

Number 90



UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

The Cambridge Fast Ring networking system (CFR)

Andy Hopper, Roger M. Needham

June 1986

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500
<http://www.cl.cam.ac.uk/>

© 1986 Andy Hopper, Roger M. Needham

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

<http://www.cl.cam.ac.uk/techreports/>

ISSN 1476-2986

The Cambridge Fast Ring Networking System (CFR)

Andy Hopper and Roger M. Needham
University of Cambridge
Computer Laboratory

Abstract

Local area networks have developed from slow systems operating at below 1MBs to fast systems at 50MBs or more. We discuss the choices facing a designer as faster speeds for networks are contemplated. The 100MBs Cambridge Fast Ring is described. The ring protocol allows one of a number of fixed size slots to be used once or repeatedly. The network design allows sets of rings to be constructed by pushing the bridge function to the lowest hardware level. Low cost and ease of use is normally achieved by design of special chips and we describe a two chip VLSI implementation. This VLSI hardware forms the basis of a kit-of-parts from which many different network components can be constructed.

Keywords

Local area networks, metropolitan area networks, empty slot ring, synchronous transmission, bridges, VLSI.

The Cambridge Fast Ring Networking System (CFR)

Andy Hopper and Roger M. Needham
University of Cambridge
Computer Laboratory

1. Introduction

The Cambridge Ring is a local network that was designed in the mid 1970s for linking digital devices. At that time the speeds easily achieved using standard TTL technology and twisted-pair transmission systems were about 10MBs. The Cambridge Ring was designed to operate at that speed and was designed to provide an easy way to link digital devices. To achieve this goal a very small packet size (minipacket) was chosen comprising only two bytes of data and two bytes of address information. This meant the whole minipacket could be easily stored in the station logic and a minimum of time critical response was required of the attached device. This proved to be good design decision and a wide variety of devices were attached to the network. The main application areas of the Cambridge Ring were for peripheral sharing, which did not require particularly large bandwidths, file transfer which required medium bandwidth, and some exploratory work was done with simple voice systems. While the minipacket size meant that buffering could be done within the station logic it also meant that the maximum point-to-point bandwidth achievable to any pair of communicating devices was constrained. Because of the empty slot scheme a transmission could only take place just under once per ring revolution and for a practical network operating at 10MBs the normal point-to-point bandwidth was about 1MBs.

This paper describes the design of the Cambridge Fast Ring (CFR) which is based on experience with the earlier system. The design goals of the CFR were to provide a faster network both in terms of line transmission speed and the point-to-point bandwidth attainable by a pair of users. It was not a design goal to make the two systems compatible. It was considered an important design goal to define a kit-of-parts which could be used in many applications encompassing both local area networks (LANs) and metropolitan area networks (MANs). It was also considered from the start that the system should be integrated onto a VLSI chip to make it suitable for use even in inexpensive systems. At the same time the design was to be expandable to encompass high-speed applications such as transmission of video.

The original system design of the CFR was completed in 1983, all CAD tools were written in-house in parallel with the VLSI design which was finished in 1984 and working silicon was received and operational at the beginning of 1986.

2. Design choices in an empty slot network

Slot size

When designing a ring based on the empty slot principle the main parameters to be considered are the speed of transmission and the way slots are used.

The speed of transmission is important because as well as providing the base transport medium it directly effects the number of bits stored in the ring. In the original Cambridge Ring the delay through a repeater was 3 clocked bits while the delay through the wire at 10MBs was about 60 bits/kilometre. Thus a typical Cambridge Ring consisting of two dozen stations and 1 kilometre of wire was about 132 bits long.

If we now consider a faster system, operating at 100MBs the number of bits stored in 1Km of wire will be about 600. The delay through a repeater is also likely to increase because for implementation reasons we may convert the serial transmission path to one eight bits wide and every clock tick will be the equivalent of 8 bits serial delay. Thus the ring above operating at 100MBs is likely to have a delay of about 1400 bits.

This greatly influences the choice of slot size since if the slots are short a great number of them will exist on the ring. With use of slots restricted to the simple once per revolution scheme the point-to-point bandwidth will be low. We later deal with how this can be overcome by altering the rules for use of slots; however with the simple scheme we have to consider increasing the slot size from the two bytes of data of the Cambridge Ring.

There are two conflicting considerations when doing this. On the one hand we want to make the slots long so that only two or three exist on a typical ring and the round-robin partitioning of bandwidth is coarse. On the other hand many studies have shown that most packets on a LAN are short, perhaps 16 bytes or less, and making slots much longer than this would lead to wastage.

The size of the data field in a CFR slot was chosen to be 32 bytes. This size means that most of the short transmissions observed on the Cambridge Ring can be contained in one slot. If a slot were to be used for transmission of voice samples then the extra delay due to the loss of one or two packets because of errors would not be catastrophic. At the same time a 32 byte data field means that even for large rings the maximum point-to-point bandwidth is going to be maintained at a reasonable level. Furthermore integration in VLSI of a controller which includes packet buffers of this size is not likely to make the chip prohibitively large.

Transmission protocol

In an empty slot ring of the Cambridge type the transmitter marks a slot full, the slot makes its way to the destination where it may be copied by the receiver, and then returns to the source where it is marked empty. This means the delay through a node can be kept to a minimum and a path for response information is available back to the sender. If the transmitter and receiver are well matched in speed then transmissions proceed in the normal way. Because the destination may be either slower than the sender or subject to fluctuations, it is useful to provide a repeat facility in the transmitter hardware which retries the packet a

number of times if it has not been accepted. If the source is not multiplexing transmissions to different destinations this does not affect any other traffic and the wasted bandwidth is not likely to be of importance.

While it is attractive to partition bandwidth in a LAN to make the guarantees on delay performance tighter, it is also useful to consider what is the maximum achievable throughput. In a Cambridge Ring a sender can transmit every $N + 2$ slots under optimal traffic conditions where N is the number of slots on the ring. The first of the two additional slot delays is lost because the returning transmission has to be marked empty. The second additional slot delay is lost because it is difficult both to arrange for the logic to load a new packet and to mark the following slot full in time. For a ring with only one slot the maximum throughput is thus approximately one third of the line rate. For this reason it is attractive to consider schemes where either more than one packet can be outstanding from the sender, or where the slot is allowed to be reused by the transmitter. More than one packet outstanding will be useful on long rings with many slots. Reuse of a slot will be particularly attractive on short rings with only a small number of slots. With a slot size dominated by a 32 byte data field it is likely that most CFR implementations will only have a small number of slots and a scheme for replenishment of slots was chosen.

One of the most important advantages of LANs is that packets do not move out of order within the network and thus elaborate resequencing procedures are not required. When considering replenishment or channel mode, we have to make sure packets do not move out of order. When a packet makes its way to the destination it is only after the address field has been read that the packet can be received. This normally means that the response information can only be placed at the back of the packet, unless we are prepared to accept very large delays through each node. With channel mode a transmitter will replenish a returning slot by leaving the full/empty bit full and writing new data on top of the old. However, the response information will only become available at the end of the slot and may well indicate the previous packet was not accepted. A second, out-of-sequence packet will now be under way and if nothing is done the higher levels of protocol have to deal with the resequencing problem.

