A Multicast Fabric for Project Fairisle

Matthew Doar

January 1993

1 Introduction

This document describes the design and implementation details of a multicast fabric for the Fairisle network. The fabric provides the capability to replicate cells prior to their entry into the port controllers and the switch fabric.

The overall design is described in Sections 2 to 6. Details of particular aspects of the design are given in Section 7. Descriptions of interfaces, switch settings and pinouts appear in Sections 8 and 9. Some possible extensions of the designs are described in Section 10 and a rough guide to fault finding appears in Section 11.

2 Background

The extension of Fairisle switches to allow multicast connections to be assisted by replication of cells in hardware has been implemented using a multicast fabric placed in front of the existing unicast switch fabric. The idea of using such a switch to provide multicast capability has been suggested by various authors, notably Turner [1, 2], Hui [3] and locally by Peter Newman [4]. Other references to the work in this document appear in ISS'90 [5], The Allerton Conference [6] and my thesis [7].

Following the conventions of the authors cited above, this document uses the the terms 'unicast' and 'multicast' in preference to 'point-to-point' and 'multipoint'. The linked series of multicast nodes is referred to as a multicast fabric. Individual cells are either unicast or multicast and their degree of multicast is known as the 'fanout'.

3 Cell Structure

The cell structure is as given in Table 1. A pattern, configurable for each switch but with default 101, as the least significant bits of byte 1 (VCI) indicates that the cell has a multicast VCI. The top nibble of byte 1 is the fanout and bit 3 is used to turn the telemetry information on and off (see Section 7.5). It could also be used

as a priority bit for fairness control algorithms (see Section 10.1). Fanouts of 0000 or 0001 both considered to be a fanout of one in a multicast cell. Any cell without the multicast pattern is considered to be unicast cell.

Byte	Name	Comment
0	Soc	Start of Cell byte
1	VCI	VCI byte Multicast cells: Fanout (4), Telemetry (1), 101
2	VCI	VCI byte
3–4	SAR	SAR bytes
(5)	CRC	Possible CRC byte
5–52 (6–53)	Data	User data bytes

Table 1: Fairisle Cell Structure (modified CCITT)

4 Overall Design

The multicast fabric is constructed from a series of multicast nodes, as shown in Figure 1, one for each input port of the switch fabric which requires multicast capability. Each node is synchronised to its transmission board controller to receive data in a bit stream and to write it into the multicast board fifo (TxFifo). Each node can read data from TxFifo and write it to the ring using the clock distributed with the ring. The data is written from the ring into the fifo on the transmission board (RxFifo) at the required number of nodes.

The ring is closed with a fifo (Closefifo) in which enough bytes are stored to make the total length of the ring an integer number of slot lengths. Each slot is the same length as a cell.

Each node is placed after the data latch on the transmission board and before the fifo. An interface to the transmission board provides the synchronising clock, reset, a signal to say when the data in the latch is valid (dataNext - valid a half cycle before the data is latched) and the half-full flag from the fifo on the transmission board. The fifo write signal (prxfifoWbar) generated by the TAXI Xilinx chip is also passed to the multicast board for use in Global Unicast mode (see section 7.2). Power and Ground are supplied from the transmission board.

The delay between copies of the same cell is determined by the number of bytes latched at each node. In the implementation described later there are seven bytes stored per node, giving a delay between copies of 560ns with a 12.5MHz ring clock. Simulation results suggest that each cell will wait somewhere between 2 and 10 slot lengths to access the ring at reasonable loads, plus half a slot on average for the start of slot to reach the node in question. This is equivalent to a total delay of between 10.6μs and 44.5μs from arrival at the head of the queue in TxFifo to being written into the fifo on the transmission board.

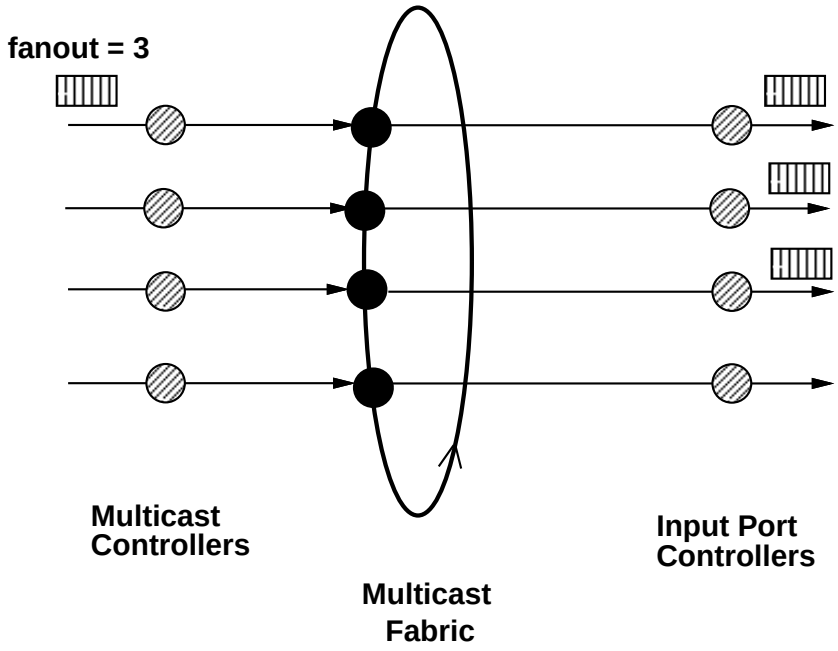

Figure 1: The Multicast Fabric Ring

Figure 2 shows the broad outline of a multicast node. Solid arrowheads indicate data, open ones indicate control. Solid boxes are latches, 9-bit wide, except for the ones on the ring data path which are 10 bits wide. TxFifo, CloseFifo and CloseLatch are the only items in the diagram not on the Xilinx chip.

