Network Cards for the Pandora Multimedia System

DJ Clarke, Olivetti Research Limited
GJ Stark, Advanced Telecommunication Modules Ltd

ORL Technical Report 94-5

Contents

1 Introduction

1.1 The Pandora System

In 1990 Olivetti Research Limited in Cambridge began to deploy a network of Multimedia workstations known as Pandora. Each Workstation consisted of a computer running Unix (an ARM-based machine from Acorn Computer) and an add-on box (the 'Pandora Box') to handle video and audio traffic. The video and audio was carried on the local 50Mbit/s network, the Cambridge Fast Ring (CFR) [3]. This remarkable set-up allowed multiple streams of video and audio to be sent concurrently between the workstations and, later, a fileserver machine as well. The whole was integrated so that a user could manipulate the video and audio using familiar-looking X-Windows tools, with the video appearing in one or more windows on the Workstation screen. [6] [7]

The Pandora Box is a MIMD (Multiple Instruction Multiple Data) multiprocessor implemented using Inmos Transputers. [11] The architecture of the Box is outlined below in figure 1.

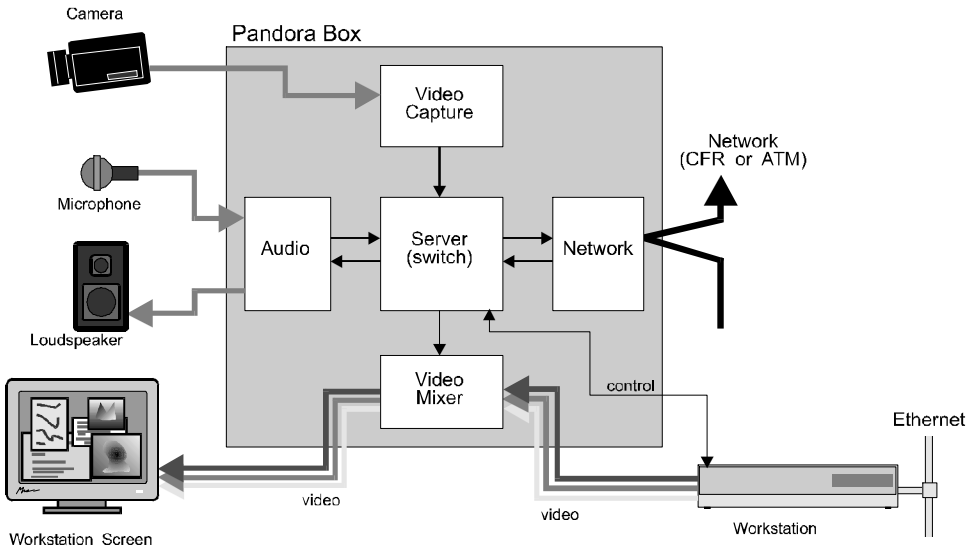

Figure 1: Architecture of a Pandora Box

The Inmos Transputer has high-speed serial links built-in. In Pandora this feature is used to pass control messages and data between processors. There are also some extra datapaths in the form of FIFO buffers to handle video traffic.

The earliest implementation of the Pandora Box did not have a separate Network Interface. The Server Transputer was used to connect the other subsystems together and also to handle the Cambridge Fast Ring (CFR) directly. It was clear that the Server was the first processor to saturate when the Box was heavily loaded; the solution was to design a new card to handle the network. This would leave the Server free to act as a data packet switch.

1.2 The Pandora Network Cards

The Network card has evolved through several iterations, but an architecture of two closely coupled processors has been retained throughout.

The first version of the Network card, which had only 64k bytes of memory, was replaced by a second version with 2M bytes. A cost and size reduction on the ‘Pandora Box’ then produced ‘Pandora lA’ which is mechanically better but retains much the same architecture. The latest version of the Network card involved replacing the Cambridge Fast Ring interface with an Asynchronous Transfer Mode (ATM) [8] interface.

1.3 The Firmware

The firmware was all written in Occam, [5] using the Transputer Development System, [12] and later on, the Occam Toolset, all from Inmos Ltd. Occam provides

a secure means of coding multiple parallel processes which use message-passing to communicate. The language's communication primitives work equally well between processes running on the same processor and also between processes on different connected processors. Despite this, code is compact. It is possible to run a useful piece of firmware with many processes in the 4 kilobyte on-chip memory of a Transputer. Occam on the Transputer features fast (sub-microsecond) context-switching which makes it very suitable for the time-critical tasks involved in handling many streams of live audio and video.

