successfully.

9. Promiscuous mode: Gating of response bits can be disabled which allows a tap to listen to another transmission without affecting it in any way. In particular the repeater number can be changed during GAP so that the contents of the source, destination and data fields of passing minipackets can be made available. This facility may be used to implement a tap which gives statistics of ring usage.

10. Direct connection for short links. The line drivers and receivers are (TTL) compatible and thus for short links direct single sided drive can be used. This reduces the number of components on a board.

5.2.2. Additions and changes

A number oc changes have been made to the specification of the LSI **ring**. In some cases these improve performance or in others they are optimizations in view of experience. The most important changes are listed below.

1. Dynamic slot count: Each tap computes the number of slots currently circulating in the ring. This is done in the same way as in the early TTL version. The maximum number of slots in one ring is 16.

2. Maintenance upgrade: Maintenance minipackets are sent by repeaters independently of whether the attached device is functioning or not. A bit is set in each byte of the data field to indicate the type of maintenance minipacket being sent. They are sent for the following reasons:

Bit 0. Parity fault on upstream link.
Bit 1. Parity fault between transmitted and returned minipacket.
Bit 2. Modulation error.
Bit 3. Slot count changed.
Bit 4. Repeater enabled.
Bit 5. Repeater disabled.

When ring is broken maintenance minipackets will be continuously sent from the repeater after the break.

They will indicate modulation error although some of the other types may be generated as well.

3. Synchronisation of control signals: The settling time for asynchronous signals has been reduced to about 4 clock times and overlap is allowed up to the end of the destination field. This means that a sophisticated access logic can be made to receive from successive slots. On the transmit side advantage can be taken of the gap to settle.

4. Error checking of returned minipackets: The bit-by-bit comparison of transmitted and returned minipackets has been simplified to verifying the return parity is the same as the transmitted parity (of unambigous bits, i.e. excluding response bits).

5. Enable synchronisation: The enable signal has been synchronised so that when a tap is enabled transmissions can only proceed after one revolution of the ring. A tap is allowed to disable only after any outstanding transmissions are marked empty. These features ensure that when a mode is switched on or off no spurious ring errors are generated.

6. Ring available signal (CS17): This signal has been deleted from the specification. If required it can be reconstructed with some external logic. However in most access logics the presence of a regular gap waveform will be sufficient to indicate that the ring is working.

7. Gap length: The maximum length of the allowable gap has been reduced from 256 bits to 125 bits.

8. Select register: The select register has been designed so that when a station is disabled it is forced to all zero (nobody). Thus a tap following the correct enable sequence will start operating with the select register set to zero.

9. Transmit and receive commands: These active low commands are triggered by the negative rather positive going edge.

10. The clock CDEM supplied by the repeater is negative edge.

11. The data gated out by the repeater (DA2) is true and not inverted as in the present system.

5.3. Repeater Description

The input and output signals for the repeater can be functionally organised into groups as shown in Fig.2. The following papragraphs provide a description of the analogue

components and digital signals and refer to the timing chart in Fig.5.

5.3.1. Analogue components

1. Two differential line receivers: Each receiver can be driven by a differential signal of 10mV peak to peak with 1/3 hysteresis. It is best centred at 3 volts and then has 3/2 volts of common mode suppression. Thus it can readily be used at the end of a long balanced and terminated cable - typically a 100 ohm line. If the appropriate terminal is driven by an output directly, then the amplifier works equally well because the other input is held at 15/8 volts by an internal resistance.

2. Two output drives: These are single ended outputs capable of driving 20mA into the load and sinking 12mA from the load at TTL levels. A short line can be driven directly, while a line up to 300 metres can be driven via a small transformer, comprising 5 to 10 turns on a small ferrite ring. This transformer can be used to step down the output voltage if required and provides isolation.

3. A voltage controlled oscillator: This supplies a clock whose frequency is defined by an external capacitor. The frequency is controlled by a control voltage between 3V and 5V as shown in Fig.8. At the centre frequency the duty cycle is 50% but at other frequencies it may alter slightly. Because the values of internal resistors vary with manufacturing tolerances by a factor of 2 it may be necessary to trim the timing capacitors to align all centre frequencies. The resistor values also change with temperature so the operating frequency will shift as the chips warm up.

4. A phase detector and amplifier: The phase of the two incoming signals is compared with the oscillator and an error signal is generated as shown in Fig.9. This is integrated by an external filter and controls the oscillator via an internal amplifier. When there is no incoming signal the oscillator control is centred so that starting is improved and wire breaks will usually allow the downstream repeaters to work correctly.

5.3.2. Signal description

Power (V5CHIP, V2, V5OSC, GNDCHIP, GNDDRIV, GNDOSC)
Pins 7, 21, 12, 1, 9, 16.

Two regulated supplies are required. The first (V5CHIP) is used as the general chip power. The second (V5OSC) is used for the oscillator only. The two supplies can be commoned if sufficient decoupling is used. In addition an intermediate supply (V2) of at least 2.0V can be provided to reduce package dissipation. This supply is internally regulated and can be implemented using a resistor of approximately 27 ohms.

GNDCHIP is the general chip ground.
GNDDRIV is the line driver ground.
GNDOSC is the oscillator ground.

Phase-locked loop control (PLLCNT, AMPIN, CAP)
Pins 13, 14, 15.

PLLCNT is a waveform whose mark-space ratio changes according to the difference in phase between the incoming data and the internal clock. The mean value increases when the oscillator is required to slow down. The phase comparator is shown in Fig.9.

AMPIN is the oscillator frequency control. It is normally connected to PLLCNT through an integrating network. A typical filter is shown in Fig.10.

CAP : The capacitor to define the mean operating frequency is connected between CAP and GND. The system can also be clocked externally through this pin although care has to be taken as this is not a TTL compatible input and requires swings of 3V to 5V to operate.

Ring drivers and receivers (LNA, LNB, AR, AL, BR, BL)
Pins 8, 10, 19, 20, 17, 18.

LNA and LNB are the single sided line drivers. AR and AL are the two inputs to one of the line receivers and BR and BL the inputs to the other. The amplifiers have high gain so that the incoming pair should be closely coupled all the way to the chip pins.

Repeater number (A0-A7)
Pins 30-37

The repeater number is internally serialised from these pins. There is no constraint as to the significance of the bits although by convention A0 is least significant. The signals are TTL compatible.

Number of bytes (B0-B2)
Pins 27, 26, 25

The number of data bytes in a minipacket is defined according to the following table:

Number of bytes	B0	B1	B2
1	0	1	1
2	0	0	1
3	1	0	1
4	1	1	1
5	0	1	0
6	0	0	0
7	1	0	0
8	1	1	0

The signals are TTL compatible.

Interconnection between chips (COUT, MODB, GAPB, TCOUT, SSQDB, MKACB, LDD3, LDD6)

Pins 11, 38, 29, 40, 28, 6, 39, 2

These signals may be used externally if required.

COUT is the clock for the station chip.

MODB is a timing waveform used to sequence the station chip. There are eight pulses per minipacket as shown in Fig.5.

GAPB is a waveform representing the gap and is one bit delayed with respect to the real gap. It is used to reset the station chip to a standard state.

TCOUT indicates to the station chip that a transmission is in progress. It goes high when an empty slot is found and low just prior to the returning response bits. In slot retention mode TCOUT goes high when an empty slot is found, is low during each revolution for one bit time during PCO2, and goes low again just before the response bits.

SSQDB is valid from the source address to the next SOP bit or GAP. When high at this time it indicates that the destination address of the incoming minipacket is the same as the repeater number (AO-A7).

MKACB is a signal to the repeater chip for marking response bits when the tap is receiving minipackets. In maintenance mode it is disabled. In repeater mode this signal is equivalent to GR in the TTL design.

LDD3 is the transmit request to the repeater.

LDD6 is the retain slot command to the repeater.

Responses (PEXTB, CS11)
Pins 23, 24

PEXT is the tristate reponse to indicate that a parity error has occured between a transmitted and returned minipacket.

CS11 is the gating waveform which enables PEXTB.

Miscellaneous (CTSR, DR2, DA2, ENEX)
Pins 3, 5, 22, 4

CTSR. This waveform is used to clock the transmit shift register. Eight positive edge clocks are put out to shift in the destination address. This is followed by a gap of eight bits during which the repeater inserts its own address. This is followed by the appropriate number of clocks to shift in the data bytes from the shift register. Finally four further clocks are provided to input the two control bits (there are four such clocks to enable the control bits to be stored in a four bit shift register).

DR2 is an input pin for data to the repeater. In repeater mode this signal is equivalent to DR in the present system.

DA2 is the demodulated data from the line receivers. In repeater mode this signal is equivalent to DS in the present system.

ENEX is a special input pin which is decoded into three internal values. It is used to switch between normal, maintenance and repeater modes. The values taken are shown below:

ENEX	MODE
TTL-Hi	Normal
TTL-Lo	Maintenance
disconnected	Repeater

5.4. Station description

The input and output signals for the station are organised into groups as shown in Fig.3. The following paragraphs provide a description of the digital signals. All external commands are active low except where stated otherwise.

5.4.1. Signal description

Power (V5, V2, GND)
Pins 21, 21, 1

There is a single 5V supply and an optional internally regulated supply of at least (V2). There is only one ground (GND).

Transmit and receive commands (CS16, CS13, CS130A, S13SS)

Pins 3, 4, 5, 6

CS16. Receive command: The minipacket in the receive shift register is discarded. The currently passing minipacket can be received if this signal is no later than the third bit of the destination address. If this command is issued while reception is taking place it will be ignored.

CS13. Transmit command: Transmission will take place in the next empty slot. The latest time at which this command can be given is fourt bits before the full/ empty bit. Thus unless the gap is used to settle it is impossible to arrange for a minipacket to be sent in the slot following the one marked empty. If this command is given while a transmission is taking place it will be ignored.

S130A. Transmit on accepted: The minipacket will be launched in the next empty slot if the previous transmission was accepted. If this command is issued while TCSA is low it will be ignored.

S13SS. Transmit into same slot: The minipacket in the transmit shift register will replace the one currently on the ring. If this command is issued while TCSA is low it will be ignored.

Transmit and receive replies (CS15, CS14, S14S, TAOKB, DEL8)

Pins 24, 8, 9, 23, 11

CS15. Receive reply: This signal will go to one at the beginning of CS16 and become zero again when the receive shift register has again been loaded.

CS14. Transmit reply: This signal will go to one at the beginning of either CS13, CS130A or S13SS and become zero again when the responses are valid again. This may or may not include an additional back-off delay as set by DEL8.

S14S. Packet shifted out: This signal will go to one at the beginning of either CS13, CS130A or S13SS. It will become zero again when the shift register has been emptied and can be loaded again.

TAOKB is a response which indicates whether the transmit on accepted command launched a minipacket. This reponse is available at the end of S14S and when low indicates that the minipacket was transmitted.

DEL8 when low enables the back-off delay on transmitted minipackets which return not accepted.

Select register (CS1, CS5, S0-S7)
Pins 32, 2, 33-40

CS1 sets an eight bit select register latch from pins S0-S7. This is copied into the select register proper once per minipacket. The command has to be issued by the third bit of the destination address for the new value to be used to receive the current minipacket.

CS5 gates the contents of the select register to tri-state outputs S0-S7.

S0-S7 are bidirectional pins for loading and reading the select register.

Responses (TB0, TB1, TB2, TB3, CS11, CS6, MK11)
Pins 12, 13, 14, 15, 16, 29, 30

TB0: Tri-state transmit response indicates ignored.

TB1 : Tri-state transmit reponse indicates unselected.

TB2: Tri-state transmit reponse indicates accepted.

TB3: Tri-state transmit reponse indicates busy.

MK11 (active high): Tri-state receive response indicates the tap has rejected a minipacket from an unselected source since the select register was last set.

CS11: Gate tri-state transmit responses TB0-TB3.

CS6: Gate tri-state receive status MK11.

Interconnection between chips (CSTAL, MKACB, MODBS, SQDBS, GAPBS, LDD3, LDD6, TCSA)
Pins 27, 28, 26, 31, 25, 10, 7, 17

These signals may be used externally as required. The corresponding signal name on the repeater chip is shown in brackets.

CSTAL (COUT) is the clock to the station chip.

MKACB (MKACB) is the gate response command to the repeater chip.

MODBS (MODB) is a timing waveform for sequencing the station chip.

SQDBS (SSQDB) indicates address recognised.

GAPBS (GAPB) indicates the gap and is used to reset the station chip.

LDD3 (LDD3) is the transmit request to the repeater chip.

LDD6 (LDD6) is the transmit into same slot command to the repater chip.

TCSA (TCOUT) is an input indicating transmission in progress.

Miscellaneous (DIN, ENEX, CRSR)
Pins 19, 18, 22

DIN is the data input to the station chip.

ENEX is the enable signal. It is internally synchronised by GAP.

CRSR is the positive edge clock for the receive shift register. This clock is enabled at the beginning of the source address field and disabled when a minipacket has been received. It includes four pulses to shift in the control bits so that these can be held in a four bit shift register.

5.5. Compatible system design

Because of the extra control bits it is impossible to design a LSI ring tap completely compatible with the current TTL system. However the repeater can be used in repeater mode on present rings. Furthermore the control bits can be made transparent to the user and thus the access box interface can be made the same as now. A system of this kind is shown in Fig.6. where the two dummy control bits appear direclty in front of the response bits. As well as the transmit and receive shift registers a number of other components are required to implement echo signals, line buffers etc. Like the TTL version the compatible LSI system is based on a sixteen bit transmit data bus and a sixteen bit receive data bus. These are primarily used for loading and reading the shift registers.

5.5.1. Repeater connections

The repeater signas are connected as shown in Fig.6. with the interconnecting lines being taken to the station chip and the control lines going to the access box interface. Inverted GAPB is used to implement the ring available signal CS17. The only repeater connection to a system bus is for response PEXTB which is available on bit 4 of the transmit bus and is gated by CS11. Transformers are used for common mode noise rejection and DC isolation. These have the same specification as in the TTL design. On the ring input side the line is terminated to 91 ohms. On the ring output side the single sided drivers are converted to balanced drive by switching a current through

the output transformer using a capacitor. A pull down resistor is used on input ENEX to ensure the repeater chip functions in maintenance mode when the station chip is powered off or disabled.