There are a number of ways of resolving this difficulty. At one extreme we can deem this kind of use of the ring to be acceptable - perhaps in applications where voice is being transmitted - and let the higher levels of protocol deal (or not deal) with sequencing. If this is not acceptable the following ring level protocol can be used to keep order. The "response" field is used in the forward direction to indicate "cancel this transmission". On discovering that a channel mode transmission has replenished a slot which contained a previous packet which was not received the sender cancels the transmission by modifying the response field. The packet now makes a superfluous circuit of the ring but is certain not to be accepted by the receiver. The transmitter is informed of this and can backtrack to the previous packet. If this operation is to be done in hardware and invisibly to the user then we need double buffering in the transmitter. The first packet is retained until it is known that it has been received successfully. This leaves some time to load the next packet and the hardware can recover the correct sequence completely. On a ring with only one slot, channel mode gives us the equivalent of a token ring but in practice it may be difficult to load successive packets in time. While double buffering is only required on the transmit side, it is useful to also provide it on the receive side to balance the traffic handling capabilities of the node hardware.

In the architecture of the CFR it was decided to control use of channel mode by using an extra bit at the front of each slot. Only slots with this bit set can be used in channel mode and the number of such slots is controlled centrally. Thus for a

particular implementation there may be a number of slots of each type. Both normal and channel slots can be used by any sender and if the transmissions are fast enough to replenish the double buffer in time, an automatic and transparent acceleration to the new speed will take place if a channel mode slot arrives. If the transmitter subsequently slows down and cannot keep up at channel mode speed transmissions will go back to normal mode automatically. It was decided that if a channel mode transmission returns not accepted and the backtrack takes place the channel mode slot has to be released.

Addressing

Traditionally LANs, whether rings or busses, have consisted of a single network interconnecting the communicating devices. This has the advantage that the system is simple, each packet moves past every possible destination, and the issue of routing does not have to be considered. As use of LANs has increased they have become interconnected using gateways. A gateway normally consists of a computer with a good size buffer which is able to smooth traffic flow between connected networks. This computer is also normally capable of interpreting a part of the protocol structure so that error checking and status information can be generated. It is attractive to consider the design of a LAN, or what is more recently called a metropolitan area network or MAN, which has the bridge facility built in at the lowest levels and which thus potentially allows bridges to be used with little protocol or cost overhead.

Address spaces can be defined to be hierarchical or flat. A hierarchical address space is often divided into ring address and station address. A bridge then examines the ring address part and makes a decision on whether to accept the packet according to some pre-determined rule. This could involve a table look-up or some simple examination of the address bits if the topology allows a quick routing decision.

A flat address space allows addresses to be assigned in any way and the address decision at a bridge is performed on the whole address field. It is convenient to arrange for this to be performed as fast as possible. In particular, if the result can be computed before the packet has completely moved past a bridge, bridge congestion is minimised. A 16 bit address field gives a large number of possible addresses and for a scheme that does not restrict topology it is convenient to use a 64KBit RAM to perform the table function. Each bit in the table indicates if the packet should be copied to the next ring and the routing tables are configured at the start. While it is possible to conceive of duplicate paths between pairs of users this is likely to lead to packets getting out of order. However, there is every possibility of many bridges existing on each ring and performance and reliability being improved by the generous supply of capacity in the network.

A simple ring system lends itself to the use of a broadcast addressing mode since each packet travels past every possible destination. With rings linked by bridges the table entries for the broadcast address have to be chosen to make sure no circular paths exist for broadcast packets since this would mean they would duplicate and circulate in the network forever. To make sure no circular paths exist the broadcast address entries in the map tables should map a tree structure on top of the system of interconnected rings.

In the CFR destination address filtering is used to route packets through the network but in addition we have also considered how further filtering may be performed by use of other fields within the packet. In some networks special type bits are used to mark packets belonging to various categories of traffic and

packets are rejected on this basis. A completely general scheme would enable any field in the packet to be examined and a reception decision made on that basis. In the CFR we have chosen a more restricted scheme where only the source address is used and is looked up in a table similar to that for the destination address. This table can be termed the hate list (or love list) and packets from any source can be prevented from being received by the hardware. Such a facility is useful where the receiver is spending a disproportionate amount of time dealing with unauthorised transmissions preventing it from doing useful work. In addition to the hate list each receiver on the CFR possesses a select register. This is a register which contains one 16 bit source address which defines the only source from which receptions are permitted. The select register can also be set to indicate receive from everybody or receive from nobody. This select register is primarily used when multiplexing is not permitted at a fine level and sequences of packets are to be received from one source without interruption from others.

3. CFR system architecture

CFR design

The CFR consists of **stations** used for communication between devices, **monitors** used for setting up and maintaining the slot structure and **bridges** used for copying packets between rings. We have implemented an integrated version of the system where a single CMOS network controller chip can be configured to perform any of these functions. A possible network configuration of the CFR is shown in Fig.1. The packet format of the CFR is shown in Fig.2.

Each slot begins with a **start-of-packet bit (SOP)** which is always set to one. This is followed by the **full/empty bit (F/E)** which indicates whether the slot is in use. Following this is the **monitor passed bit (MP)** which is used to delete lost packets at the monitor. Finally in the start information there is a **channel slot bit (CS)** which indicates the slots that are available for channel mode use. This is followed by the **destination address (16 bits)**, the **source address (16 bits)**, the **data field (256 bits)**, and a **CRC (12 bits)**.

The CRC is used both as an error check and as a way of passing information from the destination to the source and from the source to the destination. This is done by corrupting the last four bits of the CRC and ensuring that the action for a bad CRC gives the required results.

When a packet returns to the sender with a bad CRC it is deemed as accepted, on the assumption that the error occurred on the return path from destination to source. If the data is received correctly at the destination the CRC is made bad by inverting the last four bits thus also indicating to the sender that transmission was successful and no repeats are necessary. The CRC is returned correctly set if the receiver does not receive the packet for any reason.

In the forward direction a transmission is cancelled in channel mode by making the outgoing CRC bad. The destination corrects a bad incoming CRC (either due to error or a "cancel" in channel mode) but does not receive the packet, the good returning CRC indicating to the source that the packet was not accepted and may have to be retransmitted.