2 Hardware Overview

2.1 Cambridge Fast Ring Chipset

The Cambridge Fast Ring (CFR) was a 'Slotted Ring', that is, a closed loop of wiring around which a framing structure constantly circulates. Data was transmitted in cells (Then known as 'Minipackets'). The data portion of each cell was fixed at 32 bytes, with a header of 4 bytes. At Olivetti Research, the CFR was finally taken out of service at the end of 1993, having been in daily use since the laboratory's founding in 1986. It is now replaced by an ATM network.

The CFR silicon started life as a Ferranti ULA (Uncomitted Logic Array) design known as 'The Ring Chip'. Some prototypes were made, but the process was withdrawn by Ferranti and work had to start afresh. The final design involved two custom LSI components: an ECL Serialiser chip and a CMOS Station chip. Data was carried between stations serially on shielded twisted pair cable.

The Serialiser chip was designed on a Plessey ECL gate-array, and worked well. It handled the NRZ serial line coding and was connected to the CMOS Station chip by a pair of byte-wide busses.
The CMOS chip was designed at the Cambridge Computer Laboratory; the designers followed the design route which was usual at the time; namely handcrafted full custom with support from home-grown design and simulation tools. The chips were fabricated by Mietec. Not surprisingly, the first version of the chip had some problems, but worked well enough to form the basis of the Unison wide-area Multimedia networking experiment. [1]

Olivetti Research made a second batch of CMOS chips with some bugs fixed, and these were used in the Pandora project. There still remained a sensitivity to supply voltage which was made it necessary to select chips carefully. The CFR generally worked well, but had periodic bouts of instability. It was operated at a bit-rate of 50MHz, which was chosen as a compromise between the demands of speed and reliability.

2.2 Technology

In 1990 The Inmos T2 (16-bit) and T4/T8 (32-bit) ranges of Transputers had been available for several years and had recently been enhanced with higher clock speed variants (20 and 25MHz).

It was necessary to retro-fit the Network Card to existing Pandora Boxes. Fortunately, the Inmos Links on the Transputers made this a simple task. The Network Card was connected via a 20MHz Inmos Link to the Server Transputer as shown in figure 1. The link could deliver over 10 Mbits/s of data bandwidth (Total 20Mbit/s for in + out), which was as much as a Pandora Box could reasonably expect to use on a shared 50MHz CFR.

Because the CMOS Station Chip's transmit and receive cell buffers were not double-buffered, it was clear that the performance of a CFR interface was limited by the loading and unloading of cells. The design aim for the Pandora Network Interface was therefore to minimise the time taken to respond to an interrupt from the CMOS Station Chip, and then to load or unload the cell data as fast as possible.

Initially, it was intended to solve the problem with hardware based on an FPGA (Field Programmable Gate Array). The best candidate was a Xilinx 3000 series FPGA but unfortunately, the devices available were far too slow to allow the required logic to be implemented. It was therefore decided to use a *second* Transputer as a 'slave' simply for cell load and unload. Again, connection would be straightforward, utilising a pair of Inmos serial links. The only extra hardware required would be a PAL to marshal the status signals from the CMOS Station chip.

2.3 Circuit Design

A block diagram of the circuit design of the first Pandora Network Card is shown in figure 2.

