The Cambridge Fast Ring Networking System

Andrew Hopper
Roger M Needham

ORL-88-1

T-CP/37/10//18427

The Cambridge Fast Ring Networking System

Andrew Hopper
Roger M. Needham

Reprinted from
IEEE TRANSACTIONS ON COMPUTERS
Vol. 37, No. 10, October 1988

The Cambridge Fast Ring Networking System

ANDREW HOPPER AND ROGER M. NEEDHAM, MEMBER, IEEE

Abstract—**Local area networks have developed from slow systems operating at below 1 Mbit/s to fast systems at 100 Mbits/s or more. We discuss the choices facing a designer as faster speeds for networks are contemplated. The 100 Mbit/s Cambridge Fast Ring (CFR) 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.**

Index Terms—**Bridges, empty slot ring, local area networks, metropolitan area networks, synchronous transmission, VLSI.**

I. Introduction

The Cambridge Ring is a local network that was designed in the mid 1970's for linking digital devices [1]. At that time, the speeds achieved using standard TTL technology and twisted-pair transmission systems were about 10 Mbits/s. 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 2 bytes of data and 2 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 a good design decision and a wide variety of devices were attached to the network. The main application areas of the Cambridge Ring were 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 by 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 10 Mbits/s the typical point-to-point bandwidth was about 1 Mbit/s.

This paper describes the design of the Cambridge Fast Ring (CFR) which is based on experience with the earlier system [2], [3]. The design goals of the CFR were to provide a faster network both in terms of line transmission speed and the point-to-point throughput 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 (LAN's) and metropolitan area networks (MAN's) [4]. 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 [5], 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.

II. 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 10 Mbits/s was about 60 bits/km. Thus, a typical Cambridge Ring consisting of two dozen stations and 1 km of wire was about 132 bits long.

If we now consider a faster system, operating at 100 Mbits/s, the number of bits stored in 1 km 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 8 bits wide and every clock tick will be the equivalent of 8 bits of serial delay. Thus, the ring above operating at 100 Mbits/s 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 transmission scheme, the point-to-point throughput will be low. We deal later 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 2 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. At 100 Mbits/s, this implies a slot length of 32 or 64 bytes for typical ring configurations. On the other hand, many studies have shown that most packets on a LAN are short [5], perhaps 16 bytes or less, and mak-

Manuscript received July 28, 1986; revised May 28, 1987. This work was supported by Acorn Computers Ltd. and Olivetti Research Ltd.

A. Hopper is with the Computer Laboratory, University of Cambridge, Cambridge, England and Olivetti Research Ltd., Cambridge, England.

R. M. Needham is with the Computer Laboratory, University of Cambridge, Cambridge, England.

IEEE Log Number 8718427.

ing slots much longer than this would lead to poor bandwidth utilization.

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. 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.

A slot size of 32 bytes is likely to have a number of advantages for voice transmission. It is sufficiently short that the loss of one or two packets due to errors is not catastrophic. It is unlikely that a 32 byte sample, which at a sampling rate of 64 kHz represents 4 ms of speech, would be retransmitted. A sample of 32 bytes is also not likely to make resynchronization at the receiving end difficult.

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 LAN's 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 the use of the channel mode by 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 the 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.

The ring access protocol has characteristics which are attractive for voice transmission. Because of the round-robin sharing of capacity, there is always a specified worst case upper bound on service time. Thus, voice transmission systems will tend to degrade rather than fail as traffic builds up. Because packets are short, the problems of packet loss, access and bridge delays can be dealt with at a fine grain and service times are more easily achieved. For networks which are data traffic-oriented but also support voice, it is likely that average network utilization will be low and thus a number of voice channels can be supported easily. Channel mode is well suited to video and other synchronous applications. The number of channel slots can be configured to match the video load at any particular time. Since video applications are unlikely to be bursty, changing the network configuration with traffic is attractive.