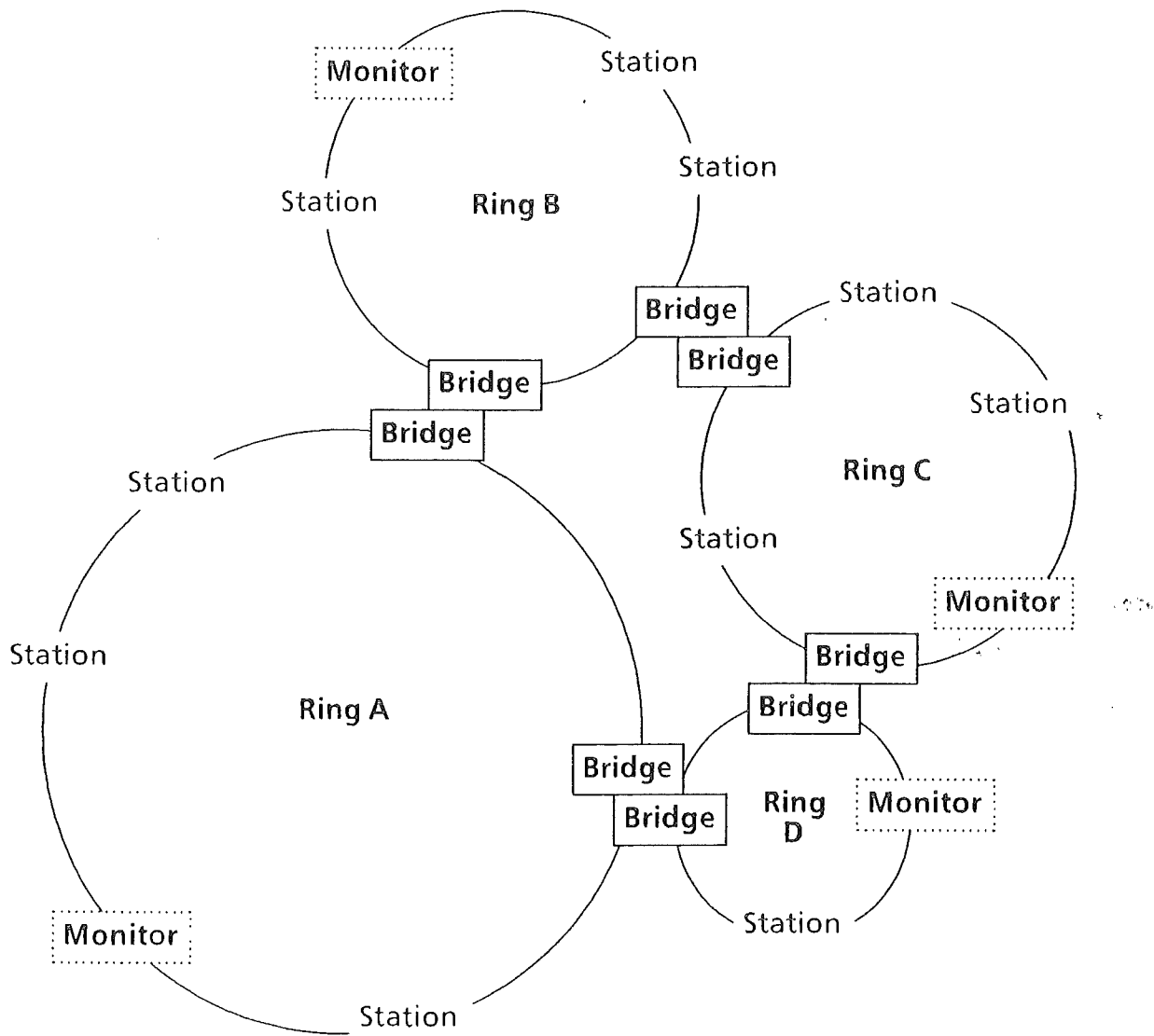


Fig.1. CFR network architecture

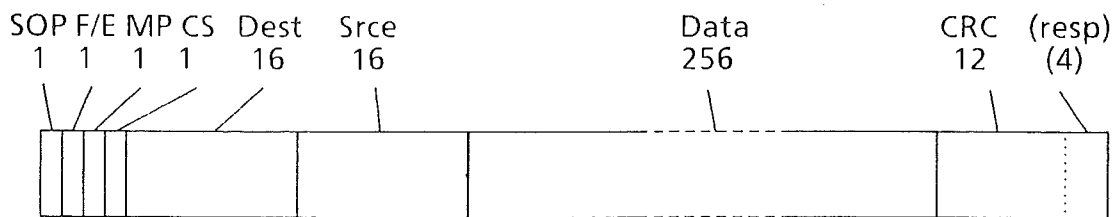


Fig.2. CFR packet format

It should be noted that with this scheme a transmission error occurring on the return path may make it look as if a packet which should be repeated was received correctly. The response information implied from the CRC is only used

by the station hardware and is not made available to the user who has no direct knowledge of the fate of a packet once it has been loaded into the station. The operation of the CRC field is summarised below:

	at receiving station	on return to transmitting station
CRC bad	not received mark CRC good	packet assumed received
CRC good	if received mark CRC bad; if not received mark CRC good	packet assumed not received

CFR CRC/Responses

When a user wishes to transmit a packet it is loaded into the transmit FIFO which is in the network controller. When an empty slot arrives the packet is emptied into the slot and makes its way to the destination. At the destination the packet may not be accepted if the destination was busy, unselected, a CRC error in the forward path was detected, or the source was in the hate-list. Otherwise the packet is accepted and the CRC field is marked in the appropriate way. The packet returns to the source which automatically repeats the transmission a number of times if the packet was not accepted and there was no CRC fault in the return path. If the packet is a **broadcast packet** the destination address FFFF is used. At a bridge no hate list is available, the 64K RAM table being used to indicate on-the-fly on which addresses the bridge potentially receives. The bridge maps (and hate-lists at stations) are initialised and maintained by the user through the **host interface**.

The monitor is used to configure and maintain the slot structure and to receive maintenance packets. The monitor has address zero and maintenance packets are transmitted to this address. The monitor can also receive in the normal way but it is not capable of transmitting normal packets. The number of slots is controlled by the user through the host interface. The monitor sets up the requested slot structure by initially marking all slots full and placing zeros in the other fields. Stations synchronise to this data stream by waiting for a one, assuming this is a start bit, and counting for one slot time. Providing no errors occur for four ring revolutions the monitor releases the ring for general use by marking the full/empty bits for all slots empty. The slot train put out by the monitor is head-to-tail contiguous, with the surplus delay round the ring occurring in one **gap** which is all zeros. The gap is used as a marker by each station to count and automatically compute the number of slots in the ring.

Maintenance packets are launched by stations to indicate a number of faults. The monitor also launches maintenance packets which can indicate additional faults only detected at a monitor. Such packets do not use the normal transmission mechanism and are deleted when they first pass the monitor. The major maintenance mechanism is to check and correct the CRC on each link for

empty slots. Thus faults are localised to a link and can be reported to the monitor by use of maintenance packets. The maintenance mechanism can also detect and report line breaks.

Network node design

The basic design of the **network node** consists of a high speed **repeater** which performs the serial to parallel conversion for a slower **network controller** which operates with an eight bit wide datapath. A diagram of a general hardware implementation of the CFR is shown in Fig.3.

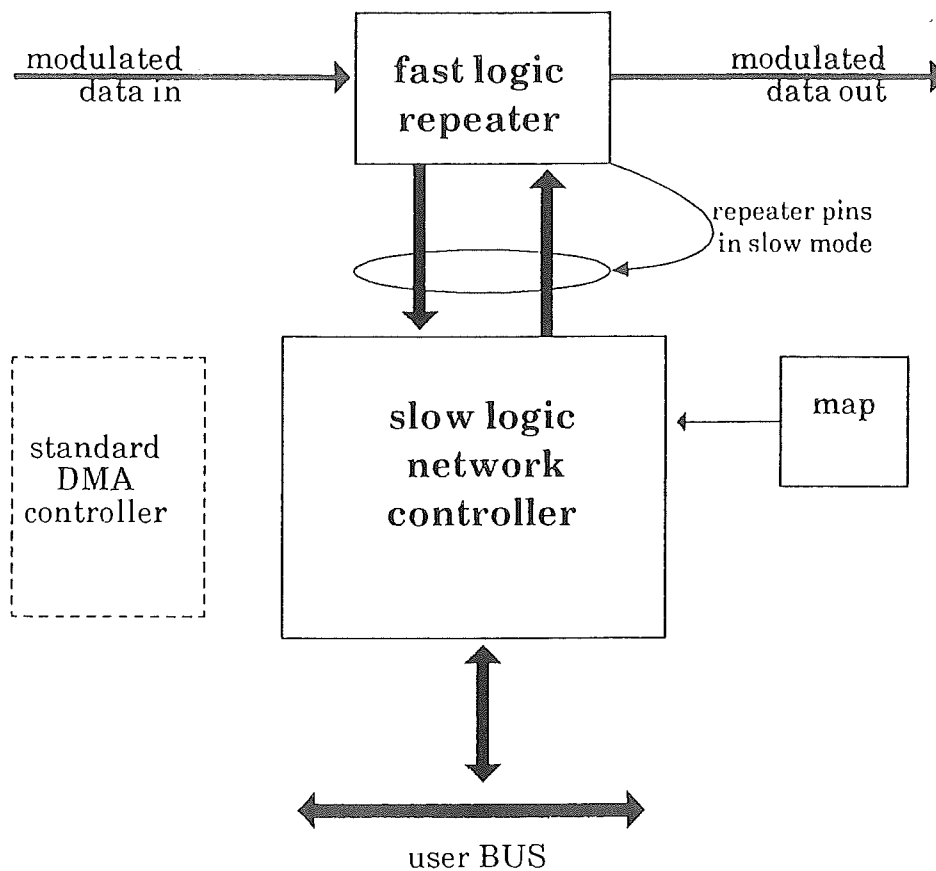


Fig.3. CFR node structure

With normal pin count constraints a one-to-eight speed division is possible and the speed difference between ECL and CMOS logic can be nicely matched. However, if the fast repeater is not used, at least 16 pins are made available for other use. Thus it is possible to easily implement a network controller which duplicates the logic of the repeater and provides a very low cost component. This component would only be able to operate at a serial rate constrained by the speed of the slower logic but there is every possibility of slow rings forming a part of the CFR networking system.