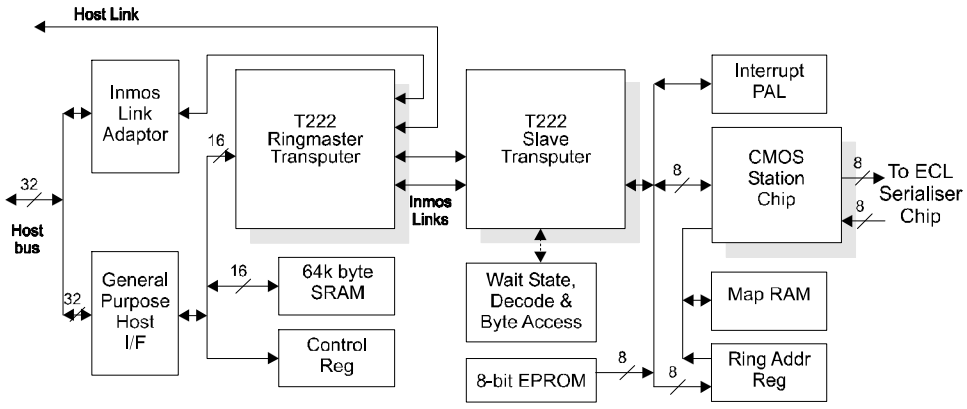

Figure 2: Block Diagram of the first Pandora Network Card

It was clear that the performance would improve if the buffer memory were larger, so an upgraded card was built. (See section 3.3) The use of a 32-bit Transputer (a T425) made it possible to increase the available memory from 64 kilobytes to 2M bytes. This improved the speed and effeciency of the Master Transputer to the point where it was no longer the limiting factor in system performance, speed being limited by the behaviour of the CMOS Station chip. This upgraded circuit was also used in Pandora lA. (See figure 3.)

As with all good hardware, Pandora continues to survive long after its predicted obsolescence. With the move to ATM networking and the abandonment of the CFR, it was required to connect the Pandora lA boxes to the ORL ATM network. Because the network card hardware and software is modular, it was possible to unbolt the CFR and bolt on an ATM interface. The resulting circuit is shown in figure 4.

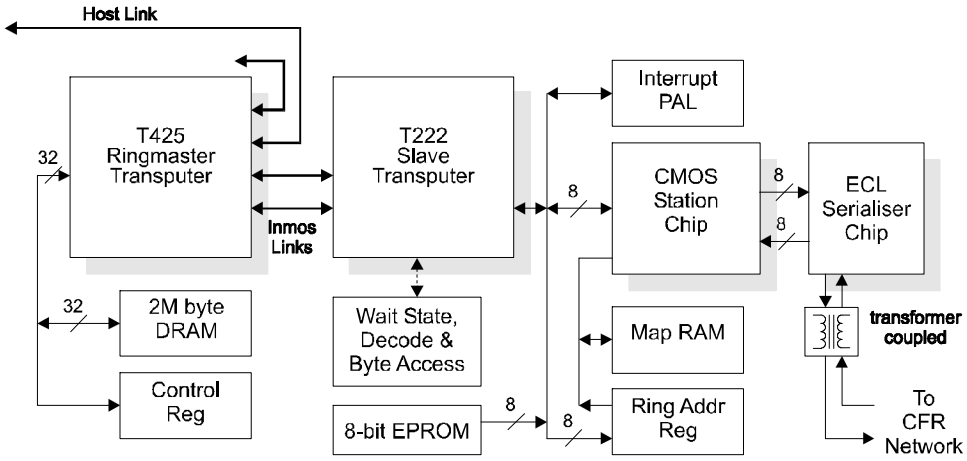

Figure 3: Block Diagram of the Pandora lA Network Card

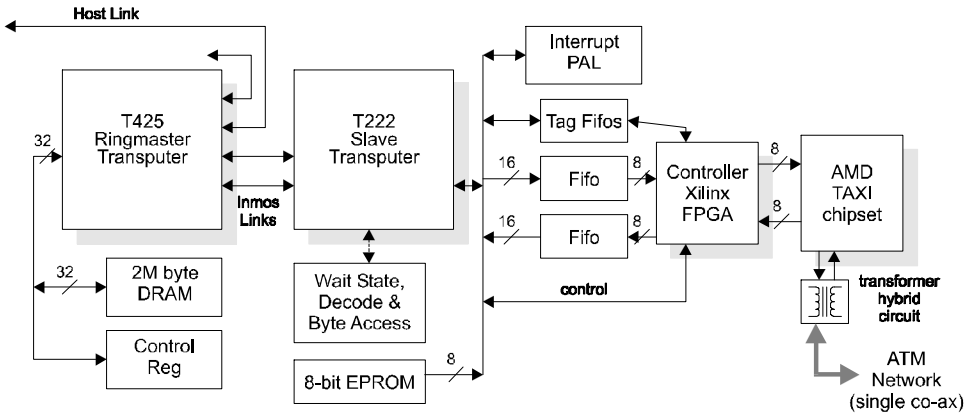

Figure 4: Block Diagram of the Pandora lA ATM Card

The new circuitry is more or less a straight copy of the relevant parts of the Atmos Card which was developed at ORL for the Medusa Project. [10] [15] [14] It uses the the 100MHz AMD TAXI chip interface, in line with the ‘orange book’ and ‘blue book’ guidelines from the Cambridge Computer Laboratory. [4]

Credits

The hardware was designed by David Clarke of ORL except for the ATM version of the card which was designed by Gavin Stark of the Cambridge Computer Laboratory (now with ATM Ltd). The boards are shown in figures 5 to 8 below.