5.5.2. Station connections

The station chip is sequenced by the repeater chip. It is also connected to the least significant bytes of both the transmit data bus and the receive data bus. On the receive side the connection is for reading and writing the select register S0-S7. On the transmit side the connection is for the minipacket responses which are gated by CS11. There is another connection to bit 6 of the receive bus for response MK11 which is gated by CS6. The unused inputs S12SS and S130A are tied high. The back-off delay for minipackets that have not been accepted is switched in my making DELB low. When the station chip is powered up DISABL should be low. Once power has been established for some time DISABL can be made high thus enabling both the repeater and station chips.

5.5.3. Shift registers, echo signals and bus amplifiers

The transmit shift register is made up of four 74L165 chips for destination address and two data bytes. A 74LS94 is used to implement the dummy control bits which are not used in the compatible system. The transmit shift register is connected to the transmit bus and is loaded in parallel by the same control signals as in the TTL system. The output is shifted round to minimise subsequent loading delays.

The receive shift register is made up from three 8546 chips for the source address and two data bytes. A 74LS94 is again used to implement the dummy control bits. The 8546 incorporates tri-state parallel outputs and the receive shift register can thus be read by the same control signals as in the TTL version. For correct shifting CS15- is also connected to the 8546's.

The transmit and receive gate echo circuits are made up from two 74LS30 and two 74LS04 circuits. Teh echo signals can be omitted in some circumstances. Their specification is the same as for the TTL system and they can be used for handshakes on long bus lines. The 74LS04's drive the return path directly and 74S04's can be used if more power is required.

For compatibility reasons all lines on the bus are buffered. This will not be necessary in most practical applications. Because the least significant byte of both the transmit and receive bus is bidirectional 74LS245 transceivers are used as buffers and are steered by signals CS1 (receive) and CS11 (transmit). All other lines are unidirectional and are amplified by 74LS244 buffers.

5.6. Typical LSI system design

In a typical LSI tap the control bits will be implemented as a part of the minipacket format. The system shown in Fig.7. is similar to the present TTL version and can be used in applications where the inferface cables are short and thus echo signals and bus amplifiers can be dispensed with. Typically this will be the case where the access box is on the same board as the ring tap.

The repeater is connected to the ring lines through transformers. These can be omitted for short links where the outputs can drive one side of the receivers directly. All other connections for both the repeater and station chips follow the compatible system design above. The major difference is in the implementation of the shift registers. On the transmit side a four bit shift register with asynchronous load and positive edge triggered clock is required as well as the 74LS165's. In Fig.7. a 74LS96 is used in four bit mode. This chip does not have an asynchronous load but individual bits can be preset. Thus if it is first cleared then it can be preset where required. This can be achieved using a single inverter although care has to be taken to ensure that the delay through the inverter to the preset input is sufficient to set the register. On the receive side a four bit shift register with tri-state outputs and a positive edge clock is required as well as the 8546's. In Fig.7. a 74LS395 is used which requires an inverter to provide the negative edge clock. Because of the inversion the clock to the 74LS395 is delayed but this only effects the tolerances on the serial data input from the repeater chip.

The control bits can be either placed on the data bus or can be made available directly to the access logic. In Fig.7. the control bits have been allocated to bits 8 and 9 of the data bus. This means that they can be read at the same time as the address field. Because separate control lines are used for gating the control bits (CS18, CS19) the data bus can be narrowed to eight bits. If the system is to be only used with a sixteen bit duplex data bus then some control signals can be omitted and the control bits can be gated by CS15 and CS10.

The design is easily altered to increase the number of data bytes in a minipacket. This requires the appropriate number of 74LS165's and 8546's and for B0-B2 to be set correctly. The bus architecture can be arranged as required; in particular a very wide structure can be used. However in most applications with a large number of data bytes in a minipacket a narrow bus will be used. The control lines for reading and writing data will be encoded to reduce the sixe of the access box interface.