Addressing

Traditionally LAN's, whether rings or buses, 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 LAN's 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 the 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 [4], which has the bridge facility built in at the lowest levels and which 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 predetermined rule. This could involve a table lookup 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 minimized. 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 64 kbit 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 indefinitely. 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 unauthorized 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.

III. 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 4 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 are received correctly at the destination, the CRC is made bad by inverting the last 4 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 the

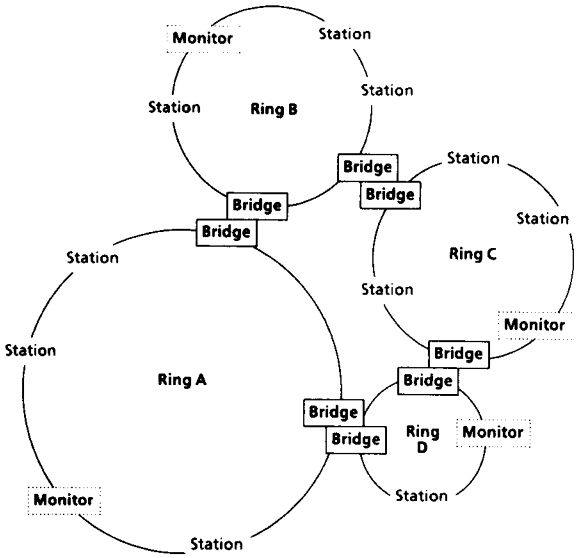

Fig. 1. CFR network architecture.

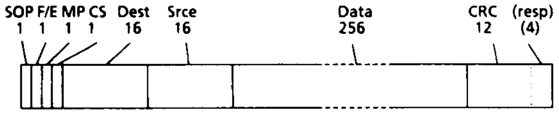

Fig. 2. CFR packet format.

TABLE I
CFR CRC/RESPONSES

	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

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.

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 summarized in Table I.

When a user wishes to transmit, a packet is loaded into the transmit FIFO of 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 the addresses on which the bridge potentially receives. The bridge maps (and hate lists at stations) are initialized 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 synchronize 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 empty the full/empty bits for all slots. The slot train sent by the monitor is head-to-tail contiguous, with the surplus delay around 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 localized 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.

Performance Under Load

It is unlikely that stations on a CFR are going to load the system with continuous transmissions of data. However, it is useful to consider the performance of the system under such conditions and to see what factors determine the maximum bandwidth available. The system bandwidth (sysBW) of a CFR is given by

$$\text{sysBW} = \frac{\text{number of data bits on ring}}{\text{total number of bits on ring}} \times \text{clocking rate}. \quad (1)$$

Thus, for a 100 MHz CFR with two slots and a 4 byte gap, the sysBW is 80 Mbits/s.

The point-to-point bandwidth (ppBW) is dependent on the number of slots because at best only one transmission from a single source can take place for each revolution of the ring. Thus, if N is the number of slots and Mc is the number of channel mode transmissions taking place, the channel mode ppBW is given by

$$\text{ppBW channel mode} = \frac{\text{sysBW}}{N} \qquad Mc \leq N. \quad (2)$$

A channel mode transmitter takes priority over normal mode (round-robin sharing). In normal mode, the transmitting station has to mark empty the returning slot and the minimum time between transmissions is $N + 1$ slot times. Thus, for a system with a short gap, the ppBW is given by

$$\text{ppBW normal mode} = \frac{\text{sysBW}}{N+1}. \qquad (3)$$

This assumes no other stations are using the ring. If Mn stations are transmitting in normal mode and Mc stations are transmitting in channel mode, the ppBW available to each normal mode station is given by

$$\text{ppBW normal mode} = \frac{\text{sysBW}}{Mn+1} \times \frac{N-Mc}{N}$$

$$Mn \geq (N-Mc), Mc \leq N \qquad (4)$$

$$\text{ppBW normal mode} = \frac{\text{sysBW}}{N-Mc+1} \times \frac{N-Mc}{N}$$

$$Mn \leq (N-Mc), Mc \leq N \qquad (5)$$

and the system bandwidth utilization (S) under such conditions is

$$S = \frac{Mc}{N} + \left\{\frac{Mn}{Mn+1}\right\} \times \left\{\frac{N-Mc}{N}\right\}$$