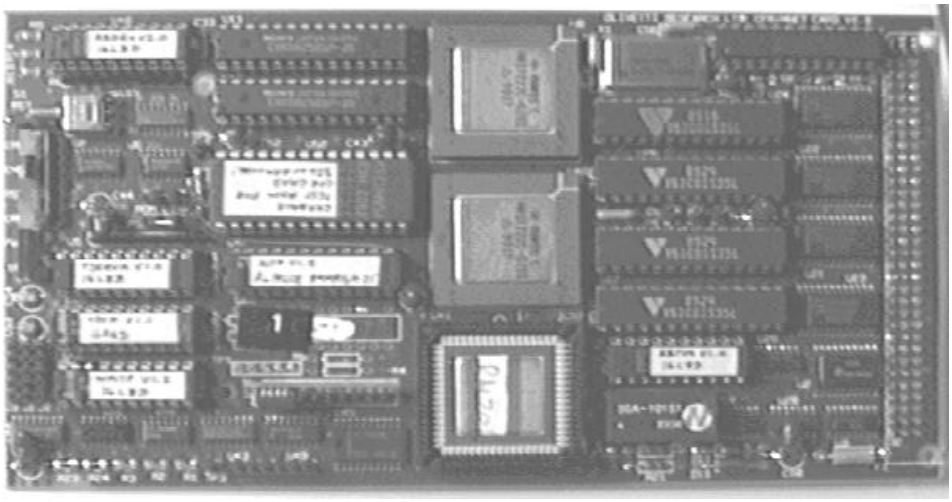

Figure 5: Pandora T2-based CFR Network card

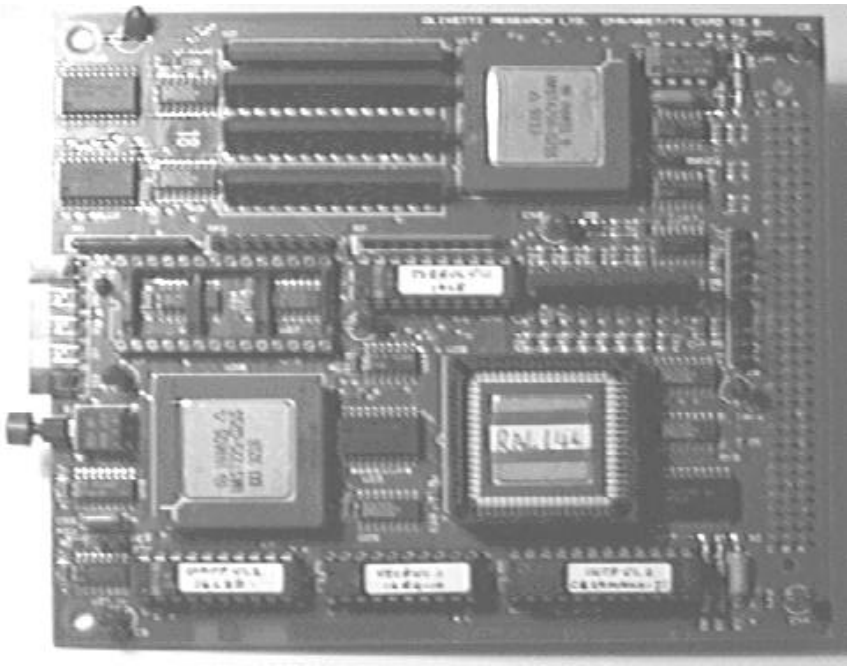

Figure 6: Pandora T4-based CFR Network card

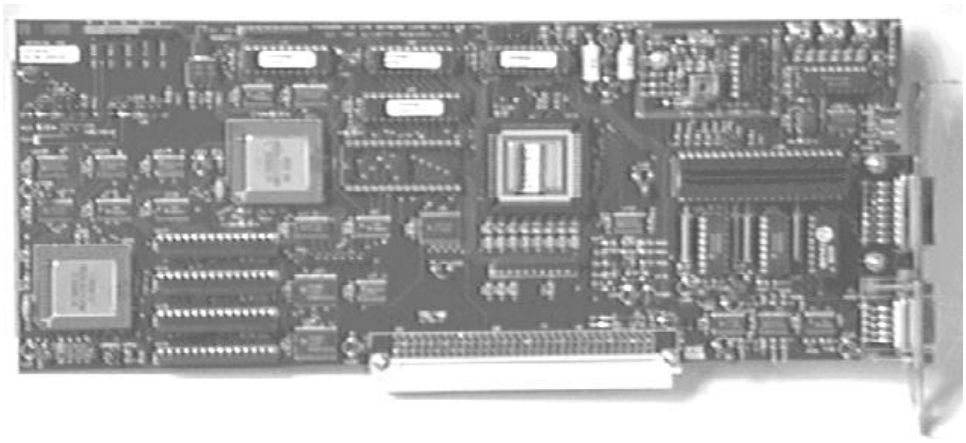

Figure 7: Pandora lA CFR Network card

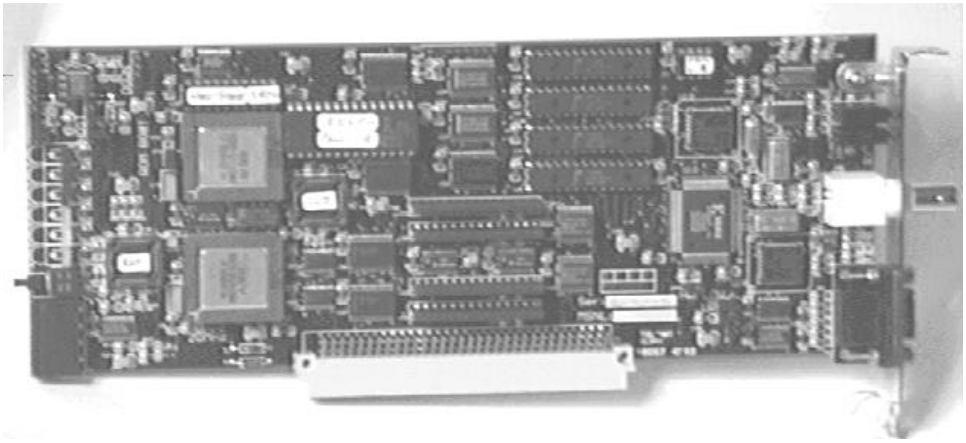

Figure 8: Pandora lA ATM Network card

3 Software Overview

3.1 Division of Labour

The Master processor deals with all the high-level tasks such as block segmentation/reassembly and connection signalling. The second Transputer (the ‘Slave’) relieves the Master processor of the time-consuming drudgery of loading and unloading cells byte-by-byte and handling interrupts.

The Cambridge Fast Ring CMOS Station chip has a certain amount of state associated with it. cells can be delivered or rejected (TOGged: ‘Thrown-On-the-Ground’). Transmit can be retried both at the hardware and the software level. There are a number of flags and status lines to do with full/empty/error status which must be properly handled. This is all dealt with by software state logic in the Slave Transputer. The Master software deals with issues such as lost cells, backoff strategies and interleaving of streams.

3.2 Slave processor software

The Transputer’s drawbacks stemmed mainly from its age, with memory access time and instruction rate lower than the fastest processors (eg the ARM2) available at the time. This meant that to optimise speed, the Slave processor code has to run entirely out of fast (1 cycle) on-chip SRAM. This in turn meant that the code had to be kept small, conveniently consistent with simplicity and fast execution.

The Slave software provides limited buffering of cells for both transmission and reception. (See figure 9.) It consists of four main processes, a transmit buffer, a receive buffer, a command buffer and an interrupt handler. Transmit queueing is optimised by assuming that a constant supply of cells is available for transmit. If the supply falters, a dummy cell will initiate a stop-and-request protocol until more cells are forthcoming. (Which is in line with a general strategy of always optimising the *worst* case and accepting less efficient code when there is less work to be done. This approach is one of the elements that have made Pandora behave well under stress.)