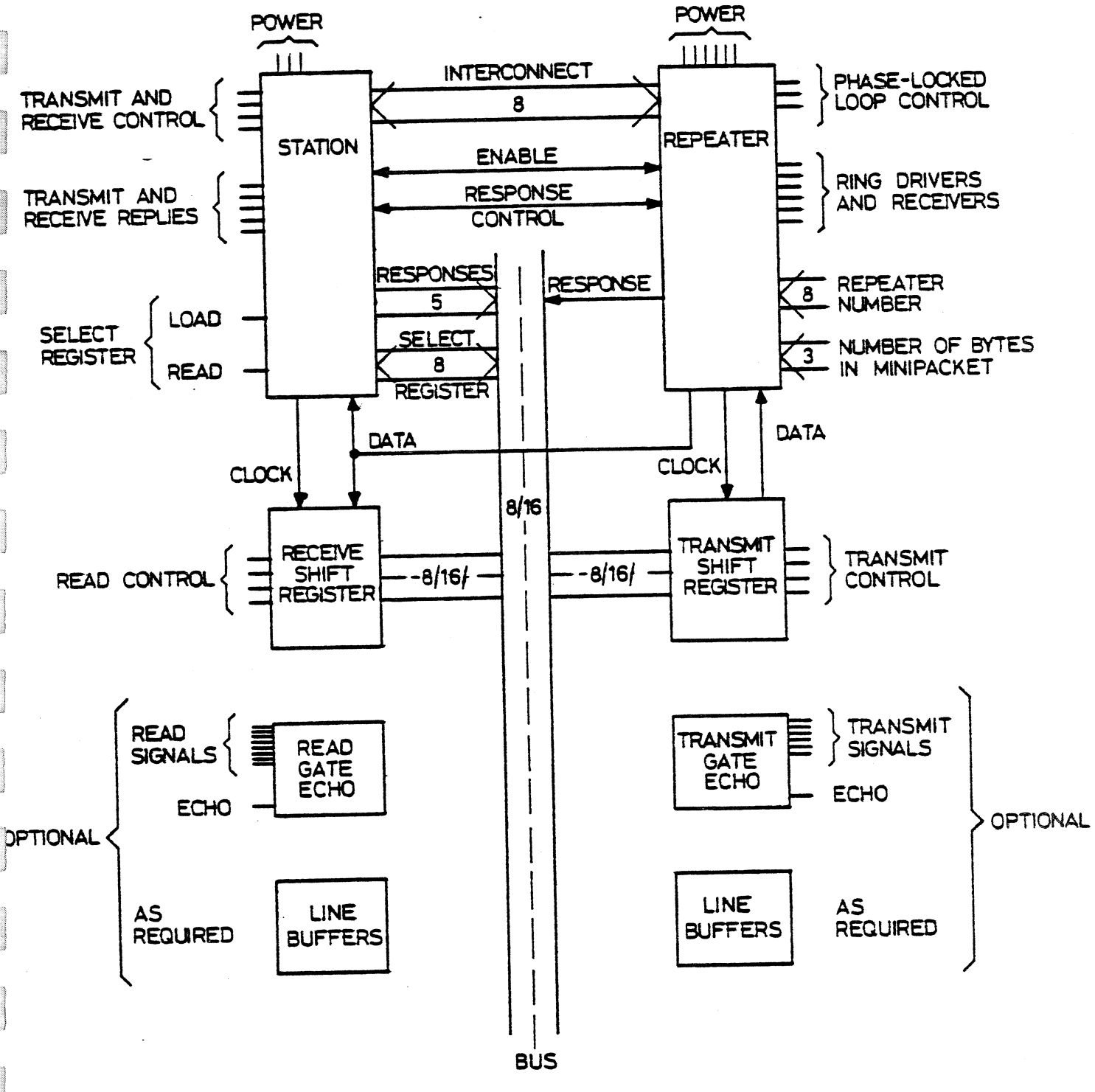


Fig.1. LSI System Design

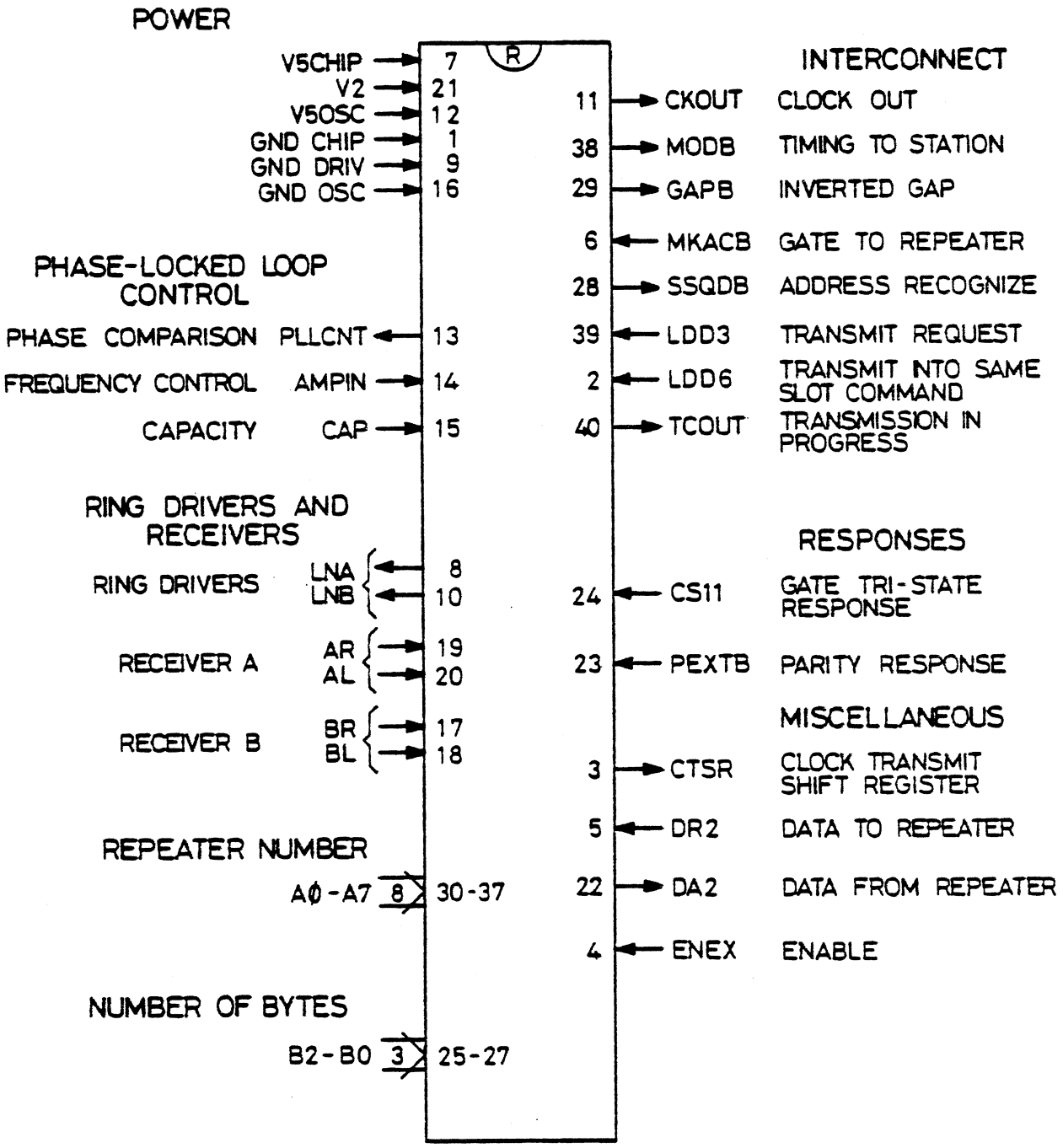


Fig. 2. Repeater Interconnections

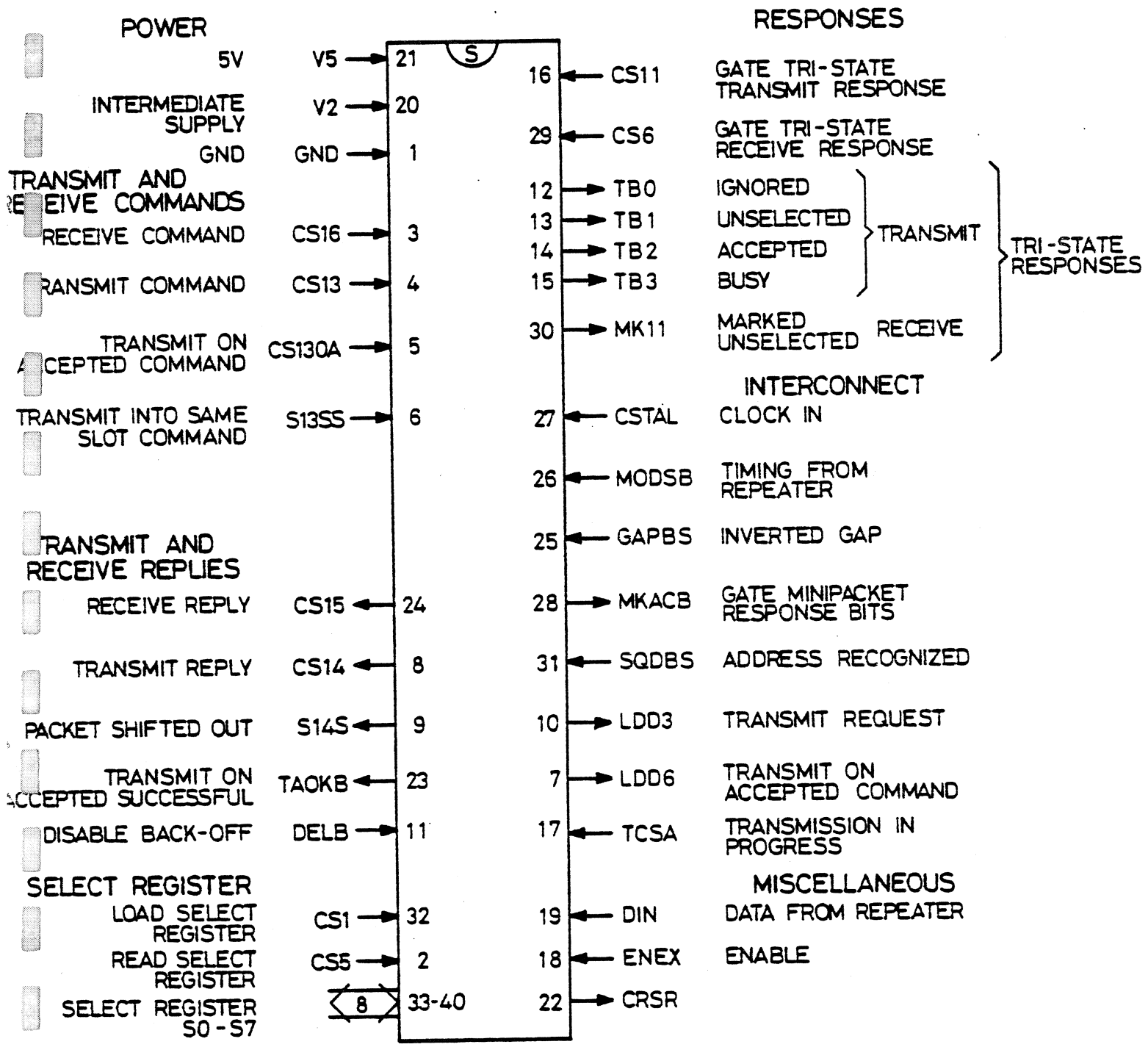
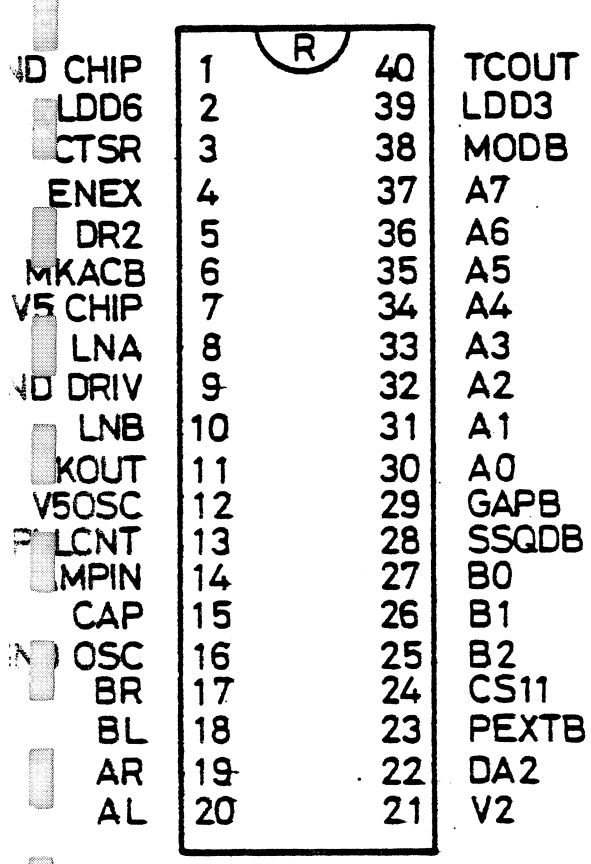