5 Normal Operation

When data is read into the multicast fabric from the transmission board the fanout byte is copied forward into the Start of Cell (Soc) byte (byte 0) by the Shifter, and left intact in byte 1. If the cell is a multicast cell, then VCI0 in byte 1 is copied forward. If it is unicast, then zeroes are inserted into the Soc byte and the cell is not replicated.

The Start of Cell (Soc) is signalled on separate input lines (the control lines, here called DataIn[9..10]). The details are in the document "Proposed Fairisle/Yes/Maybe ATM Taxi Link Protocol" [8]. The Soc is used to start the writing to the TxFifo. Spurious Soc bytes after the first one in a cell are transmitted transparently for the port controller to deal with. Thus each node in multicast mode provides cells of guaranteed length (the bytes are counted into TxFifo) with an Soc as the first byte but possibly with other spurious Soc bytes within the data.

A slot on the ring is defined as being empty if the Soc byte, which is synchronised with the Start of Slot signal at each node, contains a fanout of zero.

When an empty slot arrives at a node where a cell is waiting, the cell is read out of the TxFifo and copied into the slot, as well as being passed onto the fifo on the

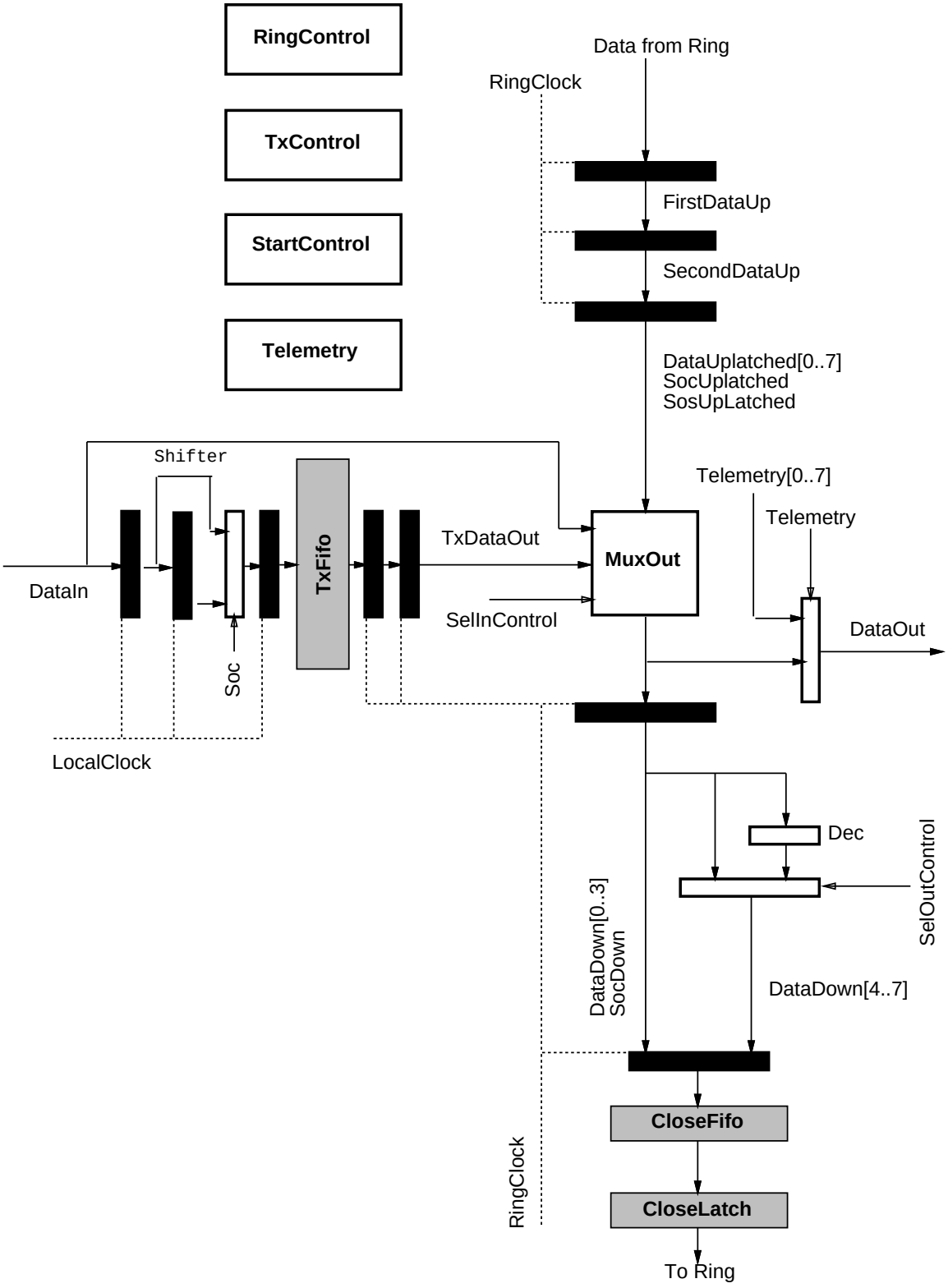

Figure 2: A Single Multicast Node

transmission board.