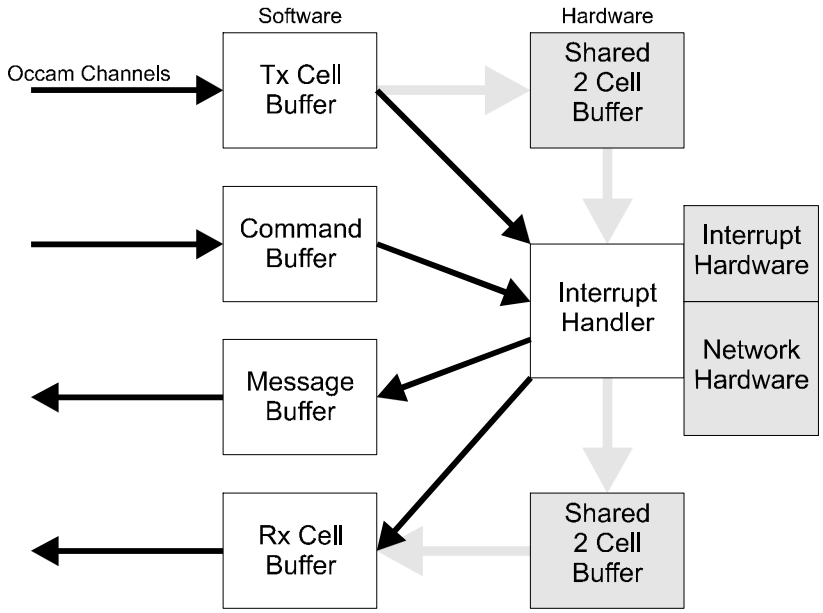

Figure 9: Process diagram of Slave Transputer software

Receive code assumes that the Master will always accept cells. There is no flow control mechanism for received cells in the Slave code. This works because the receiving process in the *Master* is high priority and will always respond promptly.

The state logic in the Slave is driven largely by interrupt. Events such as cell reception, Tx buffer-empty and so on cause an interrupt (the Transputer has only one interrupt line.) which schedules the service routine. This determines the causes and services them one-by-one before awaiting the next event. Reception of a cell is always handled first, followed by any other causes. This non-blocking behaviour means that commands from the Master will still be acted on in the face of continual cell receptions.

In the ATM version of the Network interface, the Slave code has been completely re-written. However, the general form is a simplified version of the CFR slave code. This simplification comes mainly from the much more helpful nature of the ATM network hardware. (Deep fifos, fewer failure modes.)

3.3 Master processor software

Buffer allocation is one of the major elements in the Master software. The software pre-allocates buffers of 8k bytes each, sufficient for all practical audio and video traffic to and from the Box. Buffers are kept track of by means of ‘free’ and in-use linked-lists. This scheme (unsurprisingly) gives a significant performance gain over the original arrangement of allocating buffer space on a per-cell basis. The main reason was that a received cell can be written directly into its proper place in a block. This avoids the overhead of handling the cells individually when reassembling a block.

The Pandora Box is managed by a system of commands and replies travelling between the processes, on Occam Channels. Master and Slave communicate using a small, fixed set of messages, whereas the messages from the Master to the rest of the Box are rather more open-ended. [2]

