The Medusa Video Brick:
An ATM Camera

DJ Clarke
Olivetti Research Limited

ORL Technical Report 95.4
2nd May 1995

Contents

1 Introduction

The Medusa project was conceived as a follow-on from the Pandora project at Olivetti Research. [10] [11] In Medusa, audio, video and computer data are carried by an ATM (Asynchronous Transfer Mode) network between smart network objects and workstations. The smart network objects are known as 'SAMs' (Smart ATM Modules).

This document describes the Medusa Video Brick which is a SAM for the capture of live video directly onto the ATM network. This paper explores the design aims, and presents the ways in which they have been satisfied. Some thoughts are also offered on future developments in SAMs for networked video.

2 Design Aims

2.1 About Medusa

The catalyst for Medusa was the emergence of ATM (Asynchronous Transfer Mode) as a recognised network standard. We were presented with an excellent opportunity to re-work the Pandora idea in a radical new way, providing a much more flexible mutimedia platform. Instead of having a fixed set of I/O devices in a single box, each I/O device would now have its own network interface, and the network itself would become the switch connecting them up.

Pandora was built to use the Cambridge Fast Ring network [4] which pre-dated ATM networks but had many of its desireable properties. The Medusa project, like Pandora, supports *multiple concurrent streams* of audio and video, and exploits the ability of the network to deliver fine-grain sharing of bandwidth.

Olivetti Research Ltd has put considerable effort into the development of ATM hardware and software. We have designed a generic ATM network interface card and an ATM switch fabric based on 4-port and 8-port switches. All these items contain an ARM processor (currently the ARM610) and run our ATMOS real-time kernel. [3] [7]

Medusa is both a software and a hardware project. The SAM hardware is controlled by software composed of re-useable modules which process and transport the multimedia traffic. Control is by means of scripts written in TCL. [14] [13] Connections between these 'Medusa Modules' is network-transparent, for both control *and data*, allowing applications such as videoconference and video mail to use whichever SAMs and computing resources are appropriate. We have built applictions combining multiparty multi-stream video, multistream audio, and intelligent agents all working together. [121 A typical Medusa workstation is shown in figure 1.

The advantages of this approach include the ability to build systems without the presence of dedicated workstation. For instance, the ORL rooftop camera which provides a live panorama of Cambridge for the ORL Worl-Wide-Web page is simply a Video Brick SAM connected to our ATM network. The uniformity of the hardware makes for economy in software and hardware development. All SAMs run the same ATMOS software. Hardware upgrades are done by just replacing ATMOS cards in SAMs.

Perhaps the biggest benefit of the distributed approach is that processing power and network bandwidth scales with the number of connected objects, so there is no hard limit on the size of an installation.

Figure 1: A Typical Medusa Workstation Setup

2.2 About the Video Brick

The function of the Medusa Video Brick is to deliver digital video onto the ATM network. It is designed to handle live video, such as would be required for a video-conference, but it is also capable of capturing stills to the full resolution available from PAL or NTSC TV.

2.3 The user's view

Who is really the user of a camera? Experience from Pandora showed that even in a closed community of users where abuse of the system is rare, the camera tended to be pointed at the wall or out of the window, unless the person sitting near the Pandora workstation was engaged in a videoconference or recording video mail. Medusa provides many cameras per location, so part of the design aim was to provide a user with as much feedback as possible about the status of the cameras, and to provide a few basic mechanisms in hardware for control. This is an area of research which is more social science than engineering, but is likely to assume major importance as Medusa-like systems filter into everyday use in the workplace and the home.

2.4 Network integration

The concept of a Smart ATM Module is one that emerged as the Medusa project developed. It is tempting to design ATM peripherals which have 'no brain' and merely suck and blow ATM cells continuously and indiscriminately. However, this solution brings problems of security and control, and would also make heavy demands on the functionality of the ATM switch fabric. By using a processor-driven ATM network adaptor (A so-called 'deep' adaptor,) we have been able to make the Video Brick a 'first-class network object' which can be contacted and used by any other ATM-connected entity. This fits well with the distributed processing model adopted by the Medusa software.

3 Practical Implementation

3.1 What should the Medusa Video Brick look like?

Concept: A small box with an ATM cable coming out of the back, and a status display on the front, next to the camera lens.