When a full slot arrives, its contents are copied to the output and passed onto the next node, with the fanout decremented by one, both in the Soc byte for the multicast board and in the VCI0 byte. So the VCI of each copy of the cell is different by one from its preceding one. Fanouts of zero remain at zero after decrementing.

6 Step by Step Descriptions of Events

References to signals not shown in the diagrams refer to signals in the *.hdl* files. For the current details of strobe generation, closing the ring and telemetry, see the files indicated in the boxes in Figure 2.

6.1 Reading a Cell into TxFifo

Assume GlobalUnicast is LOW and DenyAccess is LOW

1. Ignore data until an Soc signal is seen.
2. When the first Soc has been latched, enable strobing the data into TxFifo, to start three cycles later.
3. On the second byte being latched, SocB moves the fanout byte forward into the Soc byte, ready to be written to TxFifo by WriteTxFifobar. This is so that the first byte on the ring will contain enough information for each node to decide whether to take a copy of the cell on the ring or to leave it.
4. If dataNext (aka DataValid) goes low during a cell, this is either due to no data being present or a CRC byte being present. Ensure that writing is disabled for the next byte.
5. Back-pressure from the TxFifo is that the cells already in it are counted and when MaxTx is reached, no more are accepted until one has been read out (see page 17–15).
6. The end of a cell is detected by counting 53 bytes and then resetting the signal WithinCell, unless there is another cell arriving, as indicated by Blank.

6.2 Start of Slot Arrives

GlobalUnicast is assumed LOW.

Define the rising edge of the Start of Slot (Sos) as being when it is first latched from the node 'upstream'. Label this edge zero and count edges (up or down) from it. If the slot is empty (FreeSlot is HIGH), then any cells waiting in the TxFifo can access the ring for this revolution, provided that DenyAccess is LOW.

If FreeSlot is HIGH and CellWaiting is HIGH

1. The existing data in the TxFifo output latches is written into RxFifo. The read strobe goes low on edge 1, so that the data appears sometime after edge 2, is latched by the read strobe and is then passed to the node using latches clocked on edge 4. A typical delay on the read strobe is 40ns.

2. Data is strobed out of TxFifo until the next Sos.

3. SelInControl is latched HIGH on edge 4.

4. Writing to the fifo on the transmission board begins with the write strobe going low at edge 4 and high at edge 5. The delays of typically 40ns on both data and strobe mean that the data is actually written at about edge 6 with a 12.5MHz ring clock.

If FreeSlot is LOW, Soc is HIGH and the multicast pattern is present in Soc

1. The data latched from the ring upstream is written into RxFifo just as above. Back-pressure comes from HalfRxfifobar, which prevents more than 3 cells accumulating in the RxFifo on the transmission board (for an AM720035, allowing for 'fencepost errors').

2. SelOut is set LOW at edge 4 for the SOC and VCI0 bytes to decrement the fanout nibble.

7 Design Details

7.1 Closing the Ring

Each multicast board has the capability to act as the control node, which generates the first and subsequent Start of Slot (Sos) signals after reset and contains and controls the fifo which is used to close the ring. The other nodes are slave nodes and just pass the Sos signal onto the next node downstream.

A counter on the control node starts after reset (which is global to the multicast fabric) and upon reaching a value *53 - 7*NoNodes - 2)* determined by the setting of the NoNodes (number of nodes in the ring) on the board and the (fixed) length of the cell, generates an Sos. After the first Sos has been generated, when the ring fifo (also known as Closefifo) has been filled with the correct number of null bytes to make up a cell length, subsequent Sos signals are generated using the fixed length of a cell, which is embedded in the comparator circuits in *startcontrol.hdl*. The incoming Sos to the control node is used to permit cells waiting to gain access to the ring and is then discarded, since the new Sos will be generated there.

The signal defining whether a node is the control node is Startup, switch 7 on the dipswitch, active high. The number of nodes in the ring (NoNodes) is defined by switches 3 to 6, 3 being msb of the nibble. 0000 is a single node ring, through to 0011 being a four node ring. Other values are undefined.

The control node also acts as a distribution point for the ringclock. which has been tested at frequencies between 5 and 12.5 MHz inclusive. Of course, a full

12.5MHz will act as a funnel for slower speed traffic from all the transmission boards and can overload a port controller.

Note that with a small number of nodes on the ring, such as three or four, there is a substantial delay before data written into the fifo is seen at the next node. A handy check of ring/Sos cycle lengths is that SosDown from the control node should appear in the same cycle as SocDown from the CloseFifo latches.