We will now discuss how various CFR node structures can be constructed by using a parallel data path between network controllers. Only short data links of around a metre will be feasible using these methods. When several devices are to be attached to the ring at a single point, each device can have its own controller and they can be attached to one repeater as shown in Fig.4. This approach can be

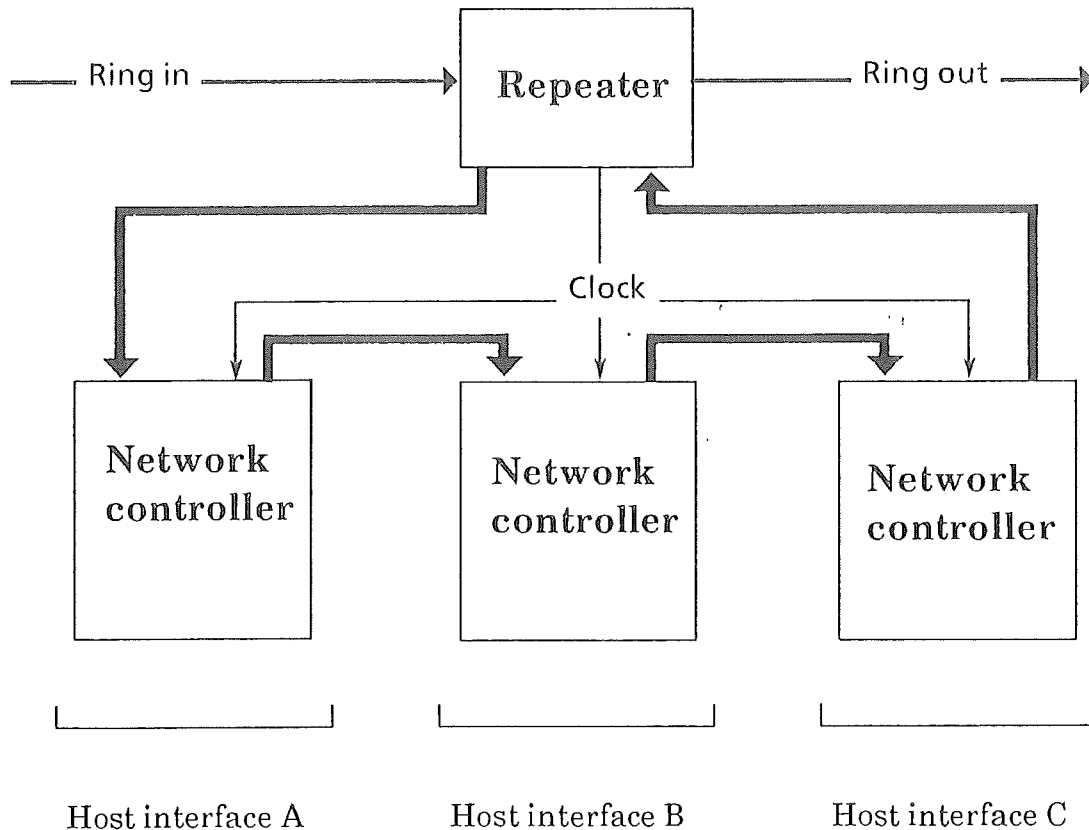


Fig.4. CFR station cluster

extended to use an 8 bit wide datapath throughout. This system can then be used to produce a fast switch, which takes the form shown in Fig.5.

The pin count constraints on a VLSI implementation of the network controller mean that only a half-duplex host interface is possible. In some high performance applications this may be too restrictive so we have implemented a controller which can be used in receive only mode or transmit only mode. Thus two such controllers can be used at a single station to provide a fully duplex connection as shown in Fig.6. Indeed several transmit controllers can be assigned one address and the transmit and receive sections do not have to be on the same ring.

Implementing CFR networks

It can be seen that the CFR networking system is an attempt to define an architecture which allows many different application areas to be encompassed. The kit-of-parts can be used to make a fast parallel inter-processor switch at one extreme, or a slow and cheap network at the other. It can be used to build single