Fig. 3. Station Interconnections



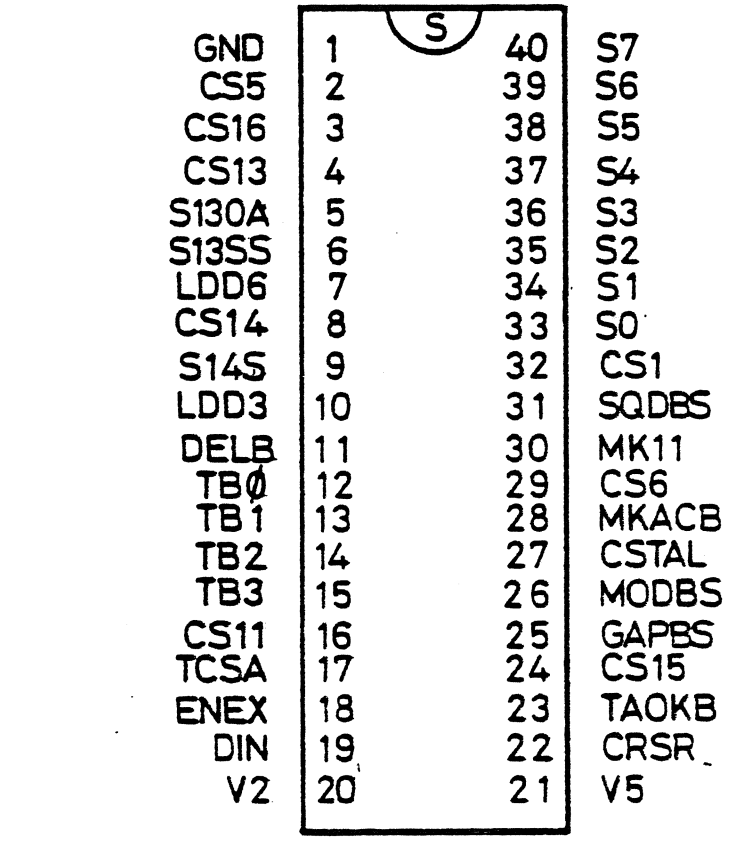


Fig. 4. Device Pinning

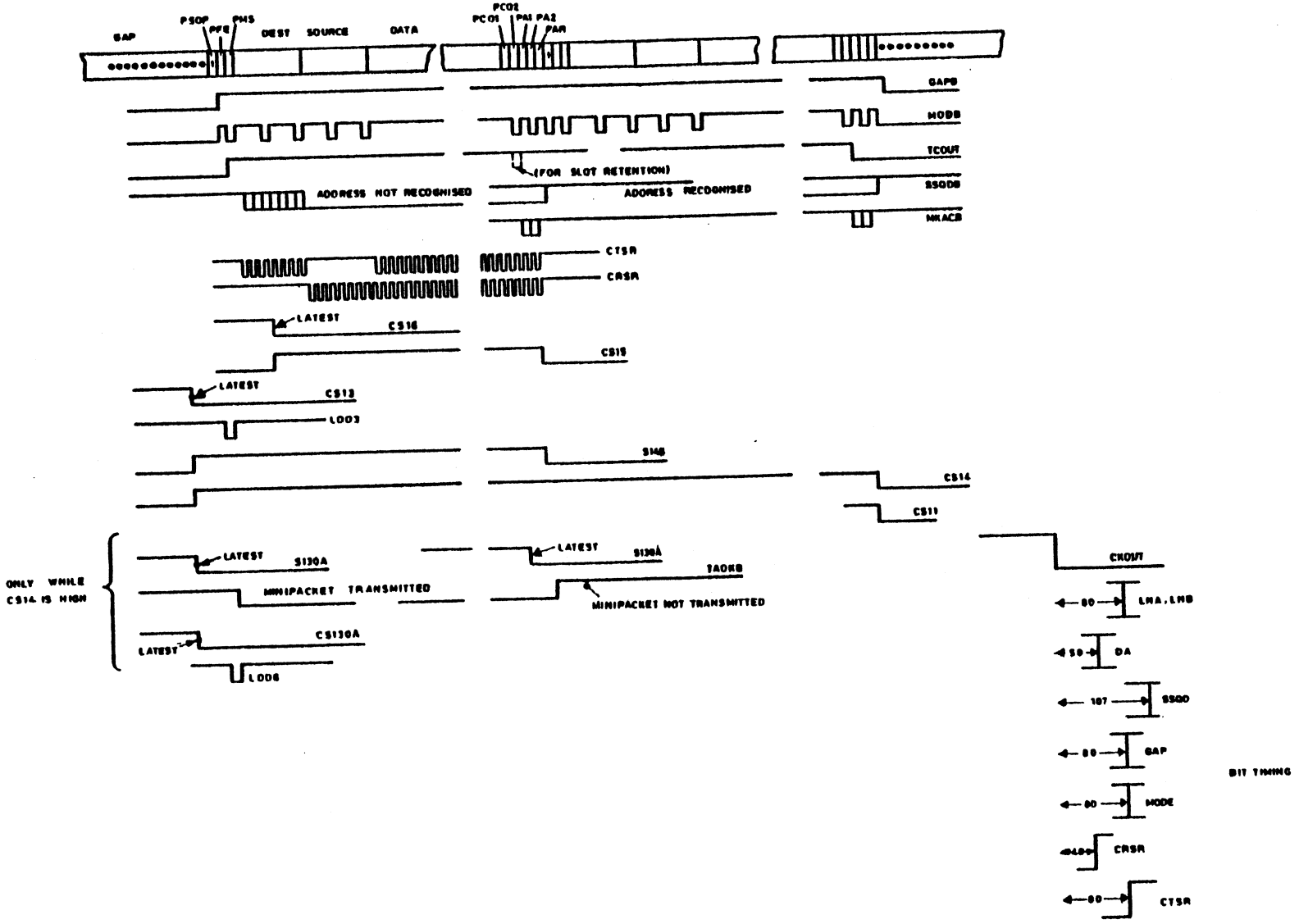


Fig. 5. System Timing

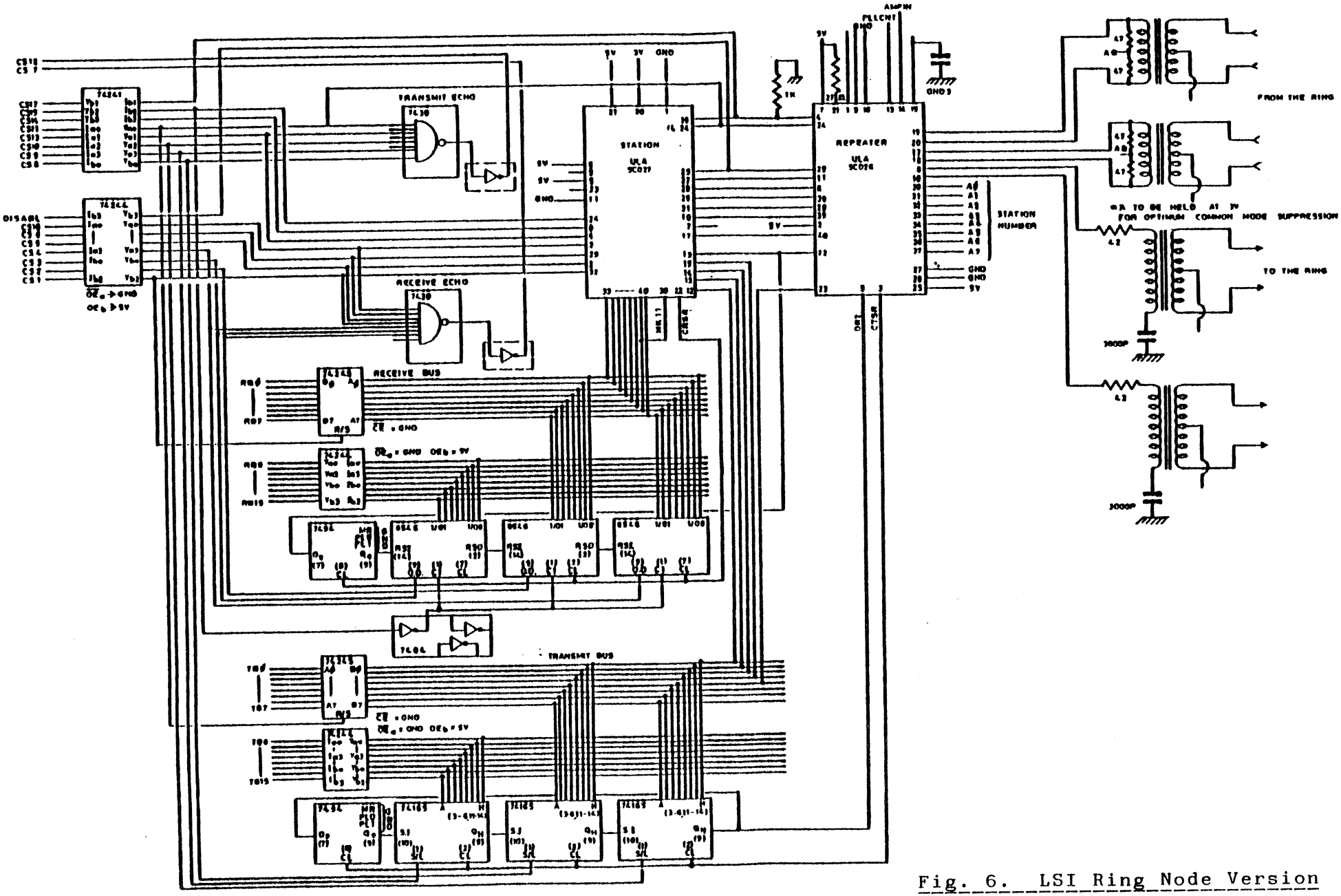


Fig. 6. LSI Ring Node Version

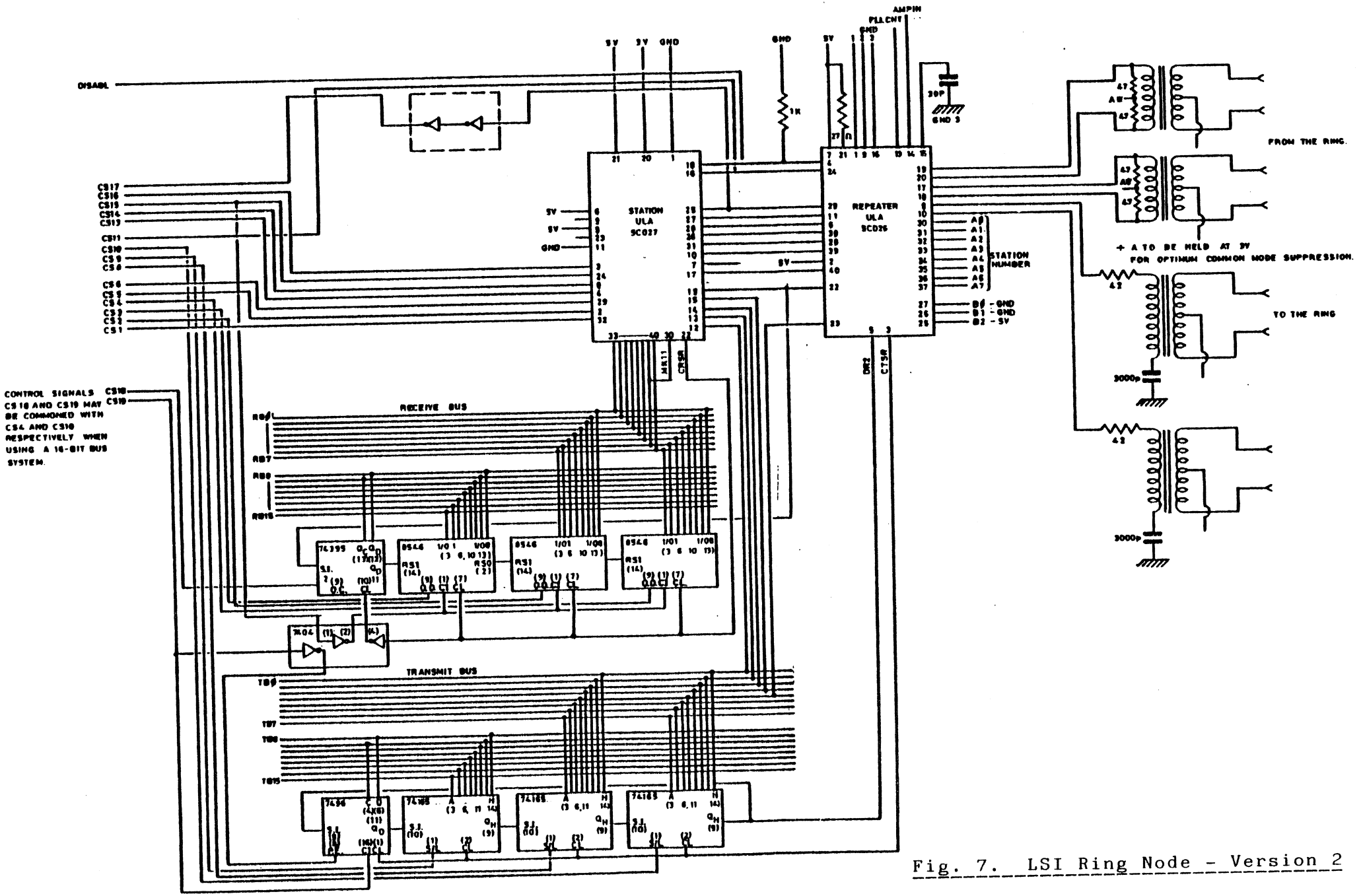


Fig. 7. LSI Ring Node - Version 2

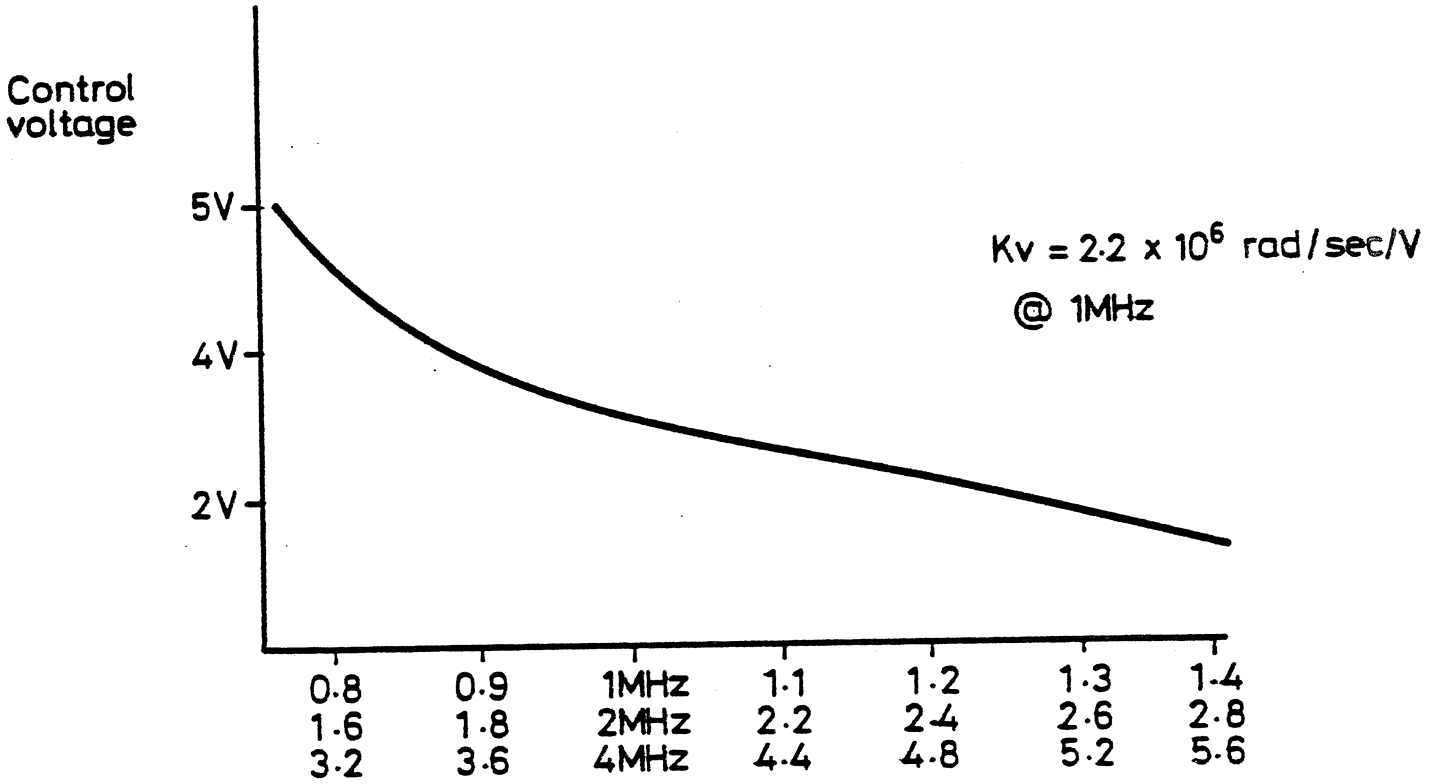


Fig. 8. VCO Control Range

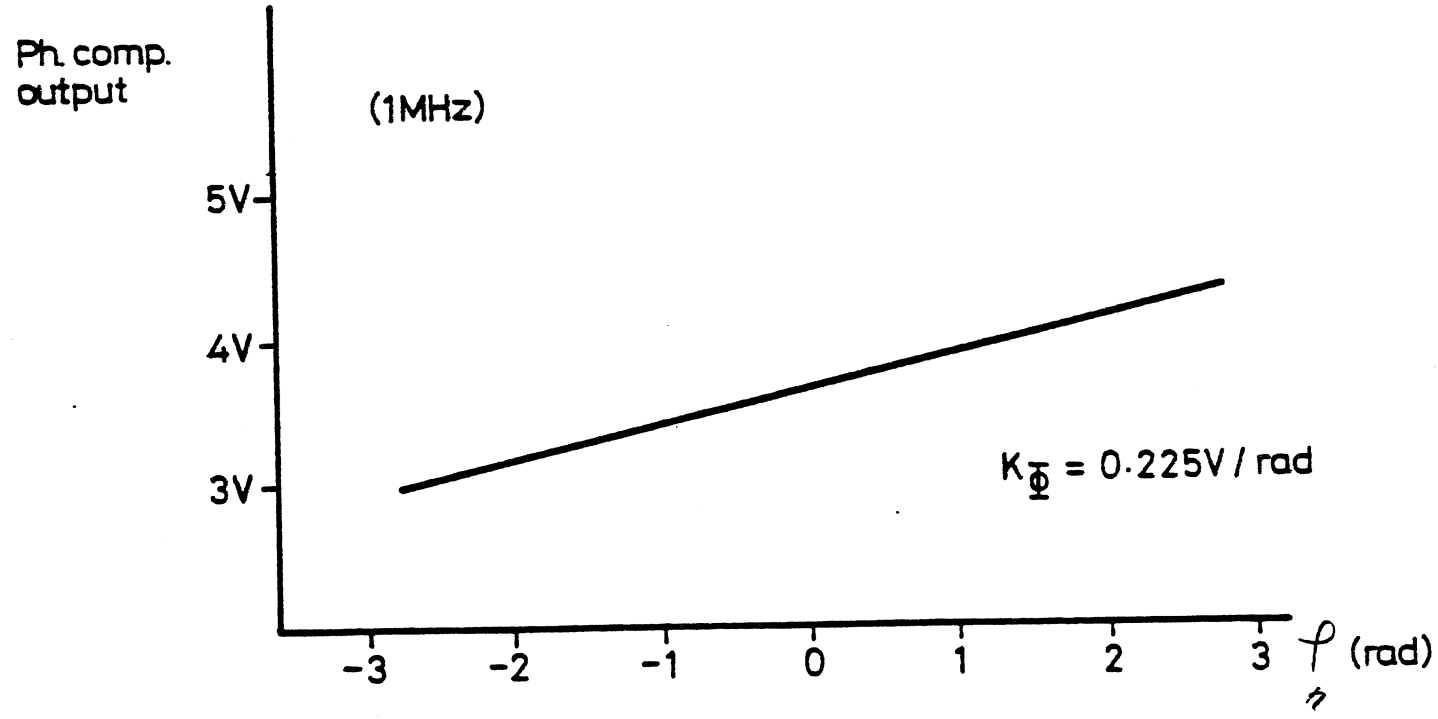


Fig. 9. Phase Comparator Output

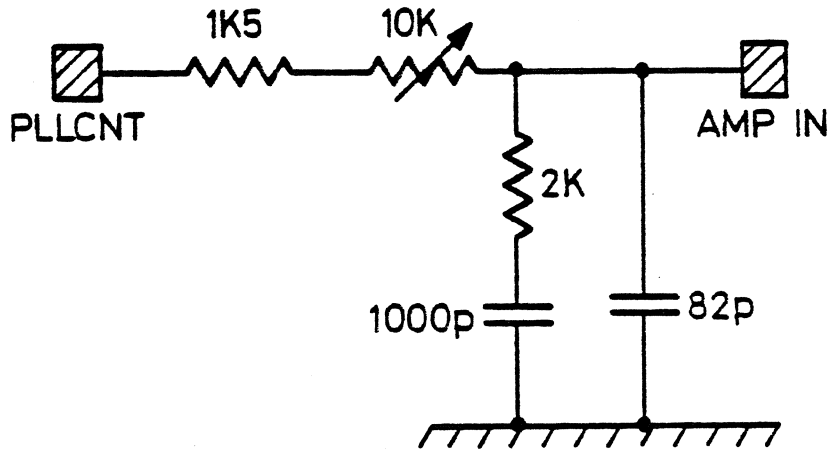


Fig.10. Typical Filter at 4MHz

Computer Laboratory,
Cambridge University, UK.
9 Oct 1981

Cambridge Digital Communication Ring.

Contents

Introduction

Various computers and devices are connected by a communication ring, arranged so that each pair of computers or devices can communicate with each other. The ring consists of a number of links going from one station to the next, and the messages are transmitted as blocks of packets of 38 bits, each of which contains 16 message bits.

Each link is short so that little drive power is needed for the link and the data rate can be high. Care is needed in the design of the repeaters as each packet passes through all the repeaters.

A packet is transmitted from the transmitting station to the receiving station and returned to the transmitter marked by the receiving station to indicate rejected, accepted, ignored or busy. The logic at the transmitter then takes proper action.

The ring itself conveys the packets in a serial manner, and as each packet has a source and destination number the ring is at the same time carrying packets from a number of sources to their various destinations.

The bit speed of the ring is about 10 megabits/sec and the empty packet scheme is used for transmitting packets. In this scheme the

complete packet is loaded into a shift register and the shift register is used to fill the first empty packet that passes the station. The packet is marked when it passes the destination and as it returns, is compared with the original packet and the passing packet is marked empty.

There are a number of consequences of this design. Firstly, each transmitter only transmits packets one at a time so that each transmitter has at most one packet in flight. Secondly, the packet has to wait until an empty one becomes available, but then the ring transit is swift. Thirdly the delay to the ring at a station is three bits, (and could be made one half bit with a more costly design) so that the total delay around the ring is small. Fourthly, the amount of equipment in the ring circuit is small so that reliability is good. Fifthly, the traffic is self regulating and needs no central control. The passage of the empties causes a "round robin" system to be in force, so that no station can block the ring and each station affects all the other stations equally.

There is a monitor station needed during turn on and to take action after certain types of error.

Design considerations

Experimental evidence indicates that the standard teletype cables or twisted pairs allow about 10 megabits/sec without particular precautions against noise, differential attenuation etc. as long as the signal is regenerated fairly frequently, say every 100 metres.

The cable system has to transmit clock, data, and power. Power is needed to keep each repeater working when the local power is switched off. It merely has to energise enough of the station to ensure that data is regenerated as it passes the station. We have chosen to use two pairs and to use them with a special symmetrical self clocking scheme which uses both pairs for clock and data and sends DC at about 50 volts around the ring as the power source, one pair being negative the other positive and both halves of each pair taking the current in parallel.

The modulation system amounts to using phase modulation along each pair with their digits half out of step with each other. However, the rules for both pairs are easier than phase modulation and the system cannot be half a bit out of synchronism. At each clock instant a change on one of the pairs indicates a zero and a change on both the pairs indicates a one. The single changes use one pair on odd clocks and the other on even clocks. The clock is recovered readily as there is always at least one change at each clock instant.

The system is completely balanced and transformers are used at each repeater to isolate the DC power and reject noise and common mode interference from each pair. The repeater is designed to demodulate and modulate the ring signals, so that the interface between a repeater and station is very straightforward, and independent of the actual modulation system. Thus logically, different parts of the ring can use

different modulation systems. In fact one section of the ring uses two optical fibres in a cable supplied by GEC Ltd. The attenuation and dispersion of the optical fibres is so low that this link could be up to some kilometres in length, without needing any intermediate repeaters. Long wired links can be made readily using better cable. A link of one kilometer using expensive low loss cable has been made to work. It is relatively easy to make long links even of the standard cable if a terminating filter is used to reduce the frequency sensitive attenuation and restore the pulse shape, and links of up to half a kilometer may be made cheaply this way while still using the standard repeaters.

The mode of operations between stations is for a transmitter to send a packet to a receiver, which copies it and marks it 'accepted'. When the packet returns to the transmitter it is marked empty.