7.2 Global Unicast

One major mode of operation is that of **Global Unicast**, in which all data is multiplexed directly from the receiving latch on the transmission board to the fifo on the transmission board and written into the fifo using the signal generated by the transmission board Xilinx chip (prxfifoWbar). Everything is handled as transparently as possible, with the only delay being from passing the data and strobe through a 3:1 multiplexer (MuxOut). This mode is supposed to imitate the performance of a transmission board when there is no node attached.

The Global Unicast setting on the multicast board should be the same for all the multicast nodes attached to a switch and is controlled by switch 8 of the dipswitch, active high, or the switch on the front panel of the multicast board, active up for unicast.

7.3 Global Unicast – Description of Events

GlobalUnicast is HIGH for all nodes.

1. The data into TxFifo is multiplexed directly to the data outputs.
2. The fifo on the transmission board (RxFifo) is written into using prxfifoWbar from the transmission board Xilinx chip.
3. The ring operates as usual, with whatever data is on it continuing to circulate. Fanouts are still decremented whether a copy is read from the ring or not.
4. DenyAccess has no effect. MaxTxbar is never reached since the bytes and cells are never counted. Hence CellWaiting is never true and neither MissedSlot nor CellAccess ever occur. No strobes for reading or writing fifos are passed to the fifos.

7.4 Reset State

When the nodes are reset, a signal is generated to clear the SocDown bit in the Soc byte so that spurious data from power-up will not be read from the ring at nodes below the current one as valid data, which would fill the fifo on the transmission board and may stop the boot sequence. This signal, ZeroRing, is also used to ensure that the FreeSlot signal seen is alway HIGH until the node has either received a multicast cell from above or (if the node is the control node) one from the transmission board.

This means that each slave node passes received cells straight through, with no replication, so long as Sos pulses are arriving. The ring is only enabled for multicast when a multicast cell is received at the control node. The transmission of this cell passes a multicast Soc byte down the ring, which enables multicast operation in all the other nodes.

7.5 Telemetry

For experimental purposes, a means of extracting performance information from each node has been added to the original design. At the time that the decision is made that a waiting cell should access the ring, the states of the counter of cells in TxFifo and the counter which registers slots missed are latched and a counter is started. When the fifth byte has passed, a control signal, Telemetry, is used to multiplex the state of these counters into the first byte of the cell's data, byte 6. Bits 0 to 3 are the count of the cells in TxFifo at the time of access to the ring, which is always greater than 0. A count of zero indicates that MaxTx has been set and cells have been lost. Bits 4 to 6 are the count of the number of times the cell failed to gain access to the ring. Bit 7 is set if this number is greater than can be represented by bits 4 to 6. The telemetry data is inserted into the cell user data of the copy which appears from the port at which the source cell arrived. Data on the ring is unaffected, so each cell is only monitored once.

This may have been altered so that the data byte 0 is as before except that the top nibble is zero and byte 1 has an unsigned 7 bit count of the number of missed slots seen. The eighth bit set still indicates an overflow.

The multiplexing of this data can be turned on and off by bit 3 in VCI0. The control point is as the cell is written into TxFifo. So the telemetry information begins upon the transmission of the first cell in the fifo when the activating cell's VCI0 byte arrived and ends upon the reception of a cell with bit 3 unset. Thus no more than MaxTx cells do not receive telemetry information between start and stoppage of telemetry.

7.6 Latencies

Raw delays across Xilinx chips have been measured at about 10ns. This is what the Global Unicast mode comes close to, going through just one CLB. Typical delays for this design using a 3042PG132-70 Xilinx chip are 40ns on most data lines. A clock speed of 12.5MHz means that the cycle time is 80ns, and the mark/space ratio is usually 3/5 after buffering etc. The read and write pulse width timings are theoretically violated by the reduced m/s ratio for -35 fifos, but they seem to work happily enough. Really ought to use -25 fifos.

Designs which work at 8MHz were obtained using just three set of latches, two above the output, to gain time to make the read/write decision. To work at 12.5MHz, another two sets had to be added, one below the output to make sure that the data down to the ring reached it within the data setup time (10ns) and another set above the output to permit even an even earlier start of the read strobe.

The essential aspect of this design was to work backwards from when the data had to be present at the outputs and add stages of pipeline as necessary. Depending upon a signal which had been though more than one or two CLBs without being latched to a known state also tended to create problems (some place and route runs produce working configurations, some don't).

7.7 Contiguous Cells

As noted above, when contiguous cells arrive, the latch which registers that the writetx strobe is within a cell must not be reset.

Reading cells from the TxFifo is not a problem since the starting and stopping depends upon the latched Sos, which is pipelined sufficiently to restart reading from TxFifo and writing to the transmission board.

An observation here is that counters should be clocked later rather than earlier, so that the reset for the count, wherever it comes from, can be removed before the next count clock edge arrives. This is due to the asynchronous nature of the Xilinx CLBs direct reset signal.

8 Implementation

8.1 HDL files

The .hdl files for the design currently reside in

`/groups/fairisle/designs/multicast/telemetry`.