$$Mn \geq (N-Mc), Mc \leq N \qquad (6)$$

$$S = \frac{Mc}{N} + \left\{\frac{Mn}{N-Mc+1}\right\} \times \left\{\frac{N-Mc}{N}\right\}$$

$$Mn \leq (N-Mc), Mc \leq N. \qquad (7)$$

Fig. 3 shows graphs of the point-to-point bandwidth and line utilization for stations transmitting at full speed as a function of the number of slots. It can be seen that with channel mode and a single slot we can in theory achieve a ppBW of 80 Mbits/s. The system is then equivalent to a token ring. As the number of slots increases, each channel mode transmission has less and less bandwidth available. The ring then behaves as a multiple token system and several variable length packets can be transmitted at the same time.

Normal mode transmissions can at best attain half the maximum sysBW speed; however, even with only one slot several can be taking place at the same time. Where we mix channel and normal mode, the ppBW available to each channel mode user is higher than to each normal mode user. However, normal mode provides the low-level sharing and low-ring access delays suitable for control packets and voice samples.

The graph of sysBW shows how utilization improves as the number of users is increased and also how, during times of simultaneous channel mode and normal mode transmissions, ring utilizations can be high.

It is interesting to note that increasing the clock rate will improve the sysBW but not significantly change the ppBW on a ring of fixed size. The best way to maintain a high ppBW is to keep each ring small, use many bridges, and arrange that the time to copy a packet through a bridge is short.

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 8 bit wide data path. A diagram of a general hardware implementation of the CFR is shown in Fig. 4.

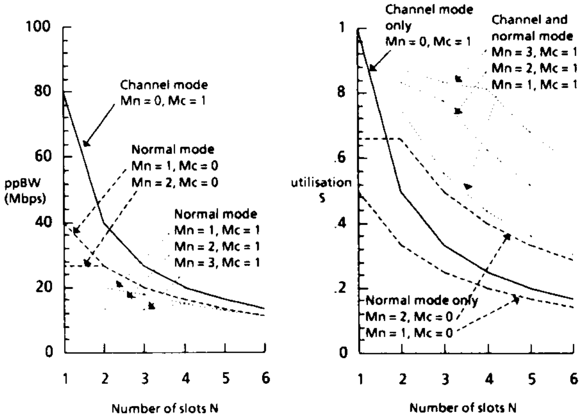

Fig. 3. Performance of the CFR.

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 implement a network controller which duplicates the logic of the repeater and provides a very low-cost integrated 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 meter 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. 5. This approach can be extended to use an 8 bit wide data path throughout. This system can then be used to produce a fast switch, which takes the form shown in Fig. 6.

The pin count constrains 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. 7. 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 architecture which allows many different application areas to be encompassed. The kit-of-parts can be used to make a fast parallel interprocessor switch at one extreme, or a slow and cheap network at the other. It can be used to build single-ring LAN's or by using bridges MAN's and other larger networks. The larger network can then be used to implement a reliable system by duplicating higher level services and functions thus

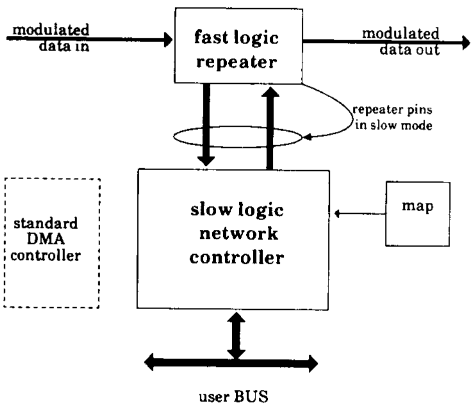

Fig. 4. CFR node structure.