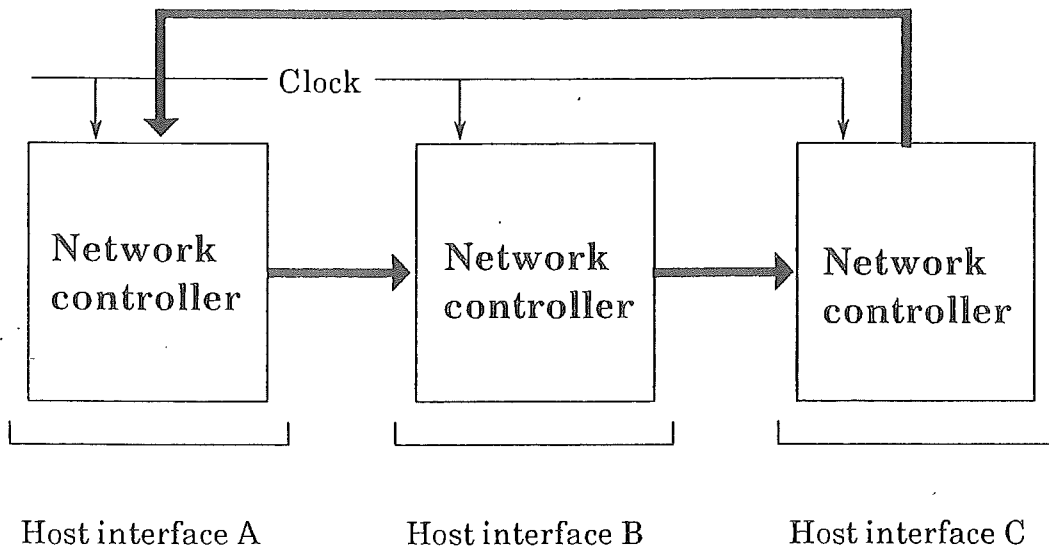


Fig.5. CFR parallel ring switch

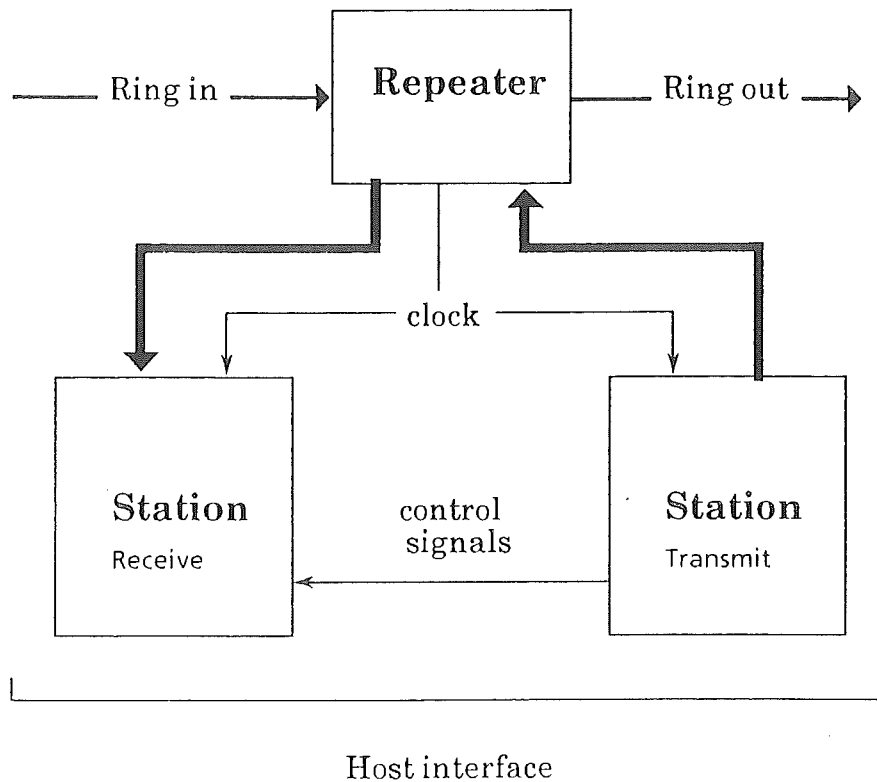


Fig.6. CFR duplex station

ring LANs or by using bridges MANs and other larger area networks. The larger network can then be used to implement a reliable system by duplicating higher

level services and functions thus making failures of distant equipment or services less catastrophic to the user.

Initial implementations of the CFR are likely to use a fast central ring with other slower rings being attached to it. The central ring may consist of a number of parallel clusters linked by high speed fibre optic cables operating at about 75MBs. The slower links are likely to drive normal twisted pairs and operate at about 10MBs. This will give a typical point-to-point bandwidth of about 15 MBs on the faster ring and about 2.5 MBs on the slower rings. It is also possible to consider using a long distance duplex line operating a CFR structure as a link between centres tens of kilometres apart while retaining a uniform architecture and addressing scheme.

One of the issues to be examined carefully when configuring a network topology is the use and location of bridges. Since the bridge function in the CFR is both simple and pushed down to the hardware it should be possible to have an unusually high ratio of bridges to stations. Thus a network may start off with perhaps 10 stations per bridge and this number may decrease to only 2 or 3 stations per bridge as load and traffic increases. A simple bridge design uses two network controllers back-to-back with little other associated hardware. This provides buffering for two packets in each direction and under high loads is likely to cause some incoming packets to be dropped. However, under light loads, which are often experienced on LANs, this amount of buffering is likely to be sufficient. A further difficulty with buffering at bridges may be experienced if fast transmitters are using bridges to send to slow receivers. While this is not a problem on a single ring since the round robin passing of slots prevents hogging, a congested bridge blocks other communication paths through it. This problem is exacerbated if a fast ring is emptying onto a slow ring or if a channel mode transmission can only achieve normal mode after passing through a bridge. One solution to this problem is to insist that a higher level handshake protocol is used between the end-to-end users. Alternatively the time for bridges to drop packets can be made short by setting the repeat packet facility to a small value.

An important consideration with bridges is the generation and maintenance of bridge address maps. Where the network topology is simple, it may be possible to constrain use of addresses so that the map entries never change. However this would mean that devices cannot be moved between rings without their address being changed, and that the network topology could not change. A more elaborate system would use the ring to distribute information to bridges. To do this each bridge would have an associated microprocessor controller with its own ring address to which packets containing map information could be initially sent. Once the first level of bridges is operating the next level of bridges could be addressed and so on. Another way of addressing bridge controllers could be by the use of broadcast packets but this may be more difficult since with such transmissions it is more difficult to be sure which packets have been received.

Another approach to the use of bridges would be to cluster them onto a single board. A single computer can be used to control the cluster and if no other bridges in the system are allowed this would provide a direct way of reconfiguring the network structure. This is not a flexible approach and it is more likely that many such clusters will exist. However, if the rule is that parallel rings of bridge clusters are linked by a single ring all bridge controllers can be addressed directly while many topologies for user traffic are still possible.

With the CFR architecture it is possible to use bridges as network controllers for normal stations. This has the effect of giving each station many addresses, which can be duplicated if on different rings, and thus allowing addressing to processes within the attached devices. This means such processes can move between

stations by only changing the contents of the address tables. Well known services, such as name translation, can now be implemented in a number of machines and activated by creating the appropriate address within the table of the device actually performing the service.

4. VLSI implementation

A first VLSI implementation of the CFR has been developed using an ECL repeater chip and a CMOS network controller chip. The ECL chip is a gate array of about 350 gates and the CMOS device is a custom design of about 25K transistors. The ECL component operates at a worst-case speed of 100MBs consuming about 1.5W of power. The CMOS chip uses 8 bit datapaths throughout and can be configured to be either a station, a bridge or a monitor and can be set up by the user in either polled, interrupt or DMA modes. It is designed to attach to standard DMA controllers and uses a 64K DRAM for table lookup. The CMOS chip was conservatively designed using single metal, single polysilicon, 3 micron technology and operates at a equivalent line rate of 60MBs. This chip is designed for direct shrinking to 2 micron technology which will increase the equivalent serial speed to about 90MBs. The delay through the repeater chip is 3 bytes and through the network controller chip 2 bytes. The minimum gap length is 3 bytes so to configure the shortest ring consisting of only one slot about 41 bytes of delay must be present. The basic hardware building blocks of a CFR node using the VLSI components are shown in Fig.7.

In the initial VLSI implementation two simplifications have been made to reduce the size of the CMOS chip to give reasonable yields. Firstly double buffering and channel mode have been left out which means that the maximum speed at which a user can transmit is reduced. However we have made the optimisation of allowing the transmit buffer to be filled again as soon as a packet has been launched on the ring providing the repeat count has been set to zero. Secondly the logic of the ECL repeater has not been duplicated on the CMOS network controller. This increases the cost of the cheapest system, however it is possible to design a very inexpensive CMOS repeater if required.

ECL Chip

This chip provides differential ECL level line drivers and receivers for the Cambridge modulation system. This modulation uses two data paths between repeaters, is well suited to twisted-pair wire and provides a change on every clock tick to enable the clock to be recovered from the data. Up to about 100 feet of standard twisted pair wire has been driven at 100MBs with an error rate lower than 10^{-9} . The ECL chip also provides unmodulated serial data inputs and outputs which have been used with other transmission media such as fibre optics.