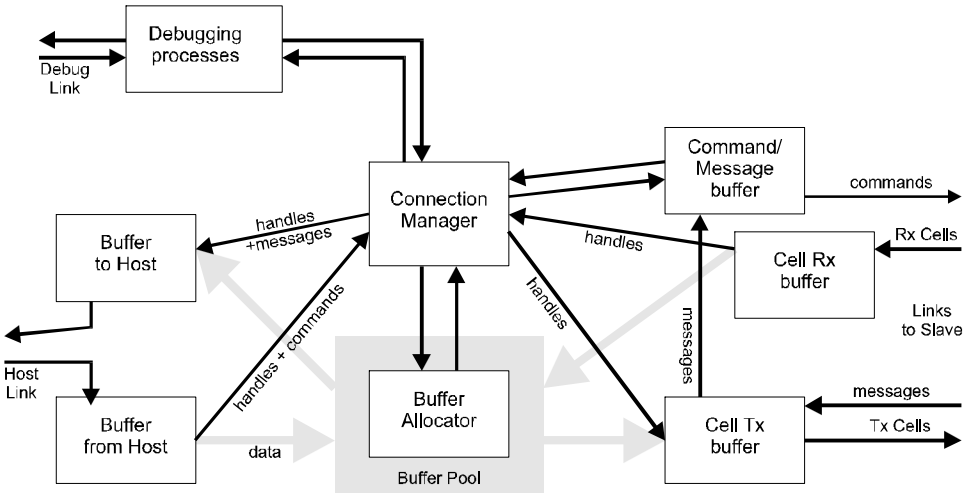

Figure 10: Simplified Process diagram of Master Transputer software

Figure 10 shows a very simplified block diagram of the process structure of the Master software. It must be borne in mind that the language used (Occam) supports fine-grain parallelism and message-passing very efficiently so that single tasks such as a buffer manager may in fact consist of a number of co-operating parallel processes. This level of detail is usually glossed over because the process structures tend to fall into a pattern, just as do other more conventional programming constructs.

Interaction with the Host

Interaction with the Host (the rest of the Pandora Box) is chiefly concerned with optimising the use of the single 20MHz Inmos Link to the Server Transputer. Buffering is used to ensure maximum use of the Link hardware, but this is optimised to ensure that cell data does not suffer unnecessary memory-to-memory copies.

Commands and messages have a rather more complex path to follow, involving processes which may already be busy, or may only run occasionally at low priority. To minimise disruption of the main task of shipping data, both commands and messages are passed through elastic buffers.

Control of the Slave Transputer

The control of the CFR Station chip is orchestrated by the Master Transputer. Actions such as resetting the Station Chip or changing the contents of the Map Ram (to accept or reject different Virtual Circuits) are initiated by commands from the Master to the Slave.

The ATM hardware requires much less management than the CFR interface. Indeed, the only reason for keeping the Slave processor on the ATM interface was for ease of implementing the software. (By porting the CFR software.) The large fifos for transmission and reception place less time constraints on the Slave software, compared to the (single-buffered) CFR. The ATM interface also requires little maintenance, unlike the CFR which has many failure modes. It is an interesting reflection on the ease of use of Occam and the Transputers that it was easier to write code for a *multiprocessor* system than for a uniprocessor.

Data Transmit

Transmitting to the CFR has a number of complications to do with failure to deliver a cell. The appropriate response is to send a cell for a different stream (if available) before retrying the failed one.

Because of the pipelined nature of the connection to the Slave Transputer, this means that wherever it can, the Data Transmit software attempts to interleave cells from different streams so that a single lost cell will not reappear out-of-order when retransmitted. There are also strategies to deal with multiple retry, backoff and priority. The last item is vital in an application involving both audio and video, as experience shows that audio should take priority over video. [9]

Data Receive

The Receive process receives direct into the correct place in a block. It does this by getting the VCI (Virtual Circuit Identifier) and header first so it can do a channel input to the right piece of memory.

It also watches a timer to see if a block remains incomplete for a long time (typically a second). This allows the reassembly to be aborted and the buffer freed on the assumption that the rest of the block will never arrive.

Connection Management

Merely shipping cells is not enough the make a Network function. Some higher-level protocol has to be engaged in to make and abolish connections between machines, and to bundle up and check blocks of data. In Pandora, the protocol is MSNL (MultiService

Network Layer). [13] The Connection Management software sends and receives this special signalling data on behalf of the rest of the Box, and sets up the conditions for correct segmentation and reassembly of data blocks.

It is possible for a data stream to arrive (either for Transmit or Receive) faster than it can be disposed of. If uncontrolled, this would consume all available buffers.

The Connection Management software includes a 'Buffer Limiting' function which restricts the number of *transmit* buffers allowed to be used on a single stream. This minimises the adverse effects on other concurrent streams. The *receive* side does not suffer from this problem because the receive rate is limited by the network, and not by the ability of the Pandora Box to sink data.

Debugging Facilities

The debugging system in the Master Transputer's software was designed with two purposes in mind:

- It allows remote access to the Pandora System's Network Interface, including Connection Management.
- To allow the collection of performance statistics without passing them through the Box message system. This avoids having to rely on 'post-mortem' type analysis via the Box log-file.

With later improvements to the Debugging software, it is possible to reset and reboot the Box remotely over the (ATM) network, booting either from EPROM or from a network-based fileserver.