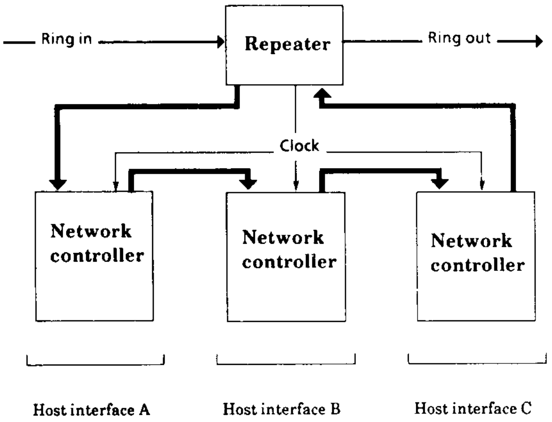

Fig. 5. CFR station cluster.

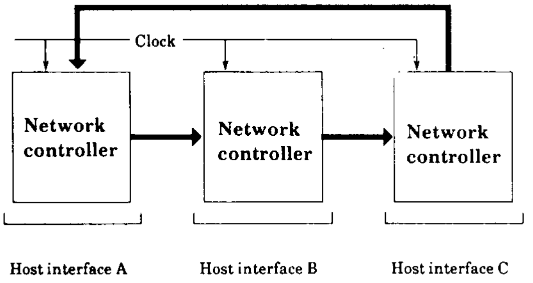

Fig. 6. CFR parallel ring switch.

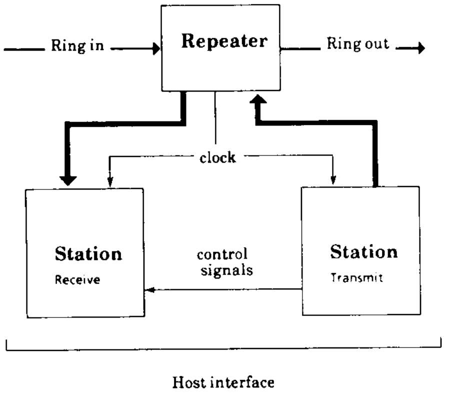

Fig. 7. CFR duplex station.

making failures of distant equipment or services less catastrophic to the user.

Initial implementations of the CFR 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 fiber optic cables. The slower links drive normal twisted pairs and operate at about 10 Mbits/s. This gives a typical point-to-point throughput of about 15 Mbits/s on the faster ring and about 2.5 Mbits/s on the slower rings. It is also possible to consider using a long distance duplex line operating a CFR structure as a link between centers tens of kilometers 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 ten 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 LAN's, 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 because 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 [7]. Alternatively the time for bridges to drop packets can be made short by setting the repeat packet counter 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 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 application 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.

IV. VLSI Implementation

A first VLSI implementation of the CFR has been developed using an ECL *repeater chip* and a *CMOS network controller chip* [6]. The ECL chip is a gate array of about 350 gates and the CMOS device is a custom design of about 25K transistors. The CMOS chip uses 8 bit data paths 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 delay through the repeater chip is 3 bytes and through the network controller chip it is 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. 8.