The file node.hdl is the overall file, which is called by mission.hdl. The file nodexilinx.hdl contains the Xilinx chip definition.

Other files include:

counters.hdl Various 3, 4 (up/down), 6 bit counters and others.

dec.hdl Combinational logic to decrement the fanout.

iobuffers.hdl The fanout shifter.

mymuxes.hdl Multiplexers.

nodebits.hdl Some mock ttl definitions e.g. dipswitch and the necessary Xilinx chip description.

ringcontrol.hdl Control of access to the ring, ReadTxFifobar and WriteRxfifobar are generated here

startcontrol.hdl Control of closing the ring and generating the Sos.

telemetry.hdl The gates for the telemetry information.

txcontrol.hdl Control of writing complete cells into TxFifo.

All other files are either part of the test-rig e.g. mf.hdl or are part of the transmission board work by Mark Hayter and Daniel Gordon.

8.2 PCB design

The PCB was designed using the Racal CADSTAR system (which explains the many six and eight letter names on the floorplan in Figure 3) and designed to sit beside the transmission board, which is in turn mounted above a port controller full-height VME card. The multicast board is joined to the transmission board by links in the areas marked 'taxi' and can also be secured to it by a post, as marked. The holes VMEHL1 and VMEHL2 are for a front to attach the card directly to a VME crate.

The links OUTLK0 to OUTLK9 are for jumpers to connect the outputs DataDownLatched[0..7], SocDownLatched, SosDownLatched respectively from the Xilinx chip to either Closefifo (ClseFo) if they are to the left or directly to the ribbon cable attached to EDGCN2, if they are placed to the right. This permits a board to be configured as a control node by switching the jumpers, installing ClseFo and ClseLc (a 9-bit latch) and setting the dipswitch appropriately.

EDGCN1 brings data from upstream on the ring. Seen from the component side of the board and looking from the bottom left hand corner to VMEHL2 , the upper layer of pins carry the signals DataUp[0..7], SocUp, SosUp respectively.

Working multicast fabrics have been built using ribbon cable up to about 40cm long at 12.5MHz.

8.2.1 Interface to Transmission Board

This consists of a block of 27 pins arranged in three columns of 10, 10 and 7 pins. Looking a the board from the component side with Closefifo at the top, the first column is the data out to the transmission board, Taxi0 to 3 and 6 to 9 being DataOut[0] to DataOut[7] respectively. Taxi4 is SocOut and Taxi5 is WriteRxfifobar. The next column is the data in from the transmission board, as for the data out, but with Taxi4 as

TaxiM0 is ground, TaxiM5 is Vcc, TaxiP0 is the reset signal, TaxiP1 is the local clock from the transmission board, TaxiP2 is connected to the multicast led on the front of the transmission board, but unused so held low. TaxiP5 is DataValid and TaxiP7 is the half-full signal from the fifo on the transmission board (HalfRxfifobar).

8.2.2 Chip Orientation

All orientation guidelines are viewing the board from the component side, as in Figure 3.