The ECL chip is designed to be clocked in a number of ways. If serial data is used then an external clock must be supplied. If modulated data is used, it is possible to self-clock the chip using the phase comparison signal although this is not recommended since jitter will not be controlled. The preferred method of clocking

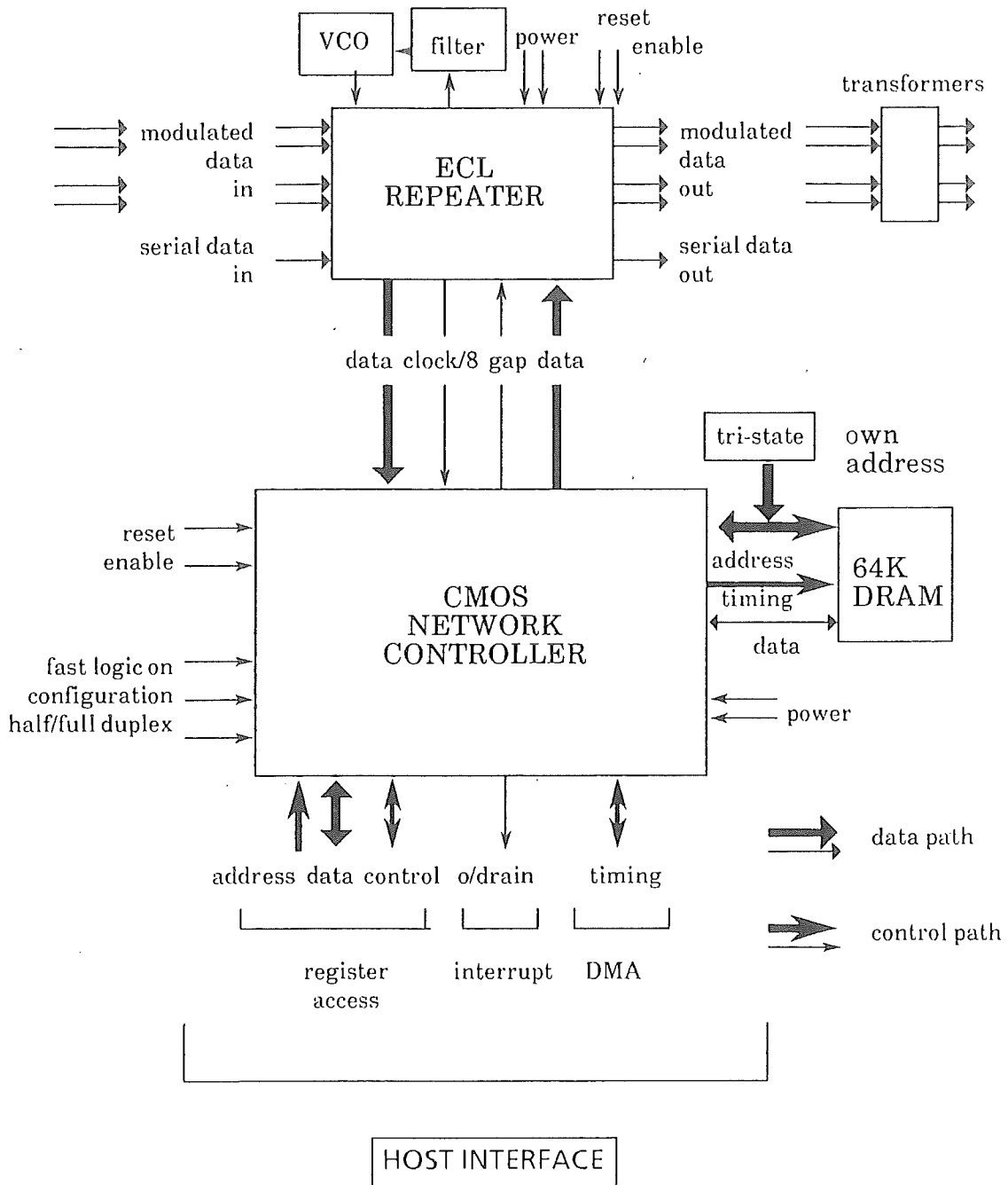


Fig.7. CFR node components

is with a phase locked loop. External circuitry is required to provide the VCO and loop filter.

The length of wire and the number of stations dictates the number of bits round the ring and will not necessarily be an integer number of bytes. The odd bits will occur in the gap, so the ECL chip must resynchronise the C/8 clock at the end of every gap to prevent the slow logic having to deal with a short clock. This is

accomplished by stretching the C/8 clock and thus a C/8 clock half-period is always guaranteed to be at least 4 bits long.

Power consumption of the chip is about 1.5W and it is run from 0V and +5V supplies. Outputs to slower logic are TTL compatible and to ensure correct operation it is essential that the 5V supply should be carefully decoupled. The ECL inputs and outputs are thus +5V offset from their normal values. A pin out diagram and a description of the pins are given in appendix A.

CMOS Chip

The design is manufactured in 3 micron CMOS technology with a single metal layer and a single polysilicon layer. The logic design is mostly synchronous to the clock and thus there should always be a speed at which the chip operates. However, the user device is likely to have its own clock and thus great care is taken to settle all signals at the host interface to minimise flip-flop settling errors.

There are four broad types of signal emanating from the CMOS chip. The repeater interface is used for writing parallel ring data to and from the chip. The map interface is used to connect to the 64K DRAM. The host interface is for attaching the chip to the user device. Finally there is a set of general purpose signals used for configuring the chip.

The CFR has a user interface which can be configured in a number of ways. A simple system will use a polling scheme by decoding the network controller somewhere in the address space. A more complex system will take advantage of the interrupt facilities and for high performance external DMA chips can be utilised.

The connection between the user device and the ring controller is achieved through a number of control registers. In addition, in order to allow high speed logics to use the registers rapidly, five hard lines are provided for manipulating data. These indicate by edges when a packet has been transmitted or received (PTXBar, PRXBar), and when register operations are complete (RDYBar). Two signals are used for moving to the next packet when not all 32 (36 at a bridge) bytes of data have been read or written (DCBar, TUBar). There are two registers (0 and 1) which do not generate the RDYBar handshake and can be used for writing high speed data. In this case the speed of writing should be controlled by an external delay.

The speed of writing to or from the CMOS chip is influenced by the type of interface in use. Reading and writing the 32 bytes of data has been optimised and in the 3 micron chip design a byte can be written every 120 nsec. After a complete packet has been loaded a further delay ensues while it is transmitted. Even if this happens immediately a further packet delay has to pass before the next packet can be loaded. Updating of the address registers is slower and is approximately equivalent to the loading of eight bytes of data. Operations such as changing the select register, maps, or broadcast register take approximately one packet transmission time because the contents of the chip have to be checked against the new register values.

5. Conclusion

The design of the CFR is aimed at finding a solution to the problem of supporting both bursty traffic generated by computers and synchronous traffic generated by voice and other real time systems. The CFR attempts by the use of fine-grain bandwidth sharing to make the worst-case excursions of performance well controlled. Channel mode slots allow some transmissions to proceed rapidly and may well be used where large files or other bulk data is being transmitted. A CFR implementation with a single slot used in channel mode makes the system equivalent to a token ring.

The hardware implementation of the CFR hides the complexity of the network from the user and thus makes it possible to attach devices easily. The network architecture makes it possible to use bridges liberally and allows partitioning into autonomous regions resilient to failure. Rings of different speeds can be connected and thus performance and cost is only increased where required. The VLSI implementation has shown that it is possible to design chips which can be used as the basis of a versatile kit-of-parts for implementing CFRs in many different applications.

There is scope for rings operating at higher speeds than those currently achieved by the CFR. The natural division of bandwidth by the empty slot principle is even more desirable as the speed of transmission increases further. It is an open research question how this bandwidth should be partitioned and what sort of performance guarantees and interface should be presented to the user.

Acknowledgment

We would like to acknowledge the help and encouragement of the many people involved in this project. Tony Mann, Trevor Morris, John Porter and Chris Stenton spent many months writing the CAD software and implementing the VLSI design. David Wheeler and Steve Temple provided continual critical comment of design decisions. Maurice Wilkes was always a source of encouragement even when a working system seemed a long way away.