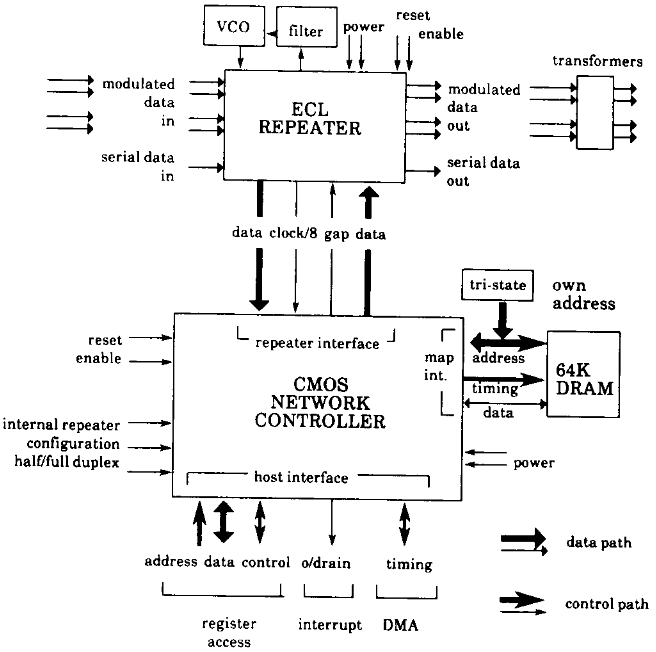

Fig. 8. CFR node components.

In the initial VLSI implementation, two simplifications have been made to reduce the size of the CMOS chip to give reasonable yields. First, 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 optimization of allowing the transmit buffer to be filled again as soon as a packet has left the outgoing side of a transmitting station providing the repeat count has been set to zero. Second, 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. Work is currently under way to implement some of these features and to shrink the network controller chip.

ECL Chip

This chip was designed using an ECL gate array with logic being performed at two internal voltage levels. This led to a compact circuit implementation and, by control of critical paths, a performance spread of 100–200 Mbits/s across the production spectrum has been achieved. A block diagram of the chip is shown in Fig. 9.

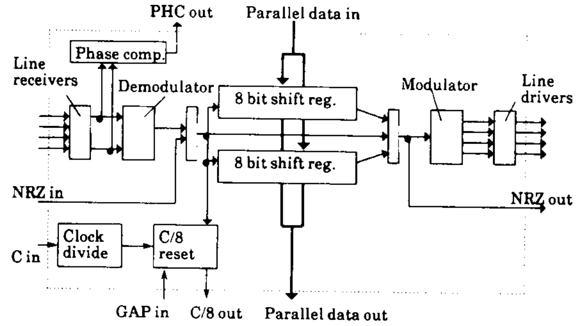

Fig. 9. ECL repeater logic and data paths.

The logic of the repeater chip consists of a demodulator, double-buffer shift registers, modulator, and associated data multiplexers.

This chip provides differential ECL line drivers and receivers for the Cambridge modulation system [1]. This modulation system uses two serial communication channels between nodes and if implemented using differential transmission techniques requires four wires. The scheme is well suited to twisted pair wire and provides a change on every clock. Up to about 100 ft of standard twisted pair wire has been driven at 100 Mbits/s with an error rate lower than 10^9. The ECL chip also provides (NRZ) serial data inputs and outputs which have been used with other transmission media such as fiber optics.

The ECL chip is designed to be clocked in a number of ways. If NRZ data are used, then an external clock must be supplied. If modulated data are used, it is possible to self-clock the chip using the phase comparison signal (PHC) although this is not recommended since jitter will not be controlled. The preferred method of clocking is with a phase-locked loop and is the method used in the prototype system. External circuitry is required to provide the VCO and loop filter. The approach used for modulation, line driving, and clocking in the CFR is similar to that used in previous Cambridge Ring systems and a more detailed analysis can be found in the literature [8].

The number of bits around the ring will be dictated by the length of the wire, the speed of transmission, and the number of stations on the ring. This will not necessarily be an exact multiple of eight and since a slot is 38 bytes.long, the odd bits will occur in the gap. Thus, the ECL chip must resynchronize 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 a phase of the C/8 clock during gap (GAP) and thus ensuring a C/8 clock half period is always at least 4 bits long.

Power consumption of the chip is about 1.5 W (worst case) and it is run from 0 and +5 V supplies. The ECL inputs and outputs are thus +5 V offset from their normal values but would normally not have to communicate with any other ECL systems. Outputs to slower logic are TTL compatible.

CMOS Chip

The CMOS chip was conservatively designed using single-metal, single-polysilicon, 3 μm technology and in this form operates at an equivalent line rate of 60 Mbits/s. It is designed for direct shrinking to 2 μm technology which will increase the equivalent serial speed to about 90 Mbits/s.