The receiver must be able to mark a packet as it passes, and be able to recognise the destination address. The transmitter must recognise when the packet returns and mark it empty. If a receiver ignores a packet, perhaps because the station is switched off, then the transmitter will note the lack of a mark when the packet returns. The receiver has a select register which enables it to accept data from one selected station, no station or all stations. Packets are marked rejected if the selection register inhibits their acceptance. They are marked busy if the destination is busy, accepted if the destination copies their data, and ignored otherwise.

Packet structure

Each packet has a front marker digit so that it is distinguished from gap digits. It needs one bit to indicate full or empty. It needs a source address and a destination address, and also space for 16 bits of data. It needs space for response bits to indicate accepted etc. and finally it is useful to have one extra bit which is changed as the packet passes the monitor station, so that the packet can be deleted if it passes the monitor station again without the transmitter having picked it up. This facilitates the monitor station in emptying packets which errors have caused to appear full and which would otherwise circulate for ever.

The source and destination address are placed in front of the data so that the receiver has ample decoding time. The response bits are at the end of the packet to allow the maximum time for the receiver logic to decide whether to accept or reject the data. The monitor passed bit is put early in the packet as this enables a packet to be converted from a full packet to an empty packet without requiring any long shift register in series with the ring at the monitor station. The full or empty bit has to be first to enable filling to be started without delay; otherwise a delay is needed in series with the ring. Thus the packet structure chosen is as follows:

Bit	Purpose
1	start of packet; used for packet framing
2	full/ empty
3	monitor passed bit
4-11	destination address
12-19	source address
20-35	data
36-37	response bits used by the receiver to indicate rejected,busy, accepted, or ignored.
38	parity bit, used for checking ring links. (see below)

The first bit of a packet is used to facilitate framing. The packets are grouped into a train of packets terminated by a gap containing zeros. The number of packets circulating is set by switches in the monitor station, which also detects, counts, and corrects framing errors and gap errors.

Each station has a plug giving the station number and number of packets in the ring.

Traffic considerations

A transmitter can repeatedly send data to a receiver which ignores it. The fastest rate is attained with an immediate retry i.e. after a busy response which will take a ring transmission delay which of course depends on the current traffic. If we assume there are m active stations on the ring endeavouring to transmit as fast as possible, and then if a station endeavours to transmit a packet by ignoring the busy and continually sending the packet until it is accepted, then it will reduce the useful traffic by 1/m. If we consider that some devices may endeavour to drive teletypes in this way we can see an appreciable proportion of the total traffic may be useless. The m stations active may of course represent perhaps 10m stations each transmitting at its own full rate but only 1/10 of that of the ring. Thus a few "bad" stations may well impede the ring with useless traffic.

The station logic is designed to mitigate this problem by delaying the response if it is busy, rejected, or lost, to about 2 ring delays i.e. the time it takes for a packet to go around the ring, and to about 16 ring delays for the second and further tries. Thus, the jamming capacity of a badly designed access logic or computer program is reduced by a factor of about 16. The ring delay is arranged to be traffic sensitive so that its minimum value is one ring delay and its value is inversely proportional to the unused ring capacity. The round robin ensures there is always some unused capacity.

This delay will not influence the useful traffic very much as the acceptance is not delayed at the station.

It should be noted that with immediate retry, the traffic depends on the number of transmitters and not their speed, so that capacity is drastically reduced by a large number of slow stations. In practice we

use buffering concentrators for teletypes and VDU's.

Ring capacity

The data in each packet is 16 bits out of a total of 38. If we take the gap digits into consideration this implies that the bandwidth of the ring is about 4 megabits/sec, shared between all the users at any one time. There is an additional restriction that the maximum rate at which one station can send data is one packet every ring delay plus two packets. Thus for an 80 bit ring with 2 packets circulating the maximum transmission rate from one station is one megabit/sec. Of course the superposed protocols will reduce these figures.

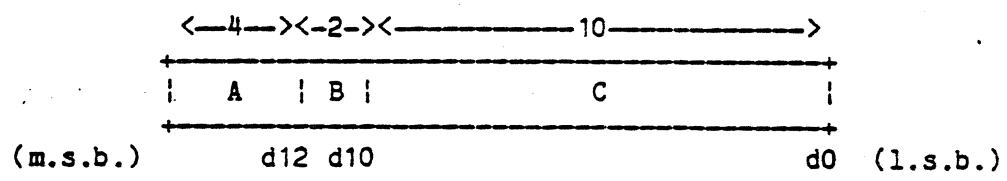
For a ring with n packets and m very active transmitters, each transmitting as fast as possible and all receivers accepting every packet without delay, the point to point bandwidth from one of the transmitters is

4/(n+m) megabits a second unless m=1 when it is 4/(n+2).

Thus if n=2 the number of transmitters needed to reduce their bandwidth to one half is seven. The formula seems to depend on total traffic and not the number of stations, so that the above ring with n=2 would sustain 14 transmitters at 1/4 megabit.

Basic ring transport protocol

A ring basic block commences with a header packet of the form:



Field A is the binary pattern 1001.
Field B is the type of the block:

=0: long block with checksum
=1: long block with checksum zero
=2: single packet carrying data C
=3: reserved for future use

A long block consists of:

- header packet as described above
- route packet
- C+1 data packets
- checksum packet

A route packet consists of a 'port number' in the bottom 12 bits, the basic block being notionally directed to that port at the destination

station. The remaining 4 bits are reserved and should be kept zero.

The C+1 data packets conform to the protocol (e.g. byte stream protocol) that is currently agreed to be in use at the port identified in the route packet.

The checksum packet for type 0 blocks consists of a 16-bit end-around-carry checksum over the entire block from the header packet up to and including the last data packet. In type 1 blocks, the notional checksum is sent as zero, and checked to be zero.

The intended method of operation for reception is as follows:

1. While a station is totally unable to receive anything, it keeps its select register zero.

2. When a station is potentially capable of receiving input, it sets its select register to 255.

3. It then listens for a valid header packet, ignoring anything which is not a valid header.

4. When a valid header has been found, if the station wishes to receive from the station from which the header came, then the receiving station sets its select register to that source, thus rejecting input from any other source.

5. The receiving station must operate either a per-block timeout or a per-packet timeout (or both) in order to recover from a block sent being shorter than the header packet suggested. The timeouts commence with the reception of the header packet. If the timeouts expire at any time henceforth, the input thus far accumulated is ignored, and the station is reset to state 2 above, ignoring the incoming block.

6. The next packet after the header is the route packet. If interpretation of this packet leads the receiver to believe that it cannot receive the remainder of the block (e.g. specified port not active) then it may reset itself to state 2 above, ignoring the incoming block.

7. On reaching the end of the block, the checksum packet is received and checked. If, for type 1 blocks, the checksum is incorrect, or for type 2 blocks, it is non-zero, then the entire block may be ignored as if it had never been received.

8. After reception of a block, the selection register may be restored to 255 if more input is possible, otherwise zero.

9. As an alternative to resetting immediately to state 2 if a partially received block is to be rejected, the selection register may be set

to zero for a short time in an attempt to cause the transmitter to stop sending. The selection may either be for a fixed time, or until the station hardware indicates that a packet has been rejected 'unselected'. If the latter strategy is used, a timeout would also be required.

For transmission:

1. When transmitting the first packet (header) of a block, due allowance must be made for the possibility of the receiving station being busy or unselected owing to it being in the process of receiving a block from another source. Attempts to transmit the header should be maintained for at least as long as the longest possible block can take at that reception station. Any other ring error can be regarded as fatal.

2. Having successfully transmitted the first packet (header), allowance may have to be made for certain reception stations to perform certain set up operations for the block, during which time the station will reject as "busy".

3. After that, the number of busy rejects that may be expected per packet should be very low, as the receiver is supposed to be concentrating on one source only. Any other ring error (e.g. unselected) is fatal. It will be necessary for a transmitting stations to have a timeout or repeat count on a per-packet or a per-block basis, in order to recover from a reception station crashing in the middle of a block. A timeout is also necessary to recover from certain ring errors (such as power off) which result in a packet never returning to its sender.

Use of parity in localising faults

Each packet whether full or empty has a parity bit which is checked and recomputed by each active station. Additional logic has been incorporated so that each such error detection will cause that station to transmit a fault packet to station zero. Thus a fault which causes a parity error can be localised to a single link. The fault packet is zero apart from the source, full bit, and possibly the parity bit. As the monitor passed digit is zero the packet will be emptied at the monitor station, so that it does not return to the transmitting station, and the transmission is simple.

The source of the first such fault packet will be displayed in the monitor station so that occasional faults can readily be localised and repaired. If a standard station with number zero is located immediately before the monitor station, then this station can log all such errors.

The station is arranged so that a continuous sequence of zeros possibly caused by a ring break, will send a continual stream of fault

messages, at about one every 292 bits. Thus ring breaks can be located in a similar way.

Errors

Currently we are experiencing about ten errors a day with thirty seven repeaters and twenty stations, so that the raw error rate is about one in 100 gigabits. The number of errors is kept small by adequate maintenance. However it is desirable to be aware of their effect.

Framing errors.

The loss of the first bit of a packet or the change of a gap digit is called a framing error. It causes stations after the error and before the monitor station to become unsynchronised and usually causes consequential errors. The monitor station will re-enter start mode until framing errors cease so that the ring is rapidly resynchronised and synchronised stations are inhibited from using the ring until that is complete.

Address errors

The transmitter detects the faulty packet when it returns. The wrong receiver may copy the data, but as our addresses are currently restricted to a 4 out of 8 code, this is unlikely and requires multiple errors of opposite type.

Response errors

A false response is returned whose effect depends on the protocols in use. For example if a busy is changed to accepted, then a packet may be lost and this will be detected in the sum check if one is used.

Data errors

The transmitter will detect the faulty returned packet. The destination will only detect the fault if a check such as a sum check is used.

Most of the above errors will cause a parity fault to be detected at the next station around the ring and a fault message sent to the logger and monitor station so that the fault is not ignored by the maintenance staff.

The monitor station helps in limiting the propagation of errors and keeping a count of the errors it detects. The monitor station forces the leader bits into the ring at each circulation and counts the number of leader bits that disappear and converts any full packet which passes for the second time into an empty packet. A count is kept of those packets which passed for the second time. Logic is included in the monitor station so that faults can be analysed for maintenance purposes.

Ring components

The design has split the ring logic into a number of separate parts; at each station there is a repeater which is powered from the ring and modulates and demodulates the data around the ring. It sends data and clock to the station logic. It is isolated when the station is turned off so that the integrity of the ring depends only on the switched on station units in addition to all the repeaters. The station unit connected to a repeater modulates and demodulates the packet into a parallel interface for the access logic which is designed for the particular device connected. It is possible to use different schemes for modulation and demodulation merely by changing the type of repeater for a particular section.

We also require one monitor station and one or more power supplies to supply 50 volts DC for the repeaters of the ring.

Bus connections to the access logic

The station is designed as a duplex station. That is, the receive and the transmit parts are independent and may be used simultaneously and independently.

The transmit part has a 16 bit two way three state bus which is controlled entirely from the access logic by gating signals. These allow data and destination to be set, and replies sensed. As there are independent gates which control each 8 bit half of the bus, these may be commoned to reduce the number of bus wires to eight.

The receive part is similar, but also contains a selection register that can be set so that packets received can be limited to those received from one source, no source, or all sources.

The two buses can be commoned together, but again subject to the constraint that no two bus driving gates should be active simultaneously.

Station to access logic interface

There are two 16 digit 3 state buses. The gates onto the buses are either controlled directly in the access logic or indirectly by gating signals which are are sent along the interface cables. The gate signals are uncoded so that each signal directly controls one gate in the station. This allows the buses to be grouped so that fewer wires are needed at the expense of more gating steps.

The control signals are all negative going signals such that the active state is the zero state, this convention has the advantage that an absent cable will not generate spurious signals.

The transmit bus is named TB0-15 and the receive bus RB0-15 numbered such that signal 0 is the least significant and corresponds to the earliest serial signal along the ring.

Control signals Action

CS1 Sets the 8 bit source selector from RB0-7
CS2 Gates received data 0-7 to RB0-7
CS3 Gates received data 8-15 to RB8-15
CS4 Gates received source address to RB0-7
CS5 Gates source selector register to RB0-7
CS6 Gates read status to RB6,7 . RB6=1 if station has rejected a packet from an unselected source since the selector register was last set.
CS7 Echo response to access logic. The OR of CS1-6,16. Used for handshake gate timing control
CS8 Set data TB0-7 in transmission packet
CS9 Set data TB8-15 in transmission packet
CS10 Set destination address in packet from TB0-7
CS11 Gate transmit status to TB0-4
TB0=0 if status is busy
TB1=0 if status is unselected
TB2=0 if status is accepted
TB3=0 if status is ignored
TB4=0 if status is erroneous packet returned
These responses are not valid until CS14 has gone down.
CS12 Echo response to access logic. The OR of CS8-11,13. Used for handshake gate timing control.
CS13 Transmit command. Will transmit after setting source and control bits.
CS14 Transmit reply. This signal will go to one after the end of CS13. It becomes zero again when the transmitted packet returns and the response bits are again valid.
CS15 Received packet available signal. This signal will go to one at the end of CS16 and become zero again when the receiver register has again been loaded.
CS16 Receive command to the station. The current packet is discarded.
CS17 This signal indicates the gap preceeded by an empty packet so that on the average it indicates the average access delay to the ring. It is sometimes used to generate a "ring working" signal.
DISABL This disables the powered station and can be used to ensure safe switching on.
RELAY+
RELAY- The station routes these to the repeater.

Note that if gates CS8-10,13 are given while the previous transmission is not complete they are ignored.

Station on off sequencing

In order to preserve the integrity of the ring there should be no random signals while the station is turned on. The following sequence is suggested.

1 The station is turned on while disabled.
2 The repeater relay is closed when station has full power applied to it for some milliseconds.
3 The station is enabled.

The station turn off is the reverse of this namely:

1 the station is disabled.
2 The repeater relay is opened
3 The station is turned off.

The above sequencing is good enough for all practical purposes, but it can be made perfect by arranging that the enable or disable signals change only at the negative edge of CS17.

Station to repeater interface

The signals are listed below.