References

- [1] Wilkes, M.V. and Wheeler, D.J., "The Cambridge Digital Communication Ring", Local Area Communications Network Symposium, Boston, May 1979.
- [2] Hopper, A., Temple, S. and Williamson R.C., "Local Area Network Design", Addison Wesley, London 1986.
- [3] Sze, D.T.W., "A Metropolitan Area Network", IEEE Journal on Selected Areas in Comms., vol. SAC-3, no. 6, Nov. 1985.

- [4] Needham, R.M. and Herbert, A.J., "The Cambridge Distributed Computing System", Addison Wesley, London 1982.
- [5] Temple, S., "The Design of a Ring Communication Network", Ph.D. Thesis, University of Cambridge, Jan. 1984.
- [6] Leslie, I.M., "Extending the Local Area Network", Ph.D. Thesis, University of Cambridge, Feb. 1983.
- [7] Bux, W., Closs, F.H., Kummerle, K., Keller, H.J. and Mueller, H.R. "Architecture and Design of a Reliable Token Ring Network", IEEE Journal on Selected Areas in Comms., vol. SAC-1, no.5, Nov. 1983.
- [8] Falconer, R.M. and Adams, J.L., "Orwell: A Protocol for an Integrated Services Local Network", Journal of British Telecom Technology, vol.3, no.4, Oct 1985.
- [9] Bergman, L.A. and Eng, S.T., "A Synchronous Fiber Optic Ring Local Area Network for Multigigabit/s Mixed Traffic Communication", IEEE Journal on Selected Areas in Comms., vol. SAC-3, no. 6, Nov. 1985.

Appendix A - ECL repeater pin-out

1	Vee	GAP	40
2	INBL	CK	39
3	INBR	SRIN	38
4	INAL	DVC	37
5	Vcc	GISD	36
6	INAR	Vee	35
7	LOUT7	RES	34
8	LOUT6	MODEr	33
9	LOUT5	PHCp	32
10	LOUT4	Vcc	31
11	LOUT3	CBY8	30
12	LOUT2	LNABar	29
13	LOUT1	LNA	28
14	LOUT0	LNBBar	27
15	Vee	LNB	26
16	LIN7	Vcc	25
17	LIN6	SDO	24
18	LIN5	LIN0	23
19	LIN4	LIN1	22
20	LIN3	LIN2	21

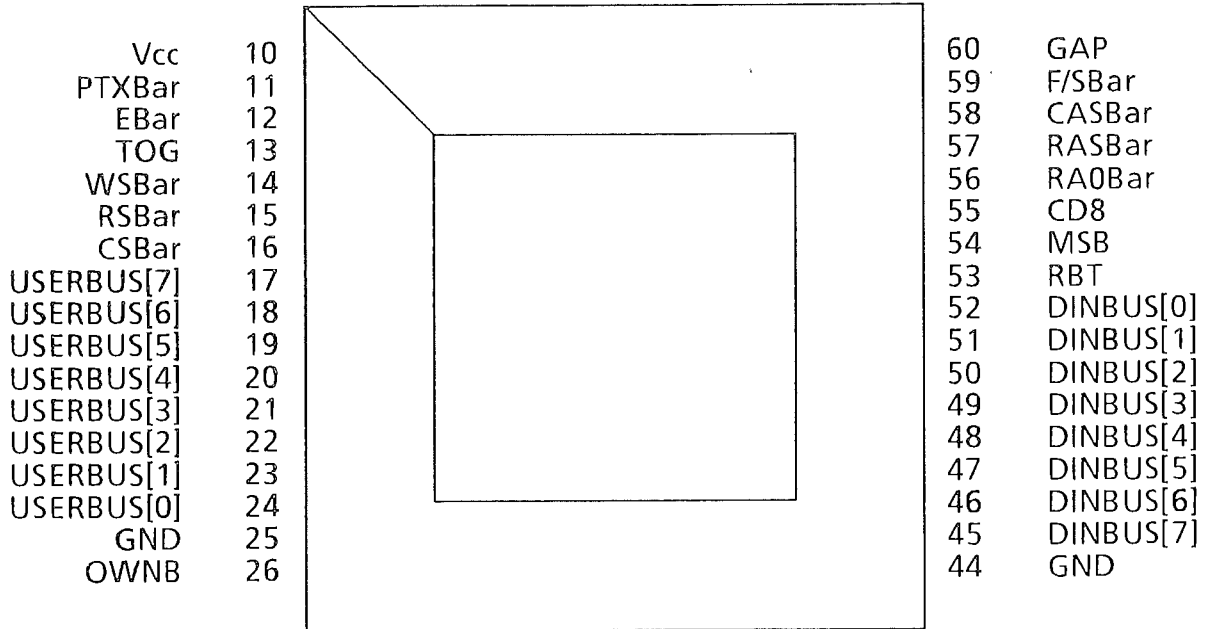
The ECL repeater pin-out

Vee	ground 0V separate supplies for peripherals 1,15 and internal logic 35	ground	1,15,35
INBL,R	differential line input B	ECL input	2,3
INAL,R	differential line input A	ECL input	4,6
Vcc	power 5V DC	power	5,25,31
LOUT0-7	data output bus to CMOS chip	TTL output	7-14
LIN0-7	data input bus from CMOS chip	TTL input	16-23
SDO	serial data out	ECL output	24
LNB, LNBBAr, LNA, LNABAr	differential line drivers A and B	ECL output	26,27 28,29
CBY8	byte rate clock to CMOS chip; positive edge	TTL output	30
PHCr	phase comparison signal between the local clock and the earlier of the incoming changes of the modulated data	ECL output	32
MODEr	modulation error - signal high if there is no change on either differential line receiver during previous clock	ECL output	33
RES	hardware reset for testing purposes only	ECL input	34
GISD	gate in serial data selects serial or differential inputs	ECL input	36
DVC	divide and copy - when high the ECL chip is enabled and operates normally, when low it merely copies the data. Note the ring is shortened when this signal is low	ECL input	37
SRIN	serial data input	ECL input	38
CK	bit rate clock; positive edge	ECL input	39
Gap	gap signal from CMOS chip enables re-synchronisation of C/8 signal at end of gap	TTL input	40

ECL repeater pin description

Appendix B - CMOS network controller pin-out and register structure

				P	R	R							M	I		R	R
	T	D	R	D	/	A	A	A	A	A	A	A	P	N		A	E
	U	C	X	Y	W	B	B	B	B	B	B	D	T		1	F	
G	B	B	B	B	B	U	U	U	U	U	U	A	B	D	B	B	R
N	a	a	a	a	a	S	S	S	S	S	S	T	a	R	a	a	E
D	r	r	r	r	r	[0]	[1]	[2]	[3]	[4]	A	r	E	r	r	r	S
	9	8	7	6	5	4	3	2	1	68	67	66	65	64	63	62	61



27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
D	D	D	D	D	D	D	D	M	M	M	M	M	M	M	M	V
O	O	O	O	O	O	O	O	A	A	A	A	A	A	A	A	c
U	U	U	U	U	U	U	U	P	P	P	P	P	P	P	P	c
T	T	T	T	T	T	T	T	B	B	B	B	B	B	B	B	
B	B	B	B	B	B	B	B	U	U	U	U	U	U	U	U	
U	U	U	U	U	U	U	U	S	S	S	S	S	S	S	S	
S	S	S	S	S	S	S	S	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	
[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]									