There are four broad types of signal emanating from the CMOS chip as shown in Fig. 8. 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 and to read the station address. 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 host interface which can be used in a number of ways. Communication between the host device and the network controller is achieved through a number of control registers. In addition, in order to allow high-speed DMA logics to use the registers rapidly, five hard lines are provided for manipulating data. A simple system can use a polling scheme by decoding the network controller somewhere in the address space. A more complex system may take advantage of the interrupt facilities and for high-performance, external DMA chips can be utilized. A description of the CFR registers and their functions can be found in the Appendix.

A block diagram of the logic and major data paths of the CMOS chip is shown in Fig. 10. The data paths around the chip are 8 bits wide and a number of tristate buses are utilized. Data are received from the repeater in the data-in latch and distributed to other logic as required. Outgoing data are collected in the data-out latch and passed back to the repeater chip. There are two CRC registers used for computing incoming and outgoing CRC's. The USER BUS is used for reading and writing registers by the attached (asynchronous) host. The MAP BUS is used to read and write the external DRAM map. It is also used to load the station address from external switches and to gate values to the select register and 8 bit wide comparator. The ADDRESS BUS is used both to gate values to the 8 bit wide comparator and to collect the outgoing data. To achieve this, it can be split into two separate sections. Since the present chip design does not function in channel mode, only single buffers are implemented for the incoming and outgoing data and addresses.

One of the most serious design problems in achieving high-chip data throughput was the synchronization of signals between the host and network clocks. To make sure the error rate is negligible, a delay of four C/8 clocks is required for all flip-flops to settle properly in the synchronization latches. This would be prohibitive when writing data and so to improve performance the data and address registers are not synchronized to the internal clock with every byte written. Only after a complete packet has been loaded, which in the 3 μm deign takes about 4 μs, does synchronization to the chip clock take place. Updating of the address registers is slower and is approximately equivalent to the loading of 8 bytes of data. Operations such as changing the select register, the map, or broadcast register take approximately one packet transmission time because the packet currently in the receive buffer has to be checked against the new values. The hold register is used for some of these slower operations (see Appendix).

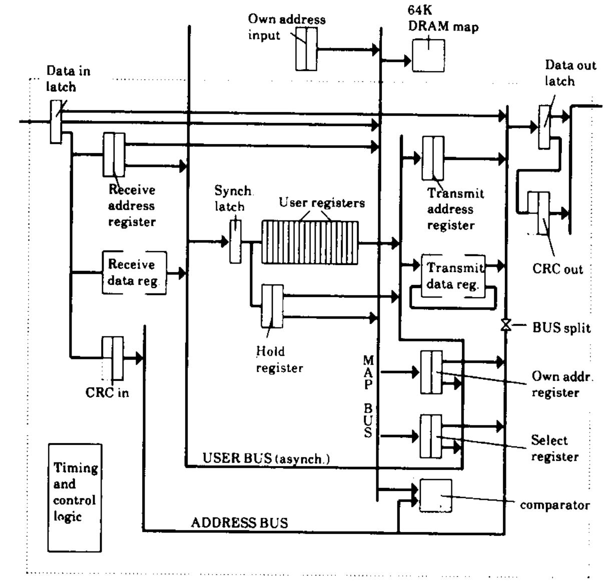

Fig. 10. CMOS network controller logic and data paths.

The layout of the CMOS network controller is shown in Fig. 11. The approach taken to the design of the chip was to automatically place and route subsections. It was found a suitable layout was only possible by then manually placing these blocks and strongly directing the autorouter which linked them. This meant that global routing was optimized and in particular buses operated at maximum speed. A standard cell approach was used for the subsections with most of the cells being specifically designed for this application. These included semi-dynamic FIFO packet buffers as well as a number of regular data path register cells. A postprocessor was used to push as much of the interconnect as possible into the top metal layer with the result that most tracks are predominantly in metal. Simulation was done at many levels from a high multiring level to the transistor level. All simulation and layout CAD tools were written in-house to enable tradeoffs between design and implementation to be made easily.