-DS serial Data to Station
DR serial Data to Repeater
-CS -Clock to Station. Timing on -ve edge.
GR Gate to Repeater to cause the station to be bypassed.
FC A control to the phase locked loop in the repeater used only at the monitor station.
RELAY+
RELAY- These control the repeater relay which only allows GR to be active when closed.

Outline of station design

The station comprises the followig parts:

1) Framing logic. This locates the passing packets and gaps.
2) Receiver logic. This recognises packets sent to it marks them accepted, busy, etc. and copies the packet and signals the access logic. It also has the selection register which determines which packets are rejected.
3) Transmission logic. This accepts packets from the access logic, and awaits a passing empty packet, fills it, awaits its return, compares it with the original packet, and marks the packet empty and will signal the access logic with the response.
4) Parity logic. This generates and checks the parity of packets. Each

fault detected will cause the transmission of a fault packet to destination 0 when the first empty packet arrives.

5) Delay logic. This delays responses other than accepted to the access logic so that unintelligent devices cannot swamp the ring by continual retries.

The design uses 67 TTL integrated circuits, the most complicated of which is an eight bit shift register. A special plug of the same size gives the station address and number of packets in the ring.

The repeater

This takes phase encoded signals from two pairs of twisted wires and demodulates them into a clock and data signal which is supplied to the station logic. It accepts gate and data signals from the station (synchronised to the clock) and combines these with the incoming data and then modulates these onto the output twisted pairs.

It also contains a DC transformer which takes power supplied along the twisted pairs and generates its own supplies for its circuits. Thus it is independent of local power. The gate signal from the station (which is the only destructive signal) is protected by an open reed relay when the station power is off.

A voltage controlled oscillator in a phase locked loop has better characteristics than a clocking system based on fixed delays. Thus the repeater uses a phase locked loop and the present ring works over a frequency range from 8 to 12 megabits/sec. The phase locked loop is centered in frequency when no signals are received, thus aiding synchronisation and location of ring breaks.

The repeater is isolated from the ring by four small ferrite ring transformers so that so that common mode induced voltages have little effect.

The repeater uses about ten TTL integrated circuits, together with some other components for the phase locked loop and power conversion oscillator.

Ring Power Supplies

A number of power supplies, currently four, inject about 50 volts DC into the ring to supply the repeaters. They are designed so that they co-operate in supplying power, are short circuit proof and cause the monitor station to go into the start mode after a short circuit, thus allowing maintenance restarts from around the ring.

Monitor Station 2

This is intended to run a ring rather than be used when commissioning one and so is simpler than the first version, which was used for commissioning.

There are two modes of operation, starting and running. The start mode is forced as power is switched on and after a framing error. It sets up the packet structure and counts the gap digits, so that the frame can be exactly maintained. The starting frame consists of full packets otherwise zero, so that synchronised stations do not interfere with with these starting packets. The number of packets and a ring extension of up to 40 bits are determined by a wired plug inserted in the monitor station.

The transition to running mode will clear the error tags and indicators. If errors occur tags are set to indicate type of error and a counter stepped to tally the errors. The counter has 8 bits and so works modulo 256. All simple errors result in a fault message packet being sent to the logging station whose address is zero and ring position just before the monitor station. The packet has source zero, destination zero, and the error tags and error count as content. The tags are then cleared but the count is left so that the logger will detect omitted fault packets and the display at the monitor station can indicate the fault count.

One extra feature is that the source of the first packet with the monitor passed digit set is held in a register if the transmission of fault packets to the logger is disabled by a switch. The register can be displayed so that faults can be localised in small rings without a logging station.

The monitor passed bit is used by the monitor station for a further two purposes. The first is to empty packets which have become full by an error or because a transmitter was turned off in the middle of a packet transit. This can be detected because a full packet should never pass the monitor station twice. The monitor passed bit is set to one at transmission time and set to zero at the monitor station so that a full packet should not have it zero as the packet enters the monitor station.

In a similar way an error could cause a full packet to become empty and there is the possibility of an infinite number of consequential errors occurring as pairs of stations fill and empty this packet. The chain of events is broken at the monitor station by forcing an empty packet with the monitor bit set to a one to be full.

Start mode is re-entered after a framing fault. For one ring cycle all packets are set to full at the monitor station, thus allowing packets already transmitted to complete their journey, but inhibiting new transmissions. Two or three cycles of start mode follow, the error tags are cleared and pseudo fault messages sent from the monitor station to log the event. The ring is reopened for use, having been synchronised in a rapid and not very destructive way. The gap count will have been reset. Thus a fault which changes the number of digits in the ring will have been countered.

If the above happens 256 times with no manual clear then the power on sequence will be re-entered. This is destructive but will reset the frequency to 10 megaherz, so that violent power dips occurring while the monitor station is unattended are dealt with. A tag and count indicates

the number of full resets (modulo 4) since the last manual clear.

The monitor station comprises about 80 TTL integrated circuits, together with indicators, power supplies, switches etc.

RING SYSTEM PROTOCOLS

TRANSMISSION SPEED AND CONTROL

To ensure adequate reliability in the transfer of data, some sort of acknowledgement or 'handshaking' facility is desirable. A simple (and tryed and tested) method is for the receiving station to 'mark' a packet received, and the packet is deleted by the transmitting station by marking the packet 'empty'. A station will not transmit the next packet until the first has been acknowledged in this way. This method has a number of advantages, in particular, it is impossible for 'hogging' (a saturation of the ring by a station transmitting with a high data rate) to occur. However, this method has disadvantages in terms of speed; since each packet must completely traverse the ring, the ring capacity is halved; also, the maximum data rate is limited by the size of the ring, since a complete ring traversal time must occur between transmission of packets.

A second method is for a receiving station to 'empty' the packet when it arrives, but forces the transmitting station to wait a ring traversal time before transmitting the next packet. Thus, if a packet is not accepted by the receiving station, it will return to the transmitter, its arrival indicating a 'rejected' condition. This method avoids the first of the problems described above, since packets are usually deleted by the receiver. However, the maximum data rate is still limited by the ring traversal time.

For a large ring, say 20 km length, and assuming an average transmission speed of 150,000 km\second (half the speed of light), this limits the transmission rate from any one station to 7500 packets\second. This speed is more than adequate for terminals and I\O devices. (With a 16 bit data field, this corresponds to a data rate of 120 kbps.) For transfering files between machines, this rate is rather slow. For example, an 'average' file of 2K 32 bit words would take approxiamately 0.5 seconds to transfer (not including any delay due to disc latency, processor usage,etc.), while a large file (8K 32 bit words) would take 2.1 seconds. While this is not excessive by itself, it is of the same order as the time taken to copy such files on a single machine, so that a file transfer might take 4 seconds under good conditions. This is unlikely to be entirely acceptable to users, while this data rate is inadequate for direct store-to-store transfer operations (Paging) using a ring as the data link. (Consider a small paged machine using remote disc\Bubble\EBAM\large core store for currently unused pages.)

GROUPED PACKETS

An increase in the maximum data rate at which a station can transmit is desirable. This can be obtained by allowing a transmitting station to transmit more than one packet (a packet GROUP) before waiting for an acknowledgement (e.g. not receiving one of its own packets in a time greater than the ring transit time). A NAP (Network Access Processor) will have to contain buffering in order to absorb all the packets when they arrive (a fast buffer will be required). The buffer size of the receiver will have to be

communicated to the transmitting station during the 'establish communications' phase of operation. In the case of a fault in a receiving device or an error in the packets received, several different courses of action could be taken. For example, the receiving station could remove all except one of the packets sent; when this arrives at the transmitter, it would retransmit all of the last group of packets. Alternatively, the receiver would empty all the packets sent to it,unless there was an error condition. If less than the expected number of packets were received, the receiver would sent an 'error' packet to the transmitting station, asking for retransmission. In either case, if the same group of packets have to be transmitted more than, say, 3 times, then the receiving device is assumed to be inoperable and the transmission aborted.

From the above discussion, it is obvious that devices connected onto the ring can be divided into two classes, dependant on whether their natural speed is greater or less than the limiting speed determined by the ring transit time when sending one packet at a time. The 'slow' devices include almost all I\O devices and terminals while the other class includes processors and disc controllers. The essential difference between the groups is that a 'slow' device need sent 'establish communications' packets <u>once</u> only (when the terminal etc. is first switched on). Thereafter, any packet arriving with the correct source and destination address is accepted by the terminal, unless there is an error condition. By contrast, a 'fast' device must establish communications every time a group of packets is sent.

Information exchanged on switch-on, for a 'slow' device:

(1) Device location, i.e. its ring address.

(2) Device type, character code, etc.

(3) Device speed, (assumes buffer size of 1).

- all sent from terminal to processor in one packet.

(4) Acknowledgement (prompt).

- from processor to terminal.

All this action occurs before any logging-in procedure.

For a 'fast' device, a more complex protocol will be required. However, it will be necessary for each participant to sent at least the following:

(1) Device location (address).

(2) Device type,

(3) Buffer size,

(4) Size of message.

- approx. two packets.

SYSTEM SOFTWARE

The distinction between 'slow' and 'fast' devices described above seems to correspond reasonably well to the GRF\CJT concepts of CHARACTER and COMMUNICATIONS SOCKETs. The Network Access Processor (NAP) described previously corresponds to the hardware of the socket, while the software would be resident in a processor on the ring. The ring hardware simplifies the network software in that no 'routing' information is required other than the destination

identifier. For character input and output devices, the ring hardware conveys characters to/from the terminal and processor, where the communications controller software (I/O manager) interprets and buffers the data. (Packet data fields can contain any bit pattern.) Provided that the ring is not overloaded, the controlling process will see a dedicated link between it and the terminal. All connection and disconnection, and error reporting would be achived by 'control' (rather than 'data') packets containing various status bits (c.f. control register V.IO.CONTROL), as well as device type, speed, etc.

Similarly, for communication input and output devices, a message would be transmitted as a number of packet groups, preceded by packets to open the transfer and concluded by packets to close the transfer. Again, control packets would indicate the status and type of each machine.

SUB-RING SYSTEMS

As an alternative to a single, uniform ring structure, a more complex architecture using multiple 'sub-rings' connected together via a 'super-ring' could be used (see Fig. 3.). The advantage of this is three-fold; it provides a partial fail-soft capability, it reduces ring loading due to packets going 'the wrong way' round the ring, and allows a higher data rate (in a sub-ring) while still acknowledging every packet before another is sent.

The sub-rings would correspond to individual buildings or other concentrations of computer equipment. Each sub-ring is unlikely to exceed 1km in length; at this length, transmission rates can be up to 2.4 Mbps per station while still acknowledging every packet. Most high-speed file transfers will take place within a sub-ring; a file transfer outside the local sub-ring will take longer, but this may be acceptable to users in view of the greater distances involved.

The sub-rings are connected to the main ring via an Inter-Ring Controller (IRC). The IRC is connected to the two rings via normal NODES so that a failure in an IRC will isolate the two rings but not prevent communication in the two rings seperately. The IRC is a two-way link, each way would inspect the destination address of the packets as they go by. If the IRC determines that the packet is on the wrong ring, it will remove it from one ring and store it until an empty packet is found on the other ring. Note that a considerable

internal buffer will be required in each direction.

A suitable identification would be to use the top 4 bits (in a 12 bit address) of the source and destination identifiers to indicate the sub-ring (address 0 indicates the main ring). Note that no distinction is made to the user between NAPs on the main ring and on any sub-ring, except that transfers are faster in the local sub-ring, and that it determines the fail-soft performance.

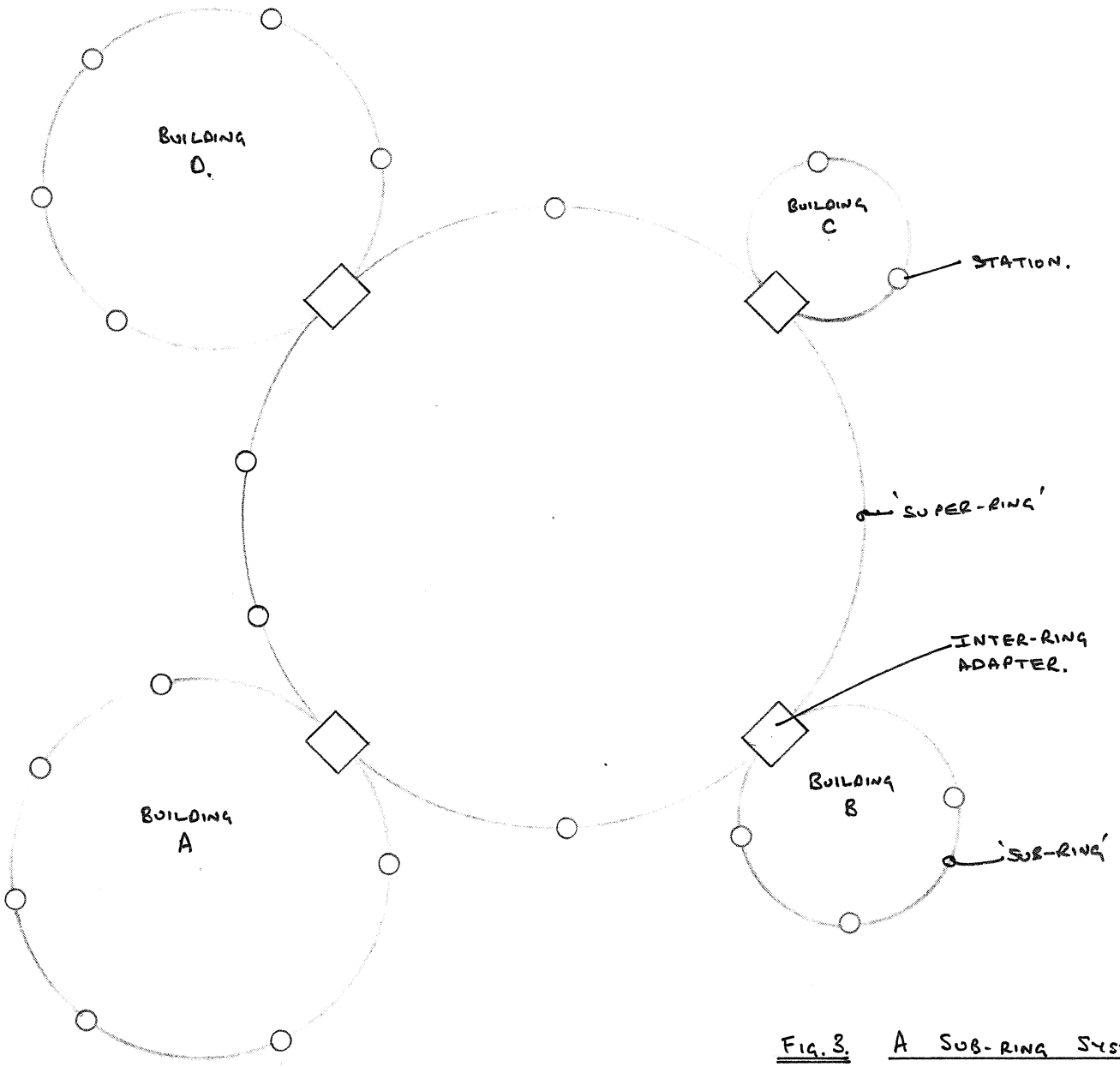


FIG. 3. A SUB-RING SYSTEM.