The CMOS network controller pin-out

ABUS0-4	address for access to control registers	TTL input	1-3, 67,68
R/WBar	read/write control for 64 DRAM	TTL output	4
RDYBar	ready; negative edge indicates write operation on registers is complete. Not used for registers 0 and 1.	TTL output	5
PRXBar	packet received; negative edge indicates packet has been received. Note: can be made invalid by some subsequent register operations	TTL output	6
DCBar	discards rest of packet on reception	TTL input	7
TUBar	tops-up rest of packet on transmission	TTL input	8
GND	ground 0V	ground	9,25,44
Vcc	power 5V	power	10,43
PTXBar	packet transmission completed; negative edge indicates packet has been transmitted and the buffer is free	TTL input	11
EBar	user synchronisation clock; negative edge	TTL input	12
TOG	thrown on ground; indicates when a node has given up transmitting a packet	TTL output	13
WSBar	register write strobe	TTL input	14
RSBar	register read strobe	TTL input	15
CSBar	chip select	TTL input	16
USERBUS0-7	user I/O bus	TTL bidirectional	24-17
OWNB	own back; used in a duplex bridge implementation to prevent reception of own broadcast packet	TTL bidirectional	26
DOUTBUS0-7	parallel data to ECL chip	TTL output	34-27
MAPBUS0-7	address output to DRAM; own address read in during first two bytes of gap	TTL bidirectional	35-42
DINBUS0-7	parallel data from ECL chip	TTL inputs	52-45
RBT	receive only - TTL lo both transmit and receive - Hi-Z transmit only - TTL hi	special input	53

CMOS network controller pin description

MSB	monitor - TTL lo. Note: a monitor can receive in the normal way but only transmits maintenance packets. station - Hi-Z bridge - TTL hi	special input	54
CD8	DINBUS and DOUTBUS byte rate clock; positive edge	TTL input	55
RA0Bar, RA1Bar	read station address enable (lo and hi)	TTL output	56,63
RASBar	row address strobe for DRAM	TTL output	57
CASBar	column address strobe for DRAM	TTL output	58
F/SBar	fast/slow refresh rate control	TTL input	59
GAP	gap signal to ECL chip	TTL output	60
RES	reset for chip testing only (active hi)	TTL input	61
REFBar	refresh control for DRAM	TTL output	62
DRE	disable - TTL lo responses off - Hi-Z enable - TTL hi	special input	64
INTBar	interrupt	open drain pull-down	65
MPDATA	data to and from 64K DRAM	TTL bidirectional	66

CMOS network controller pin description

	Read	Write
0	Receive Data Register. This is the 32 byte receive packet FIFO. When 32 bytes have been read out the next packet can be received. At a bridge this register is 36 bytes deep and includes the destination and source addresses of the packet. At a monitor this register is used for receiving maintenance packets, bit 5 in each byte of data indicating a parity fault, bit 6 gap too long, and bit 7 a monitor fault.	Transmit Data Register. This is the 32 byte transmit packet FIFO. When 32 bytes have been loaded they are automatically transmitted. At a bridge this register is 36 bytes deep and includes the destination and source addresses of the packet.
1	Receive Address Register. This is the 2 byte source address. It is available until the last byte of data is read. At a bridge this register is not used. At a monitor this register indicates the source of the maintenance packet.	Transmit Address Register. This is the 2 byte destination address. It has to be written before the last byte of data and is used in all subsequent packets until changed. At a bridge this register is not used.
2	Interrupt Mask Register. Gives value of interrupt mask.	Interrupt Mask Register. A high in a bit position enables the interrupt.
3	Interrupt Status Masked. Gives value of ISR after mask has been applied. The logical-OR of this register is used to switch on the open-drain pull-down (assert) of the interrupt pin.	
4	Interrupt Status Register. Contains values of bits used to generate interrupts. PRX, BPRX and PTX interrupts are cleared by either completely reading or loading the appropriate FIFO. RB, TOG and R814Bar are cleared by writing a zero to their bit position in the mask register.	
5	Transmit FIFO Available. Bit 0 indicates that the next byte can be written to the transmit FIFO and remains asserted until the last byte of a packet is written.	Top Up. Writing any value causes the rest of the bytes in the packet to be filled with undefined values and the packet to be transmitted.

CFR network controller registers

	Read	Write
6	Repeat Control Register.	Repeat Control Register. Bits 0,1 indicate number of repeat transmissions; 00 = >0 repeats (care to be taken with high speed devices); 01 = >3 repeats, 10 = >7 repeats, 11 = >15 repeats. Bit 2 indicates number of ring delays between each repeat; 0 = >0 ring delays, 1 = >3 ring delays.
7	Receive FIFO Available. Bit 0 indicates that the next byte can be read from the receive FIFO and remains asserted until the last byte of a packet is read. If a write to registers 8-14 is performed this signal will be de-asserted (unlike the pin which remains unchanged) while the contents of the receive FIFO are checked against the new configuration.	Discard. Writing any bytes in the receive FIFO and the next packet can be received.
8	Broadcast Receive FIFO Available. This is similar to register 7 but will only be asserted on reception of a packet on the broadcast address.	Broadcast Enable. Writing a 1 in any bit position enables reception on the broadcast address. At a bridge both this register and map location FFFF have to be hi to enable broadcast reception. The select mode takes priority over the broadcast mode.
9	Selected Everybody. Bit 0 is high when select register is set to everybody.	Select Everybody. Writing any value sets select mode to all sources.
10	Selected Nobody. Bit 0 is high when select register is set to nobody.	Select Nobody. Writing any value sets select mode to no sources.
11	Select Low. Gives low byte of select register.	Next Auto Select. Writing any value sets select mode to everybody until a packet is received, to which the select register is then set.

CFR network controller registers

	Read	Write
12	Select High. Gives high byte of select register.	Select Value. Writing any value causes the select register to be loaded with the current value in the latch register and the chip will only receive from that source. Note: values 0000 and FFFF are used as normal addresses.
13		Write Map. The address used for writing to the map is that currently in the latch register. The value written is a one if any bit position in this register is set and a zero otherwise.
14	Map Data. Bit 0 contains map value after last read request.	Read Map. Writing any value causes the map to be read at the address currently in the latch register.
15	Configuration. Hardware configuration of the chip. Bit 0 high when chip enabled on pin and in register. Bit 1 high when writing of response CRC enabled. Bits 2,3 indicate transmit only and receive only. Bits 3,4 indicate bridge mode and monitor mode.	Disable. The chip is enabled when either the pin or the register are high. Both have to be low to disable. Writing all zeros to this register will disable the chip after any outstanding transmissions are completed providing the enable pin is low.
16	Ready. Bit 0 is low when operations in response to a write to registers 2 - 24 are being performed.	
17	Latch Low. Gives low byte of latch register.	Latch Low. Updates low byte of latch register.
18	Latch High. Gives high byte of latch register.	Latch High. Updates high byte of latch register.
19	Own Address Low. Gives low byte of own address loaded through the map bus. Slot Control. At monitor this register gives current value of the slot control register.	

CFR network controller registers

	Read	Write
20	Own Address High. Gives high byte of own address loaded through the map bus.	
21		Slot Control. Writing a one in any bit position updates the slot control register from the latch register; 0 = >0 slots, 11111 = >31 slots.
22		Run Mode Enable. Writing a one in any bit position allows monitor to enter run mode providing no framing faults are being observed. Writing all zeros forces monitor into reframe mode. During reframe mode the monitor writes slots which are marked full but otherwise contain zeros.
23		Random Enable. Writing a one in any bit position enables insertion of a random number into empty slots.
24		Auto-Reframe Enable. Writing a one in any bit position enables auto reframe to take place. Auto reframe starts when a framing fault is detected and finishes when no framing faults are observed for 3 ring revolutions.
25	Monitor Status. Bit 0 gives run mode enabled, bit 1 gives random enabled, bit 2 gives reframe enabled.	

CFR network controller registers