V. Conclusion

The design of the CFR is aimed at finding a solution to the problem of supporting both bursty traffic generated by

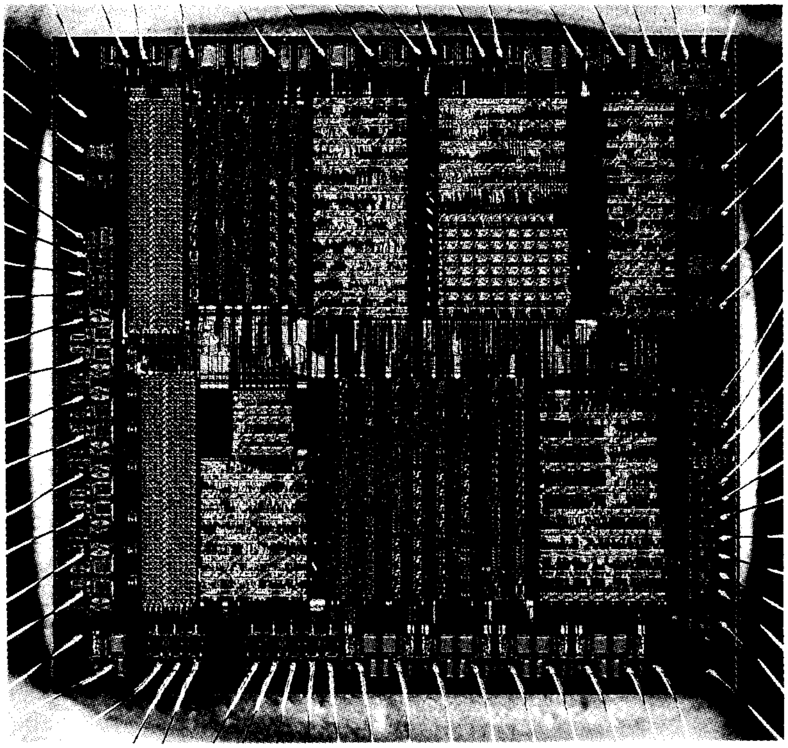

Fig. 11. CMOS network controller layout.

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 are being transmitted. A CFR implementation with a single slot used in channel mode makes the system equivalent to a token ring [9].

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.

It was particularly encouraging that the CAD tools were able to precisely predict the behavior of the VLSI components. This was true both at low levels where gate delays and high-speed effects are important and also at the higher levels where the implementation of a complex state machine is necessary. The chips worked first time to industrial tolerance criteria. 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 CFR's in many different applications.

There are many similarities between the CFR approach and the FDDI [10] proposals which use a token ring as the basic access mechanism. In the FDDI scheme, a low-priority station is made to pass on a token before it has cleared its transmit queue to provide finer bandwidth sharing. In the CFR, we start off from a position of fine sharing and allow a station to continue using a slot if it has enough data to send. The use of several slots in channel mode at the same time is a true multiple token system.

There are applications for rings operating at higher speeds than those currently achieved by the CFR [11]. The natural division of bandwidth by the empty slot principle is then even more desirable. It is an open research question how this bandwidth should be partitioned and what sort of performance guarantees and interfaces should be presented to the user.

APPENDIX

Reg	Read	Write
0	Receive Data Register (32)	Trańsmit Data Register (32)
1	Receive Address Register (2)	Transmit Address Register (2)
2	Interrupt Mask Register	Interrupt Mask Register
3	Interrupt Status Masked	
4	Interrupt Status Register	
5	Packet Transmitted	Top Up Transmit FIFO
6	Transmit Repeat Control Register	Transmit Repeat Control Register
7	Packet Received	Discard Receive FIFO
8	Broadcast Packet Received	Enable Reception of Broadcasts

APPENDIX (*Continued*)