Reality: Video cameras are complex and can be costly. Until very recently, all affordable cameras produced only an analogue output to some television standard such as NTSC or PAL, so the Video Brick needed to contain decoding and digitisation circuitry, an ATM interface and space for optional compression hardware. It would not be 'a small box'.

Therefore the design was split into two parts as shown in figure 2:

- A 'small box' containing the analogue camera and some displays
- A larger box containing digitisation, processing and ATM interface.

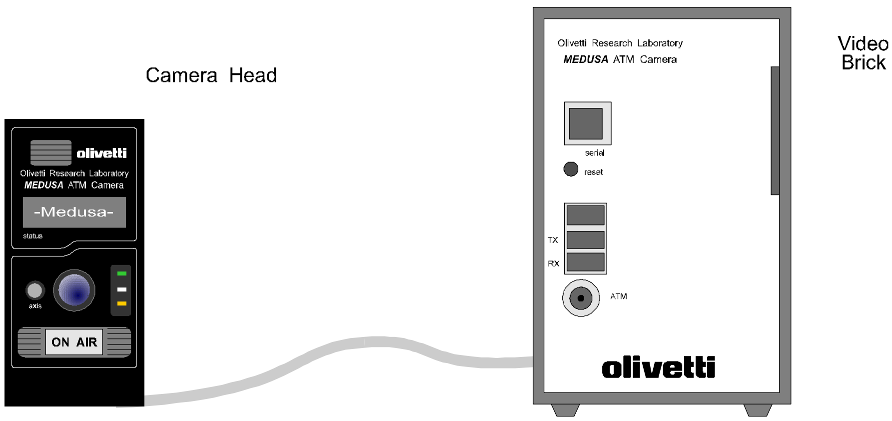

Figure 2: Medusa Video Brick and Camera Head

There are many shapes and sizes of video camera. (See figure 3.) We chose a 'card camera 'from Pacific Corp of Japan, because it was small enough to build into a case with a status display. It is well adapted to the generally poor or localised lighting found in an office, a factor which was not given much weight at the outset, but which has turned out to be important.

It had been intended to put some pushbuttons on the camera head to allow a user to interact with the status display. However, many cameras would be placed where they were awkward to reach, so an infra-red receiver was installed instead. This allows a user to send commands from a low-cost TV remote-control type handset; it also gives the Video Brick the capability of sighting beacon signals from Olivetti Active Badges [6] [5] [8] worn by users nearby.

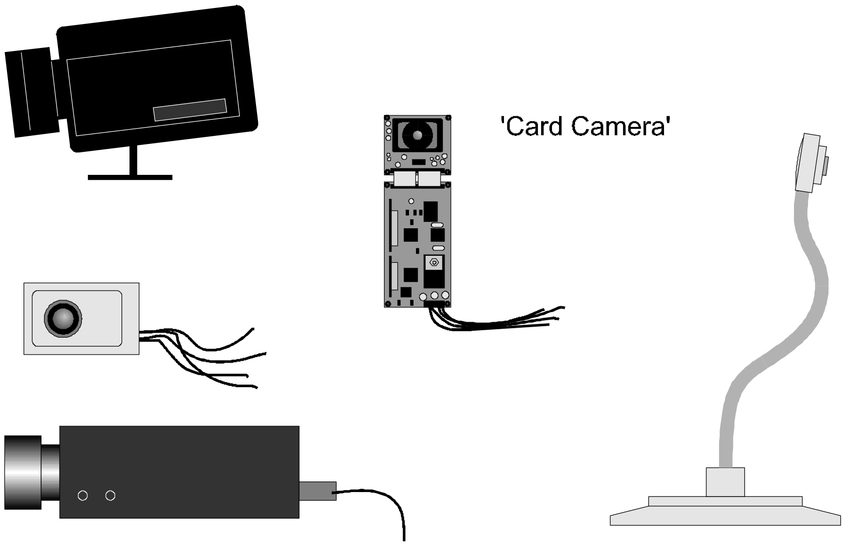

Figure 3: Various standard camera shapes

3.2 Overview of Video Brick subsystems

The Video Brick consists of several circuit boards plugged into a short backplane. The network interface is the ubiquitous ‘Atmos Card’, which for this application is always used with an ‘XSI Adaptor. The two plug together to make a unit which is single-extended-eurocard size (100mm x 220mm). [2] There is space in the Brick for one or two complete video capture subsystems. Each plugs separately into the XSI backplane. (See figure 4.)