MU6 COMMUNICATIONS

INTER-COMPUTER COMMUNICATIONS

This may be divided into a number of areas, as follows:

1. Range of a few tens of meters:

1.1 Multiprocessors; e.g. MU6.C machines, obviously need a very high data rate, therefore a serial system is not suitable.

1.2 Multiple Processors; e.g. MU6.G - MU6.C communications, a high data rate is required, but it is possible to use a high-speed serial system.

1.3 Terminals; e.g. Terminal to Multiplexor\Common access point.

Relatively low speeds are required here, so that conventional serial techniques can be used.

2. Range of a few kilometers:

2.1 Local Networks within a single building;

2.2 Local Networks over a wider area, e.g. University campus, factory site, etc.

Both these are obvious candidates for serial links.

3. Range of very many kilometers.

3.1 National networks

3.2 International networks.

Both these are outside the scope of this discussion.

For the MU6 project, the immediate areas of interest are 1.2 and 2.1 above. Undoubtedly, further work will be done on a wider range of interests.

A RING ARCHITECTURE

It is suggested that a uniform ring-type architecture with a data rate of 20 Mbps could be used for 1.2, 2.1 and 2.2 above. This has a number of advantages over alternative ad-hoc systems.

1) The system would be simply expandable during development, and down-time due to mechanical reconfiguration would be small.

2) A hierachical structure could be enforced on the users (not the system) by limiting the data rate of each device on the ring according to its notional priority. (See next section.)

3) Reliability in ring systems is a problem, but for the proposed system:

a)NODES on the network could be seperate from the NETWORK ACCESS PROCESSOR (NAP).

b) NODES would be simple repeaters with a small amount of gating to allow signal entry/exit. Simplicity ensures that the NODES have high reliability. The NODES could be powered (local

regulation) from two or more central points via power cable running with the communication link (co-ax, fibre-optic cable, etc.).

c)NAP's could be disconnected (or fail) without damaging the remainder of the network

4) Simplicity, uniformity and an economic advantage would ensue by designing the NAP's in three parts:

a) A RING ACCESS CONTROLLER (RAC), all of which would be identical except for the address of the NAP and it/s allowed data rate. This may be switched or programmed into the RAC.

b) An interface (probably 8 or 16 bit parallel, with a number of control lines).

c)A USER INTERFACE CONTROLLER (UIC) ; several different types would be required. Obviously, the UIC for a few terminals would be vastly different from one designed for an MU6.G.

5) A high data rate ring is possible using co-ax or fibre-optics as the transmission medium, and using the ISO High-level Data Link Control (HDLC) procedure.

6) A NAP to interface to X25 protocol links would be feasible, for national/international traffic.

PRIORITY INDEX

The priority of any unit connected in the ring system is determined by its maximum data rate. Thus, a low-priority device like a VDU would automatically use far less resources than an MU6.G at 100+ kbps. A terminal connected via a multiplexor would have the same effective priority as a single terminal of the same speed.

A Priority Index (P.I.) can be given by:

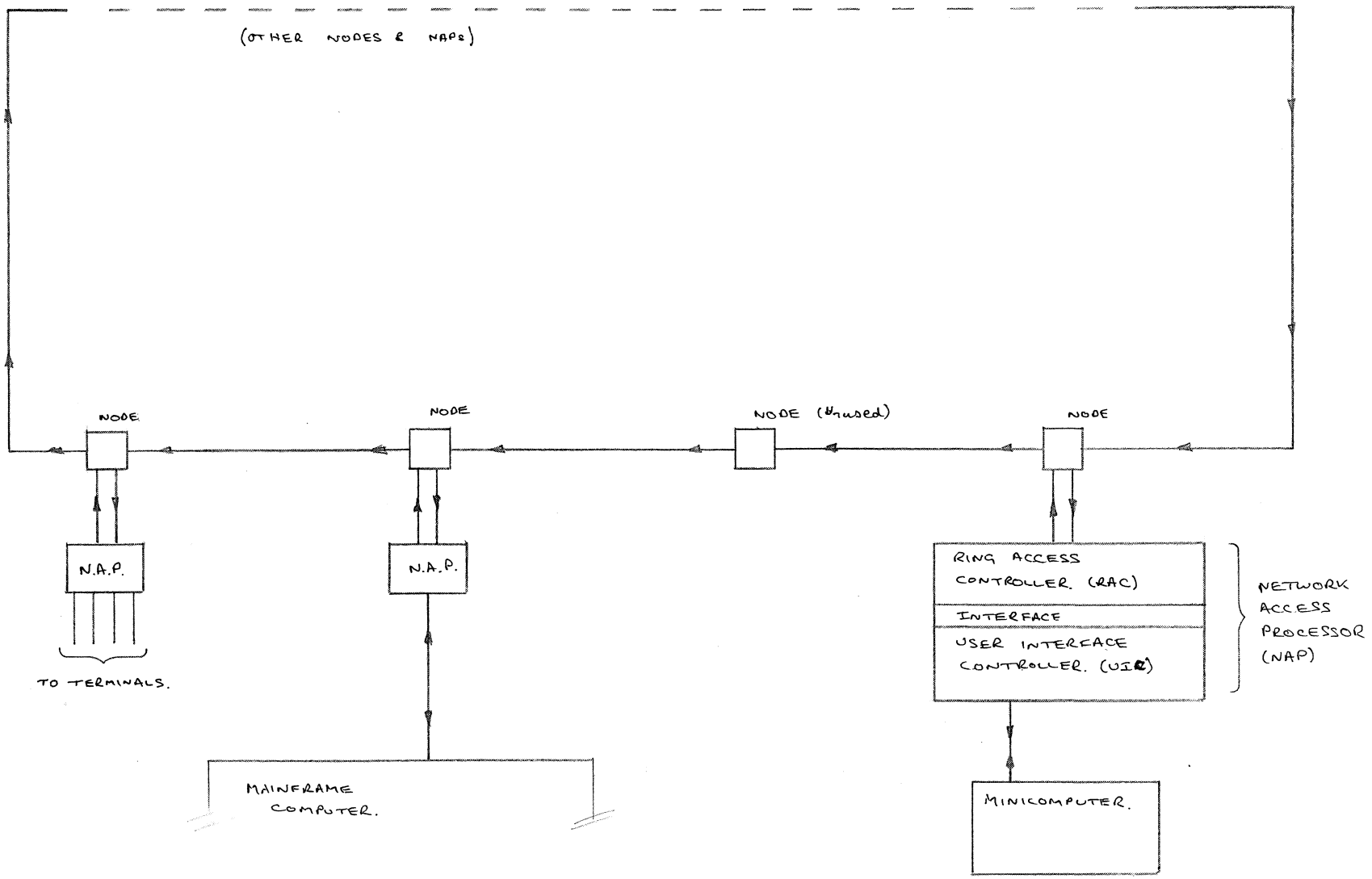
P.I. = 100 / (Max. Data Rate in kbps.) .....(1)

The P.I. will vary from 1 (or less) to about 1000 (for 110 baud TTY's).

HARDWARE EXPERIMENT

To gain some experience with ring systems, it is proposed to construct a simple system, operating at approx.10 Mbps. Such a low-speed link could be easily constructed from TTL logic, using co-ax or twisted-pair connections. As an independent development, a third year student working for Mr. A. Rawsthorne is already designing suitable stations for use in the microprocessor laboratory to connect together a number of 380Z microcomputers and slow and fast peripherals.

FIG. 1. SUGGESTED COMMUNICATIONS SYSTEM.



FURTHER NOTES ON RING COMMUNICATION SYSTEMS

NODES AND NAPs

As previously stated, the physical connection points on the ring (NODES) are seperate from the device control functions (NAPs). Ideally, the NODES should be as simple as possible since they must be highly reliable; all must function correctly for the ring to work at all. Also, more NODES should be provided than are actually used at any one time. This allows easy relocation of NAPs and associated terminal(s) if necessary without disrupting other communication, since it should be possible to remove or insert a NAP into a NODE without ill effects.

Internally, a NODE would consist of three parts (see Fig. 1.). These would be:

(a) A recieving unit; and

(b) A transmitting unit.

for use with either coaxial or fibre-optic cable.

(c) A simple gate (See later).

Not shown in Fig. 1. are the power supply arrangements. It is suggested that unregulated power can be supplied to the NODES via cables running with the signal connections. Several power supplies

at central points should be provided , each of which should be capable of powering the entire ring. Each NODE would have an internal IC regulator to supply its own power; these are cheap and quite reliable.

Only three connections are required between a NODE and its corresponding NAP (Fig. 1. again). These are DATA OUT, DATA IN and CONTROL. DATA OUT delivers to the NAP whatever is currently on the ring at that point. When control is inactive (this state must be forced if the NAP is disconnected) , data from the reciever is passed directly to the transmitter; i.e. data passes straight through without alteration. When CONTROL is active, the data present on the DATA IN line is fed into the ring, overwriting what was on the ring previously. In this way, a packet can be built up in the NAP and, when a 'gap' is seen in the data passing through, the packet can be released onto the ring.

COMMUNICATION BETWEEN NAPs

To perform the required communication, a NAP must be able to:

(a) put a packet into an 'empty' space on the ring, and

(b) recognise and 'remove' a packet addressed to it, or one of the devices attached to it.

Therefore, a packet must contain the following:

(1) start and stop bits,

(2) source device identifier,

(3) destination device identifier,

(4) data field,

(5) some control bits - the exact nature of these is highly dependent on the implementation of the ring.

The choice of data field size is a difficult one; too small and the overheads due to the packet size are excessive, too large and the ring will contain many 'half-full' packets. (Consider the case of packets containing an 80 byte data field being sent from a terminal every time 'newline' is pressed.) One possibility (assuming 'asynchronous' communication) is to have a variable length data field, with an extra control field specifying the size of the data field.

Another question to be answered is the size of the source\ destination identifiers, and whether they refer to NAPs or to the devices attached to them. For example, does each of 16 terminals

attached to a NAP have a unique source\destination identifier, or will they have to be identified using part of the data field as a selector. The first method is perhaps more elegant in principle, but it complicates the interface between the Ring Access Controller (RAC) and User Interface Controller (UIC). Also, the additional facilities for partial decoding would complicate the RAC. The second method makes the software in 'active' devices more complex (a computer would have to 'know' that it was talking to a terminal via a multiplexor rather than directly), but the hardware is somewhat simpler.

Obviously, each device\NAP must be uniquely identifiable (c.f. a conventional parallel bus system - each device\memory location must have a unique address). This must be ensured by setting the RACs correctly during commissioning. Also, the rate at which a computer can sent to any particular device must be controlled; sending 2400 bps to a teletype will be disasterous. Therefore, each machine must 'know' the data rate at which it sends and expects to recieve.

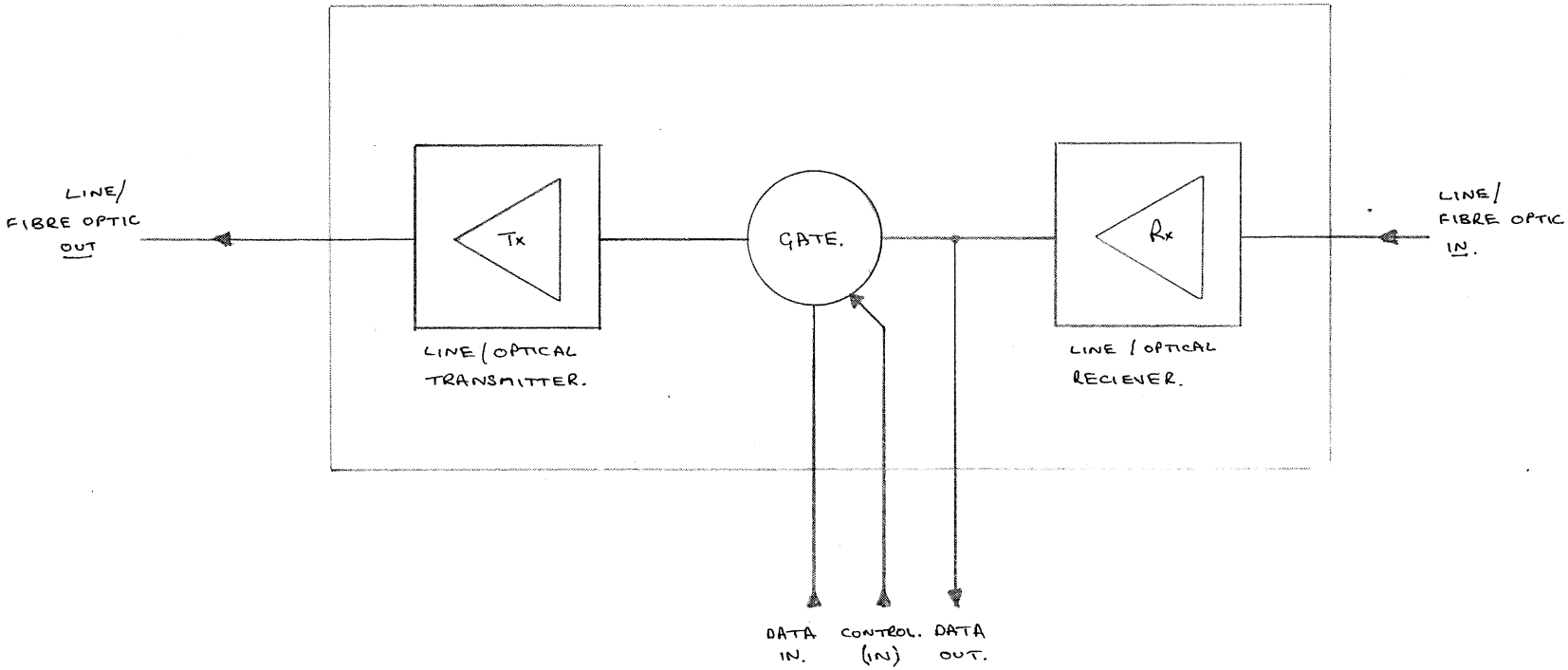


Fig. 1. SCHEMATIC DIAGRAM OF A SINGLE NODE.