Reg	*Read*	*Write*
9	Receiver Selected Everybody	Receiver Select Everybody
10	Receiver Selected Nobody	Receiver Select Nobody
11	Select Register Low	Receiver Next Auto Select
12	Select Register High	Receiver Select (from Hold Reg)
13		Write Map (addr. in Hold Reg)
14	Map Data	Read Map (addr. in Hold Reg)
15	Chip Configuration Register	Disable Chip
16	Slow Write Op. Complete	
17	Hold Register Low	Hold Register Low
18	Hold Register High	Hold Register High
19	Own Address Register Low	
20	Own Address Register High	
21		Monitor Slot Control Register
22		Monitor Run Mode Enable
23		Monitor Random Enable
24		Monitor Auto-Reframe Enable
25	Monitor Status Register	

Acknowledgment

We would like to acknowledge the help and encouragement of the many people involved in this project. T. Mann, T. Morris, J. Porter, and C. Stenton spent many months writing the CAD software and implementing the VLSI design. D. Wheeler and S. Temple provided continual critical comment of design decisions. The Unison collaborators were a source of encouragement at crucial moments. The project has received industrial support from Acorn Computers Ltd. and Olivetti Research Ltd.

References

[1] M. V. Wilkes and D. J. Wheeler, "The Cambridge digital communication ring," in *Proc. Local Area Commun. Network Symp.*, Boston, MA, May 1979.

[2] A. Hopper, S. Temple, and R. C. Williamson, *Local Area Network Design*. London, England: Addison-Wesley, 1986.

[3] R. M. Needham and A. J. Herbert, *The Cambridge Distributed Computing System*. London, England: Addison-Wesley, 1982.

[4] D. T. W. Sze, "A metropolitan area network," *IEEE J. Select. Areas Commun.*, vol. SAC-3, Nov. 1985.

[5] S. Temple, "The design of a ring communication network," Ph.D. dissertation, Univ. of Cambridge, Jan. 1984.

[6] Unison Project, CFR Chip Set Datasheets, Unison Doc. UA013, UA014.

[7] I. M. Leslie, "Extending the local area network," Ph.D. dissertation, Univ. of Cambridge, Feb. 1983.

[8] W. P. Sharp and A. R. Cash, "Cambridge Ring 82—Interface specifications," U.K. Science and Engineering Council, Sept. 1982.

[9] W. Bux, F. H. Closs, K. Kummerlel, H. J. Keller, and H. R. Mueller, "Architecture and design of a reliable token ring network," *IEEE J. Select. Areas Commun.*, vol. SAC-1, Nov. 1983.

[10] FDDI Token Ring Draft Proposals, ANSI X3T9.5, 29 Feb. 1985, July 15, 1986.

[11] L. A. Bergman and S. T. Eng, "A synchronous fiber optic ring local area network for multigigabit/s mixed traffic communication," *IEEE J. Select. Areas Commun.*, vol. SAC-3, Nov. 1985.

Andrew Hopper received the B.Sc. degree in computer technology from the University of Wales, University College of Swansea, in 1974 and the Ph.D. degree from the University of Cambridge, Cambridge, England, in 1978.

From 1977 he has worked at the University of Cambridge where he holds the position of Lecturer. His interests include local area networks, VLSI design, micro and multiprocessor design. He has had many links with industry as a director of Acorn Computers Ltd. and Qudos Ltd., and from 1986 as Director of the Olivetti Research Laboratory, Cambridge, England. He is also a member of the Olivetti Research Board.

Roger M. Needham (M'84) received the Ph.D. degree from Cambridge University, Cambridge, England, in 1961 and has, apart from sabbatical leave, been at the Computer Laboratory even since.

His research interests have included, successively, automatic classification, operating systems, time sharing systems, protection, and distributed and networked systems. Recent substantial projects include the Cambridge Distributed Computing System, the UNIVERSE project for interconnection of local area networks by satellite, and the system and application design of the Cambridge Fast Ring.

Dr. Needham was elected to the Royal Society in 1985.