3.3 Computer Video Basics

Current analogue broadcast television standards evolved from those designed by EMI at Hayes for the BBC’s 1936 monochrome 405-line transmissions from the Crystal Palace transmitter in London. [1] This standard was designed to allow a television set to be constructed with just a few valves (no semiconductors!), taking all the timing information directly from the radio signal. In order to minimise *flicker* and fool the eye into seeing a more steady image, it was necessary to trace out a picture on the screen 50 times a second. The technique of *interlace* was devised to achieve an apparent increase in resolution without increasing the bandwidth of the signal. This is achieved by denoting alternate *fields* as ‘odd’ or ‘even’ and making the television set display one slightly displaced vertically relative to the other. This aspect of restricted bandwidth was an important factor in containing the cost of the whole system, so much so that a video line rate of 10kHz was deemed satisfactory despite the consequence that every television set would whistle quite audibly because of vibration of the display tube’s deflection coils.

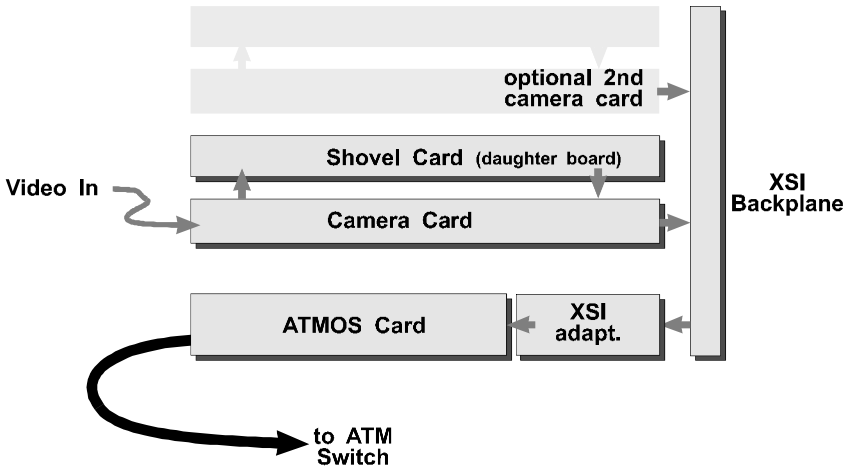

Figure 4: Arrangement of cards in a Medusa Video Brick

Current systems such as PAL now use a line rate of just over 15kHz, so only some of the population now hear the whistle, but the system remains otherwise similar to the original monochrome standard. Interlace has served television well, but for computer video it is pretty much of a disaster.

Computer screens rarely use interlace, and almost always use a *refresh rate* of 65Hz or higher. The eye may be prepared to ignore the vertical wobble of an interlaced picture on a television set, but when an interlaced video image is mapped directly onto a bitmapped computer screen it can look dreadful! If both odd and even (interlaced) fields are mapped to the **same** pixels on the screen, motion will be very smooth, but the whole image will exhibit a subtle but intrusive vertical wobble. If pairs of fields are **merged** to make a sequence of higher-resolution *frames*, the vertical edges of moving objects acquire a serrated appearance. The most satisfactory option is usually to display only odd fields (or only even fields). Motion will be less smooth and, of course, the image is less detailed, but at least the problem of flicker is now eliminated by using the computer display's framestore.

In fact there are now domestic television sets which contain a digital framestore from which they refresh the screen at 100Hz, producing a display as steady as a computer workstation screen and eliminating the 15kHz whistle. These sets still display an interlaced picture, but some have digital processing to determine the best order to display the fields. (Either [odd-odd-even-even] or [odd-even-odd-even], depending on whether static graphics or moving images predominate.)

Fortunately (for computer video, that is), only the very best cameras can resolve all of the the vertical detail implied by having two interlaced fields. Futhermore, the way in which the colour information is encoded for broadcast by PAL or NTSC demands a loss of detail in the *chrominance* (colour) signal, relative to the *luminance* (brightness) signal. This loss applies in both the horizontal and vertical directions. (Most commercial video cameras exploit this feature in that their CCD sensors are designed not to resolve chrominance detail as well as they do luminance.) Being able to display only one of the two interlaced fields is therefore not a serious problem as the single-field image generally retains most of the useful image detail.