Chip	**Pin 1**
NodeXi	Bottom left
TxFifo	Bottom right

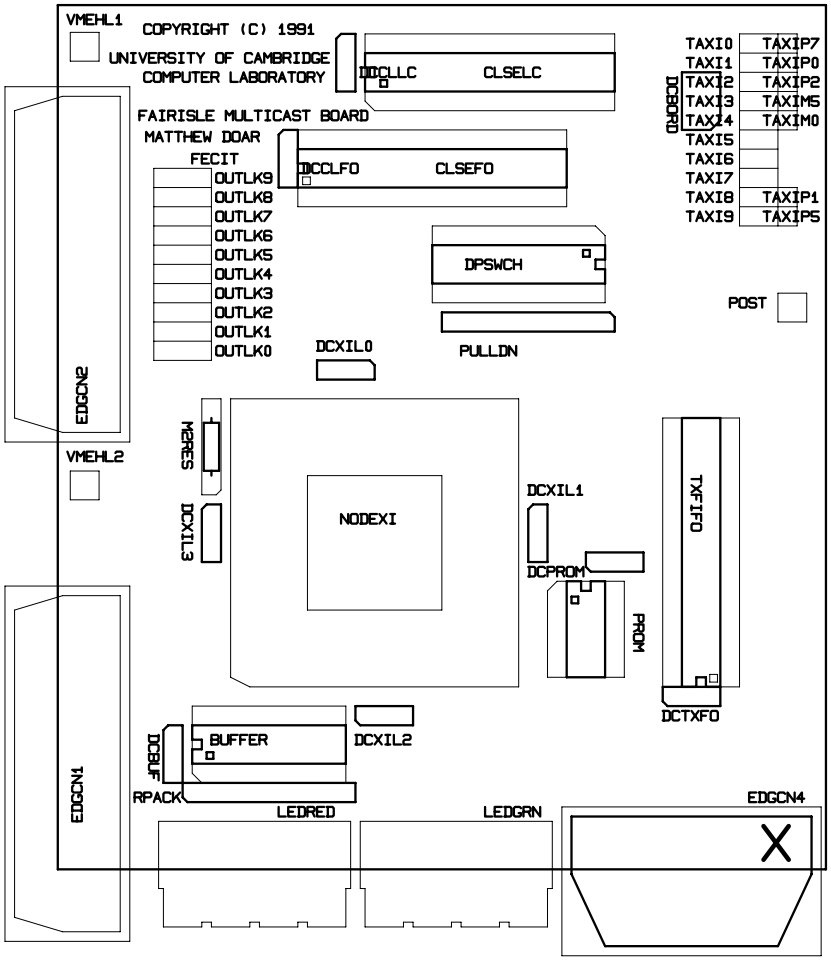

Figure 3: Floorplan of the Multicast PCB

ClseFo	Bottom left
ClseLc	Bottom left
Buffer	Bottom left
XiProm	Top left

9 Interfaces

Figure 4 gives the signal interface between the transmission board and the multicast boards and other multicast boards.

9.1 Hardware Patches

1. Fifo patch to correct floating expansion and retransmit inputs due to incorrect QUDOS model. Connect pin 14 to pin 7 and pin 28 to 23 on all fifos.

2. The HalfClosefifobar input to the Xilinx chip is actually connected to the FullClosefifobar output on Closefifo. Cut the track near pin 8 (FF) of Close

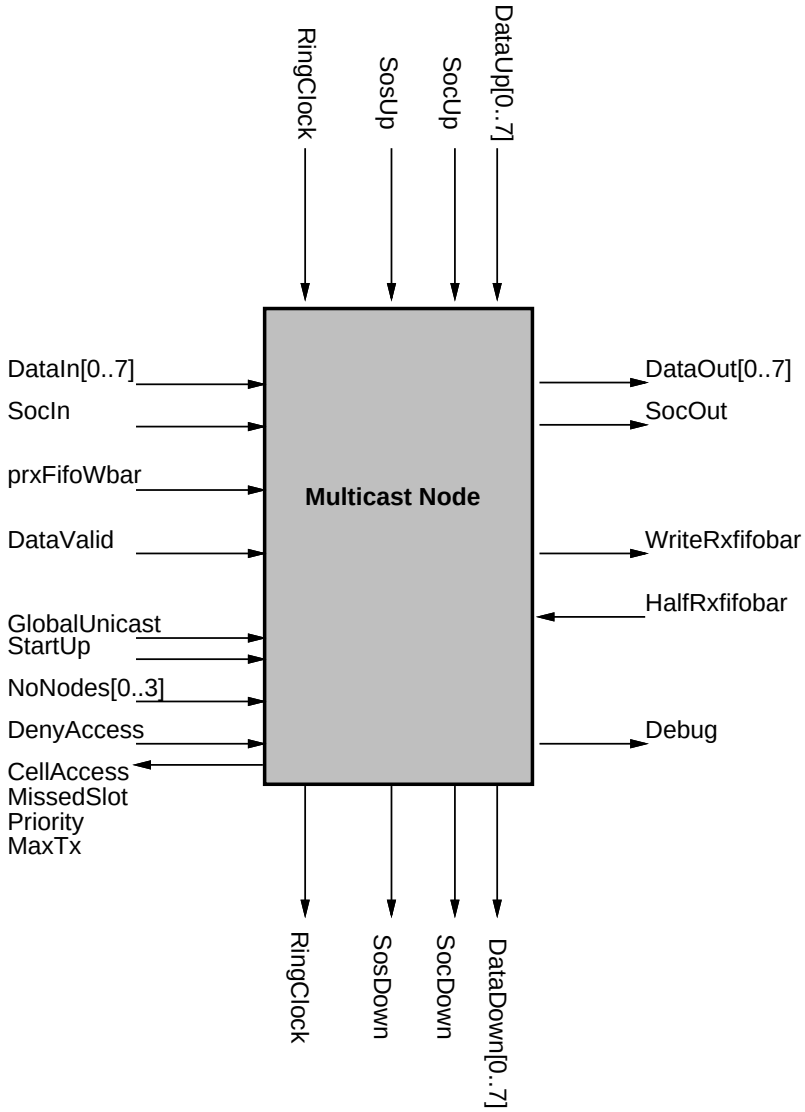

Figure 4: Interface to the Multicast Node

fifo and connect the track to pin 20 (HF). This is unlikely to ever be necessary and has not been done.

3. The track from pin 5 of the led buffer (an AC04) should be cut before it branches and pin 5 joined to N7 on the Xilinx chip. This ensures that the WriteRx led is driven from the signal generated by the Xilinx chip and not by the signal received as prxfifowbar from the transmission board. This has not been done.

4. The major patch is to distribute the clock for the ring (yellow wires).
   - The track from P13 on the Xilinx chip to pin 13 of TxFifo is cut on the component side.

- A buffer is made from an ACO4 hex inverter, connecting pins 2 and 3 and using pin 1 as the input and pin 4 as the output. This chip is piggy-backed on the ledbuf ACO4.
- P13 on the Xilinx chip is connected to the output of the buffer and the input of the buffer is connected to the jackplug on the front panel.
- Connect pin M12 to pin 13 of TxFifo.
- For one board, usually the control board, connect C3 on the Xilinx chip to the jack-plug.