Interested ?

CCR

CAMBRIDGE RING BIBLIOGRAPHY

(12)

W.P. Sharpe

Rutherford & Appleton Laboratories

3rd Sep 1981

Most of the documents listed here are available to DCS related projects from RAL. Copies of INDRA notes should be requested directly from UCL. This bibliography makes no claims to be exhaustive. Please send any items for inclusion to:

W.P. Sharpe
Computing Division
Rutherford & Appleton Laboratories
Chiton
Didcot
OX11 0QX

General

1. "Cambridge Digital Communication Ring", Computer Laboratory Cambridge (19 Oct 1979).
   An overview of basic ring hardware. prepared for distribution to intending users

2. R.P.A. Collinson, "The Ring and Small Unices", University of Kent (November 1980).
   Possible ways of implementing BSP/BBP on small Unix systems and some of the problems.

3. J. Dion, "File Server External Specifications. Version 6", Cambridge Computer Laboratory (18th March 1980).
   At Cambridge one machine, an LSI4 with 150 MB of disk storage, centralises the control of bulk storage for other machines. This document describes the services provided and the interfaces to them.

4. D. A. Duce, "Report of the UCSD Pascal SIG at the University of Keele, 4 Mar 1981", DCS 378, Rutherford & Appleton Laboratories.
   The following presentations are reported: P. Brereton (Keele) - preliminary performance measures of a BBP based message system; S. Currie (Edinburgh) - Brief description of the proposed Edinburgh network development to utilise the ring J.W. Hughes (UMIST) - description of the Pascal Plus virtual machine to be implemented on CYBA-M; T.E. Schutt (Kent) - report on the development of a Pascal compiler server on a Western Digital Microengine; W.P. Sharpe (RAL) - obstacles to software portablity; C. Silk (Keele) - Summary of a development proposal to use a microprocessor to interface the ring to well-known computer interfaces (e.g HDLC).

5. J.J. Gibbons, "The Cambridge Nameserver", Cambridge Computer Laboratory (11th January 1981).
   A specification and description of the nameserver in use at the time of writing. The note is not concerned with wider design issues.

6. C.J. Kennington, "The UCL Ring-based Terminal Multiplexer. Issue 1", INDRA 1016, University College, London (Nov 1980).
   Like the UCL Ring interface this is based on the UMC-Z80. This note is a description of the hardware with some notes about programming it.

7. C.J. Kennington, "Programming the 'SW' Z80/Ring Interface. Version 2", INDRA 1017, University College, London (5th Feb 1981).
A working paper describing the UCL general purpose ring interface (designed by Steve Wilbur), based on the UMC-Z80, with detailed programming instructions. This interface is very similar to the one now marketed by Logica VTS.

8. R.M. Needham, "Systems Aspects of the Cambridge Ring", Proceedings of the Seventh Symposium on Operating Systems Principles, SIGOPS, ACM, Pacific Grove, California (10-12 December 1979).
Describes some of the principles on which the Cambridge hardware and protocols are based.

9. E.B. Spratt, "Operational Experiences with a Cambridge Ring Local Area Network in a University Environment", UKC CL Report 3, University of Kent (Oct 1980).

10. E.B. Spratt, "Selecting and Installing a Campus Network", IUCC Management Conference Proceedings (April 1980).

11. E.B. Spratt, "Developments of the Cambridge Ring at the University of Kent", Local Networks & Distributed Office Systems, ONLINE, London (May 1981).

Protocols

1. I.N. Dallas, "Transport Service Byte Stream Protocol (Revision 3)", UKC Comp. Lab. Report No. 1, University of Kent (17 Aug 81).
This is an agreed standard that enhances the Cambridge BSP to provide the Yellow Book Transport Service. It also provides the current definitive specification of BSP and BBP. An appendix gives a guide to BBP as implemented at Kent and ERCC.

2. D.A. Duce, Report on the Cambridge Ring SIG on Internetworking at Darwin College, Cambridge, 27 Apr 1981.
I. Lesley (Cambridge) :- "The Cambridge Gateway". Cambridge are gaining experience with a high performance gateway between two service rings; the level at which a gateway should operate and addressing problems were considered. P. Corcoran (Strathclyde):- "A Gateway between a Cambridge Ring and an Ethernet-like Network". J. Butler (ERCC):- "The Edinburgh PSS Gateway". A. Hinchley (ULCC):- "Case for Transport Service Gateways".

3. D.A. Duce, "Report of the Cambridge Ring Protocols SIG Meeting at University College, London, 12 Feb 1981", DCS 381, Rutherford & Appleton Laboratories.
A. Herbert (Cambridge) - background to the design of the Cambridge protocols; Sweetman[Jan81]; Kennington - Why a New Protocol? Kennington[Jan81] (INDRA 1040); Lloyd[Jan81] (INDRA 1032); Hinchley[12Feb81]; This SIG was mainly concerned with the adequacy of BBP and BSP for use in an environment where there is heavy multiplexing of block and character traffic.

4. J.J. Gibbons, "SSP - a single-shot protocol for the ring", SRG Note, Computer Laboratory Cambridge (25 Sept 1980).
This is a widely used protocol for transactions which consist of a

single basic block 'request' to which the response is a single basic block 'reply' e.g. name-server enquiries.

5. J.J. Gibbons, "Gizmo - Outline Description", Computer Laboratory Cambridge (19 May 1981).
   Cambridge proposals for building a hardware interface to provide high performance byte stream protocol access to the ring.

6. A.J. Hinchley, "Interworking of new protocols on the ring - a summary", London Network Team (12 Feb 1981).
   Requirements that must be met by any new ring protocols.

7. M.A. Johnson, "Ring Byte Stream Protocol Specification", Computer Laboratory, Cambridge (April 1980).
   This is the original specification of BSP and BBP. It was originally produced for internal use at Cambridge and therefore is not a complete guide to use and recommended implementation; for further information see Dallas[17Aug81], Sharpe[Mar81].

8. C.J. Kennington, "Problems Inherent in Current Ring Protocols", INDRA 1040, University College London (January 1981).

9. J. Larmouth, "Low Level Protocols on The Cambridge Ring", University of Salford (Janurary 1981).

10. P. Lloyd, "A low Level Multiplexing Protocol for the Cambridge Ring", INDRA 1032, University College London (Jan 1981).
    A proposal for a protocol that exploits the ability of the ring to muliplex byte stream and block traffic at the mini-packet level.

11. R.M. Needham and M.A. Johnson, "Problems inherent in ring protocols - reply to C.J. Kennington", Computer Laboratory, Cambridge (February 1981).

12. W.P. Sharpe, "A Guide to BBP and BSP. Second Draft", DCS 380, Rutherford & Appleton Laboratories (Mar 1981).
    Supplementary information on the definition and implementation of the protocols.

13. D. Sweetman, "The Case for a Simple Unreliable Datagram Protocol", Logica VTS (Jan 1981).

14. S.R. Wilbur, "Low Level Protocols in the Cambridge Ring", INDRA 893, University College London (13th Mar 1980).
    Discusses the way in which protocols for the Ring differ from protocols, such as X25, designed for long haul networks.

Software

1. S.E. Binns, "Ring programs for the DEIMOS operating system", University of Kent (October 1980).

2. P. Brereton, "User Interface to the Transport Service on an Apple", DCP/WD/45, University of Keele (31 July 1980).

3. R.P.A. Collinson, "Unix Programmer's Manual pages for VAX ring software", University of Kent (1980).
Implementation of BSP, BBP, and Edinburgh ITP, RJE within the Unix kernal.

4. S. Cook, "Unix on the Cambridge Ring - Preliminary Specification of User Interface", CSL 262, Queen Mary College (July 1980).

5. T.E. Schutt, "Ring Handler [1.0] User Manual", University of Kent (Apr 1981).
A package written in UCSD Pascal (version independent) to run as a stand alone utility or part of the operating system providing a straightforward user interface to ring services: BBP, BSP, SSP, nameserver, file transfer, down line loading, line printer spooling. Written for the Western Digital Microengine, it has the hardware dependencies isolated in routines that may easily be modified for other hardware. Available under licence to DCS investigators from Rutherford.

6. W.P. Sharpe, "Unix Programmer's Manual pages for ring software", Rutherford & Appleton Laboratories (July 1981).
An implementation orientated towards small Unix systems - BBP in the kernal, process-per-stream BSP in user space.

Hardware

1. R.E. Carbonell, "LSI11 Ring Access Logic", INDRA 916, University College London (26 May 1980).

2. A.R. Cash, C. Maclean, and W.P. Sharpe, "Specification of RAL Cambridge Rings", DCS 283, Rutherford & Appleton Laboratories (3 Oct 1980).
Hardware specification of the Ring systems built at RAL for the DCS programme.

3. A. R. Cash and R. S. Milborrow, "IEEE 488 Bus Extension over the Cambridge Ring", Rutherford & Appleton Laboratories (June 1981).
RAL intend to build an IEEE 488 ring interface; this is a draft proposal that is out for comment.

4. J.P. Chevreau, "Description of the Z8000 to Ring Interface", INDRA 911, University College London (May 1980).

Modelling and Simulation

Includes performance studies.

The maximum achieved throughput for BBP is approx 600 kbits/sec; for BSP it is approx 60 kbits/sec. A great deal of attention is currently centred on discovering why there is this drop in performance between the levels. At Kent they have found that buffer manipulation and task scheduling are a major factor. Theoretical and experimental work in progress at Edinburgh will be looking at the effect of the BSP 'window of one' and examining a low-level multiplexing alternative. So far, the low level multiplexing protocol proposals (eg Lloyd[Feb81]) have not been able to demonstrate the advantage suggested by the work of Singleton &

Peake (Duce[Mar81] SIG report). In parallel with this there is interest in developing high performance transport service 'black boxes' (Gibbons[May81]) for use as front ends that may be an alternative means of raising performance.

1. G.S. Blair, "Comparison of Point-Point and Broadcasting Communication Links", University of Strathclyde (1980).
   Report of a one year project. A summary of this work was presented at the DCS Modelling and Simulation SIG.

2. P. Brereton, "Performance Figures for Message-Passing over a Cambridge Ring", University of Keele (April 1981).
   Figures obtained during the tuning stages of an implementation of BBP on LSI11/03s.

3. D.A. Duce, "Report of the Cambridge Ring Modelling and Simulation SIG, Darwin College Cambridge, 19 Mar 1981", DCS 395, Rutherford & Appleton Laboratories.
   P. Singleton & P. Peake :- simulation showing theoretical performance advantage when SSR is always kept open. E. Gelenbe (l'Universite de Paris-Sud) :- studies in progress on the charateristics of CSMA type networks. G. Blair (Strathclyde) :- simulations of Ethernet and Cambridge Ring (see Blair[80]). S. Temple (Cambridge) :- dependence of system bandwidth and point-point bandwidth on the slot count. Breakdown of observed ring traffic at Cambridge by various criteria. Bozyigit[Mar81]; Boyd & Hughes[Mar81]; Mayne[Mar81] (INDRA 1002); Ody[10Sep80].

4. P. Lloyd, "Ring Data-Capture Measurements", INDRA 934, University College London (25 July 1980).
   Measurements on data-capture style ring traffic between an LSI-11 and a Unix system using both interrupt-per-packet and 'burst mode' strategies at either end.

5. K. Lunn and K.H. Bennett, "Message Transport on the Cambridge Ring - A Simulation Study", Software - Practice and Experience, Vol 11, 711-716 (1981).
   Concludes that very high message (basic block) transmission rates are possible with use of the SSR and compares the performance of various queue handling strategies.

6. A.J. Mayne, "A Simple Model of Multi-Station Time-Outs on a Ring using the Basic Block Protocol and Byte Stream Protocol", INDRA 1002, University College London (March 1981).

7. N.J. Ody, "Logging of Ring Transactions at the basic block level", Computer Laboratory Cambridge (10 Sept 1980).
   Examines three approaches to monitoring ring traffic: two involving special hardware and a third based on the transmission of extra logging packets to a logging station at the beginning and end of every block. The latter approach is being implemented since it has the advantages of requiring no special ring hardware and a degree of data compression is provided at source.

8. T.E. Schutt, "Byte Streams on a Microcomputer", UKC Computing Lab Report 6, University of Kent (May 1981).

Considers in detail the theoretical and measured effects of various implementation parameters on the performance of BSP.

Applications

1. P. Lloyd, "Data Capture Across the Ring", INDRA 933, University College London (25 June 1980).
   A working paper describing a basic data capture facility implemented between an LSI-11 and a Unix system.

Network Standards

1. "A Network Independent Transport Service", SG3/CP(80)2, Study Group Three, Post Office PSS User Forum (February 1980).

Other Related Material

1. C.J. Bennett, "Clean and Simple", INDRA 790, University College London (September 1980).
   A Unix transport service interface in use at UCL.

2. R.T. Boyd and C.J. Hughes, "High Availability Rings for control of Telecommunication Switching", British Telecom (Mar 1981).

3. M. Bozyigit, "Graphics-Aided Interactive Modelling for Distributed Computer Systems", Polytechnic of Central London.

4. G. Fayolle, E. Gelenbe, and G. Pujolle, "An Analytic Evaluation of the Performance of the 'Send and Wait' Protocol", IEEE Transactions on Communications, Vol. Com-26, No. 3 (March 1978).
   Error control by positive acknowledgement and time-out periods ('send and wait') is a widely used technique, particularly in existing ring protocols. This analysis obtains optimum values of the time-out in order to maximize throughput, or to minimize average delay through the node or buffer overflow probabilities.