Credits
All the software was written by Gavin Stark of the Cambridge Computer Laboratory (now with ATM Ltd), apart from the CFR Slave code which was written by David Clarke of ORL.

4 Conclusion

Pandora, is a good demonstration that it is possible to develop a multiprocessor system and have it work, provided the software is written in a *secure* language which supports communication primitives *efficiently*.

The writers of the Pandora Box firmware had very little difficulty in working together. Was this in spite of the crude nature of the development system or perhaps *because* of it? The TDS effectively limited the opacity of how the software was constructed. In particular, because all the programming, compilation and loading takes place in a 'shell' programme, it prevented the 'let's play guess the filename' problems that can hinder understanding in other more conventional systems.

A network interface needs 'enough' buffering, although you hope never to have to use it. 64 kilobytes is too little but a megabyte is probably generous. (A megabyte of audio delay is far too much for a live conversation to tolerate.) If an ATM network interface needs megabytes of buffer memory then something is wrong and the quality-of-service will suffer.

The use of a second processor introduced a data pipeline, which was a bit of a nuisance, but manageable. A second processor can prove more effective than an FPGA in terms of both cost and chip count.

The final performance of the CFR interface was limited by the inherent performance of the CFR CMOS Station Chip, giving up to 1.5 megabits transmit capacity and up to 3 megabits receive capacity at the same time. Transmit capacity was particularly affected by a large (7 slot) CFR, because of the longer round-trip for acknowledgement of cell delivery. The ATM version of the card gives improved performance primarily because of the properties of the ATM packet-switched network. In practice it can source and sink as much as the rest of the Pandora Box can handle, and is not limited by the Inmos Link throughput.

Both the CFR and the ATM -connected boxes have been demonstrated on bridged networks, but the performance was often reduced. Properly interconnected ATM networks effectively remove this problem.

Pandora showed that it is possible to do live multimedia on a packet-switched ATM network. The bandwidths required to deliver a useable system (a few megabits) are not as high as many people believed.

References

[1] Various Authors. The final report of the unison project. Technical report, Loughborough University of Technology, 1990.

[2] Dr A Hopper Dr A. H. Jones. Handling audio and video streams in a distributed environment. Technical Report 93-4, Olivetti Research Limited, 1993.

[3] Prof. Roger Needham Dr Andy Hopper. The cambridge fast ring networking system. Technical Report 88-1, Olivetti Research Ltd, 1988.

[4] Mark Hayter. Fairisle/yes/maybe atm taxi link protocol. Internal working document, University of Cambridge Computer Laboratory, 1992-4.

[5] C. A. R. Hoare, editor. *occam 2 Reference Manual.* Prentice-Hall and Inmos Ltd, 1988.

[6] Dr Andy Hopper. Pandora - an experimental system for multimedia applications. Technical Report 90-1, Olivetti Research Ltd, 1990. Also published in ACM Operating Systems Review, April 1990.

[7] Dr Andy Hopper. Digital video on computer workstations. *Proceedings of Eurographics,* 1992.

[8] David Tennenhouse Ian Leslie, Derek McAuley. Atm everywhere? *IEEE Network,* March 1993.

[9] Dr Tony King. Pandora: An experiment in distributed multimedia. Technical Report 92-5, Olivetti Research Ltd, 1992. also presented at Eurographics '92.

[10] Olivetti Research Limited. *The Atmos II Reference Manual.* ORL, 1994.

[11] Inmos Ltd. The Transputer Reference Manual. Prentice-Hall, 1988. ISBN 0-13-929001-X.

[12] Inmos Ltd. *Transputer Development System.* Prentice-Hall, 2nd edition, 1990.

[13] Dr D. R. McAuley. Protocol design for high speed networks. Technical Report 186, Univ. of Cambridge Computer Laboratory, Jan 1990. (Doctoral Thesis).

[14] GJF Jones K Sparck Jones SJ Young MG Brown, JT Foote. Video mail retrieval by voice: An overview of the cambridge/olivetti retrieval system. *Presented at ACM Multimedia DBMS Workshop,* 1994. Describes an application of the Medusa Project.

[15] Andy Hopper Stuart Wray, Tim Glauert. The medusa applications environment. *Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Boston,* pages 265—273, May 1994.