5. The ring clock is now derived from a separate crystal mounted on the control board to enable the ring to be slowed down for the purposes of testing. It is distributed using buffers exactly as before.

9.2 Design Hazards

1. The signal (CellAccess) which decrements the count of the number of cells in TxFifo had to be altered so that if the ring was much slower than the receiving speed, the count didn't get decremented multiple times.

9.3 Debugging

Four red and four green leds show the state of various signals as listed below. Led numbers are from the left as seen from the component side of the board with the leds at the bottom.

ReadTxFifo Red 1. Alight indicates reading from the fifo that receives the data from the transmission board.

WriteTxFifo Red 2. Alight indicates writing to the fifo that receives the data from the transmission board.

WriteRxfifo Red 3. Alight indicates writing the fifo that receives the data from the multicast ring, but see the patch above.

Red 4 Power on.

MissedSlot Green 1. Alight indicates that a cell waiting in TxFifo has missed access to the ring.

CellAccess Green 2. Alight indicates that a cell waiting in TxFifo has gained access to the ring.

MaxTx Green 3. Alight indicates that the Tx fifo is full of cells and that any further cells arriving will be discarded.

Green 4 Power on.

9.4 Xilinx Pinout

DebugBuf	N14	NoNodesBuf< 1 >	D2
NoNodesBuf< 2 >	D1	NoNodesBuf< 3 >	E2
NoNodesBuf< 0 >	F3	MaxTxBuf	E1
SocInBuf	K12	DataInBuf< 0 >	K14
DataValidBuf	G2	DataInBuf< 2 >	J14
DataInBuf< 1 >	K13	DataInBuf< 3 >	J13
DataInBuf< 4 >	H14	DataInBuf< 5 >	H13
DataInBuf< 6 >	G14	DataInBuf< 7 >	G13
GUBuf	K3	DataIntoTxBuf< 7 >	M1
DataIntoTxBuf< 6 >	N1	DataIntoTxBuf< 5 >	M2
WriteTxBuf	M3	DataIntoTxBuf< 4 >	P2
DataIntoTxBuf< 2 >	N4	DataIntoTxBuf< 3 >	P3
DataIntoTxBuf< 1 >	P4	DataIntoTxBuf< 0 >	N5
SocIntoTxBuf	P6	prxfifoWbarBuf	M9
TxFifoSetBuf< 7 >	P7	TxFifoSetBuf< 6 >	N8
TxFifoSetBuf< 5 >	N9	WriteRxBuf	N7
TxFifoSetBuf< 4 >	P10	TxFifoSetBuf< 3 >	N10
HalfRxBuf	M10	TxFifoSetBuf< 2 >	N11
TxFifoSetBuf< 1 >	P12	TxFifoSetBuf< 0 >	N12
TxSocSetBuf	M12	DataDownBuf< 1 >	B14
DataDownBuf< 0 >	C14	SosDownBuf	E12
StartupBuf	D13	HalfClosefifoBuf	D14
WriteCloseBuf	F12	PriorityBuf	E14
DenyAccessBuf	F13	CellAccessBuf	F14
DataOutBuf< 7 >	L2	DataOutBuf< 6 >	L1
DataOutBuf< 4 >	K1	DataOutBuf< 5 >	K2
DataOutBuf< 2 >	H2	DataOutBuf< 3 >	J2
DataOutBuf< 0 >	G1	MissedSlotBuf	J12
DataOutBuf< 1 >	H1	SocOutBuf	F2
ReadTxBuf	M13	SocUpBuf	B4
ReadCloseBuf	C5	SosUpBuf	A4
DataUpBuf< 7 >	B5	EmptyClosefifoBuf	C6
DataUpBuf< 0 >	A5	DataUpBuf< 1 >	B6
DataUpBuf< 2 >	A6	DataUpBuf< 3 >	B7
DataUpBuf< 4 >	A7	DataUpBuf< 5 >	B8
DataUpBuf< 6 >	A8	SocDownBuf	A9
DataDownBuf< 7 >	B9	DataDownBuf< 6 >	A10
DataDownBuf< 5 >	B10	DataDownBuf< 4 >	B11
DataDownBuf< 3 >	B12	DataDownBuf< 2 >	C12

10 Possible Extensions of the Design

10.1 Fairness Control

The design as originally presented in [4] is inherently unfair in the manner in which individual nodes access the ring. Simulations [6] have suggested that the design can tolerate moderate imbalances of traffic at each port, especially at low loads.

Four outputs and one input to each node have been provided in the design to permit a possible implementation in the future of a control mechanism (distributed or centralised) to improve the access fairness to the ring. In order as they appear on the edge connector, looking at the socket from the component side of the board, right to left, on the top row, they are:

- **MissedSlot**
  A cycle pulse active high in the low phase of the clock, occurring when there is at least one cell waiting and it has just been denied access to the ring, either because of another cell from upstream or since DenyAccess is set.

- **CellAccess**
  A cycle-length pulse active low from the rising edge of the clock when a waiting cell gains access to the ring.

- **Priority**
  Intended to give some idea of the current state of the cells in the fifo. Could be set by the count of high priority cells arrived being non-zero and cleared by the count becoming zero, since it is decremented every time a high priority cell gets access to the ring. Currently indicates whether a cell is waiting in TxFifo.

- **MaxTx**
  Actually MaxTxBar. Active low, when the number of complete cells read into TxFifo reaches a number hard-wired in *txcontrol.hdl*. The AM7200(35) can hold 4 complete cells, but since MaxTx changes at the end of a cell's acceptance into TxFifo and CellAccess alters the number of cells counted at the start of a cell's transmission, the maximum number of cells is one less, i.e. 3. Since the decision to read the next cell may be taken before the count is incremented from the last cell read in, contiguous cells would mean that almost another complete cell could be read in. So as a safety margin, the number hard-wired in should be two cells less. Similarly, the AM7201 can hold a value of 8 cells and the AM7202 16 cells, with values of MaxTx set at 6 and 14 cells respectively.

- **DenyAccess**
  When this signal is set during the cycle that Sos is latched, any waiting cell is denied access to the ring and remains intact in TxFifo.

Other operations, such as removing the cell at the head of the line, could be

implemented by reconfiguring the Xilinx chip so that the five i/o pins mentioned above are reused.

10.2 Control Algorithms

Two designs of control mechanisms have been considered, either of which could be implemented on a contra-rotating ring. The first design considers that the states of all the nodes would be known at any time (centralised), the other that only a subset of the states would be 'seen' by any one node to make its control decisions (distributed). The major factor in the latter case is the size of the subset of nodes.

Some work has already been done on simulating the performance of various control mechanisms [6].

11 Problem Shooting

Points which should be carefully checked for during measurement and fault finding.

- Check power-on leds on transmission board (bottom green) and multicast board (red and green 4).
- Check secure interfaces to transmission board (the front-panel bolts can separate the transmission board from the port controller) and the interface to the multicast board is aligned correctly (be aware that power and ground can be shorted here!)
- Check that the chip probe clips have not dislodged the fifos, since they are longer and thinner than usual.
- Check that none of the fifo legs have been bent by the mismatched sockets.
- Check the position of the unicast switches (they are or'ed together). The front-panel one is unicast up, the dipswitch (8) is also unicast up.
- Does GlobalUnicast work as expected? This is often a good way to check that the problem is with the multicast part of the switch, not anywhere else. Turn all the GU switches to up and reboot. Turn them back to multicast one at a time, from the first portc below the control node.
- Check that the Sos signal is being generated and reaching all nodes. Either probe the SosDown signal (either outer pin of OUTLK9) or the Debug pin (N14) on the Xilinx chip, which gives FirstSosUp, the first latched version of Sos inside the Xilinx chip.
- Check the Soc going into TxFifo together with the WriteTxFifobar.
- Check the read and write sequence when the next Sos appears.

- Check the time between Sos pulses and that the length of the read/write strobes is correct as well. If the data and Soc appears to be a cycle behind the Sos, the ring may be a byte too long.
- Check whether the MaxTx led has been set.
- Is the fanout decremented properly in the ring?
- Try running the control node at 8MHz.
- The sequence of booting nodes has to be firstly the gateway, then reset all the others, try to load all the kernels roughly together. If a node begins to multicast cells to other nodes, they will not be able to boot, since they will have what they see as spurious information in their rxfifos.

12 Measurements

Some measurements which are likely to be of interest are given here. Averages can be taken per ring revolution, per input cell, per call, per input port. Telemetry could take place through the Xilinx teaching board but seems easier using the method described in Section 7.5.

- Average delay across the switch with and without the multicast nodes in operation.
- Average queue size in TxFifo.
- Signs of unfairness affecting the performance of the ring for an individual node, e.g. maximum difference in mean delays at each node.
- Minimum acceptable size of the fifos for some predefined standards of service.
- Maximum acceptable size of the fifos for some predefined standards of service.

References

[1] J S Turner, *Design of a Broadcast Packet Switching Network*, IEEE Trans. Commun., Vol.36, No.6, June, 1988, pp.734-743

[2] R G Bubenik and J S Turner, *Performance of a Broadcast Packet Switch*, IEEE Trans. Commun., Vol.37, No.1, Jan., 1989, pp.60-69

[3] J Y Hui, *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer Academic Publishers, Norwell MA 02061, 1990

[4] P Newman and M Doar, *A Slotted Ring Copy Fabric for a Multicast Fast Packet Switch*, Proc. Int. Switching Symp. 1990 (ISS '90), Stockholm, Sweden, May, 1990

[5] *A Slotted Ring Copy Fabric for a Multicast Fast Packet Switch*, P Newman and M Doar, Proceedings of International Switching Symposium 1990, Stockholm.

[6] M Doar and N G Bean, *Multicast in ATM Switches*, accepted for The Thirtieth Annual Allerton Conference, University of Illinois at Urbana-Champaign, 1992.

[7] M Doar, *Multicast in the Asynchronous Transfer Mode Environment*, PhD Thesis, University of Cambridge Computer Laboratory, 1993.

[8] M D Hayter, *Proposed Fairisle/Yes/Maybe ATM Taxi Link Protocol*, October 1991.