The situation is complicated by the tendency of computer manufacturers to supply 8-bit colour-mapped displays, as the framestores are cheaper than full-colour ones and the processor can update the screen faster (less data to write). Video requires full colour, so if the range of apparent colours is to be acceptable, some kind of dithering must be employed at the display. Dithering is a well-established technique which trades image resolution for more apparent colours. It relies on the property of the eye that it tends to perceive the average colour of an area rather than the colours of the individual dots. On a computer display, one might elect to use a scheme which used four 8-bit screen pixels to represent each of the incoming video pixels. This has the effect of doubling the linear dimensions of a displayed picture; a full **field** of PAL video would then cover over 700x500 pixels, which would be quite respectable on most current screens. In fact the resolution of a shadow-mask CRT display is usually poorer than the pixel-spacing of the framesore driving them, so this doubling of picture dimensions is probably desireable anyway to show all the detail present in the image.

3.4 Atmos Card and XSI Bus Adaptor

All the SAMs (Smart ATM Modules) currently in use at Olivetti Research Ltd use an Atmos Card as their ATM interface. This is a single-board computer containing an ARM processor (currently the ARM61O), boot ROM, NVRAM and an ATM network interface. Current models have 8MByte of DRAM. The cards do not carry application-specific firmware; they all have a standard boot ROM and bootstrap accross the ATM network from any available bootserver machine. The Atmos Card has a generic interface connector which brings out most of the ARM processor's bus signals. Being unbuffered, these are not suitable for driving a backplane; an XSI Adaptor [2] is used here because it provides bus buffers, strobes and bus cycle timing.

The Video Brick's network interface can readily be upgraded or changed by exchanging the Atmos Card and XSI Adaptor for cards with different features, such as higher processor performance. Hardware support for AAL5 checksums will be available, as will support for different physical layers such as Unshielded Twisted Pair cabling.

3.5 Camera Head

The Video Brick can accept standard PAL (or NTSC) video from any source. However, it was designed to be connected to a special camera head (see figure 5) which includes a status display and Infra-Red receive and transmit. A photograph of a complete Video Brick is shown in figure 6.

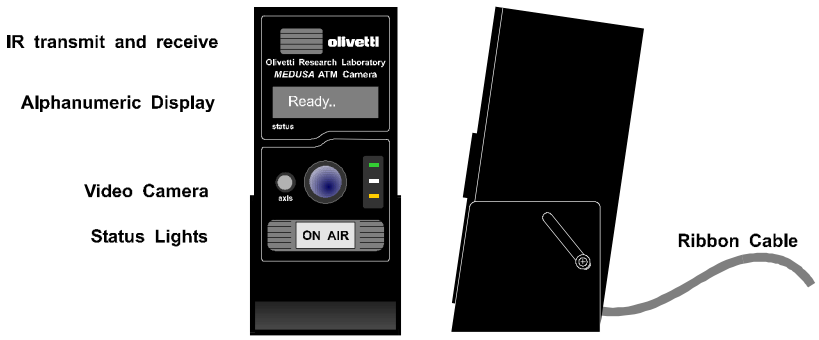

Figure 5: Camera Head for Medusa Video Brick

Figure 6: A Complete Video Brick

3.6 Camera Card Hardware Description

Figure 7 shows a block diagram of the circuitry which turns an Atmos Card into a Video Brick. (The split between the Camera Card and the Shovel Card is to allow the removal of the Shovel Card and its replacement with compression hardware.)

Decoder

All the video inputs to the Camera Card are analogue, including the input from the camera head. An electronic switch selects one input which is then decoded by a Philips SAA7191 Digital Video Decoder chip. This chip samples the video at a clock-rate of 14.75MHz (for PAL TV) to give full-resolution (768x576 pixels) colour digital video with square pixels. The aspect-ratio of the pixels is important, as the video will be shown on bit-mapped computer displays, which invariably use pixels with a 1:1 aspect ratio. By comparison, broadcast-standard digital video to the CCIR601 standard uses a pixel sampling rate of 13.5MHz for PAL. If video captured to this standard were displayed on a computer, the image would appear to be about 10% too narrow.

Digital Video Scaler

The Digital Video Decoder is directly connected to a Digital Video Scaler, the Philips SAA7186, which contains digital filters and decimation circuitry. The pixel data which emerges is a reduced-size version of the original image, containing less pixels (and hence less detail) but substantially free of aliassing artefacts. The filters are programmable, as is the size of the output image. The ‘active area’ of the original image is also programmable allowing a limited amount of pan and zoom to

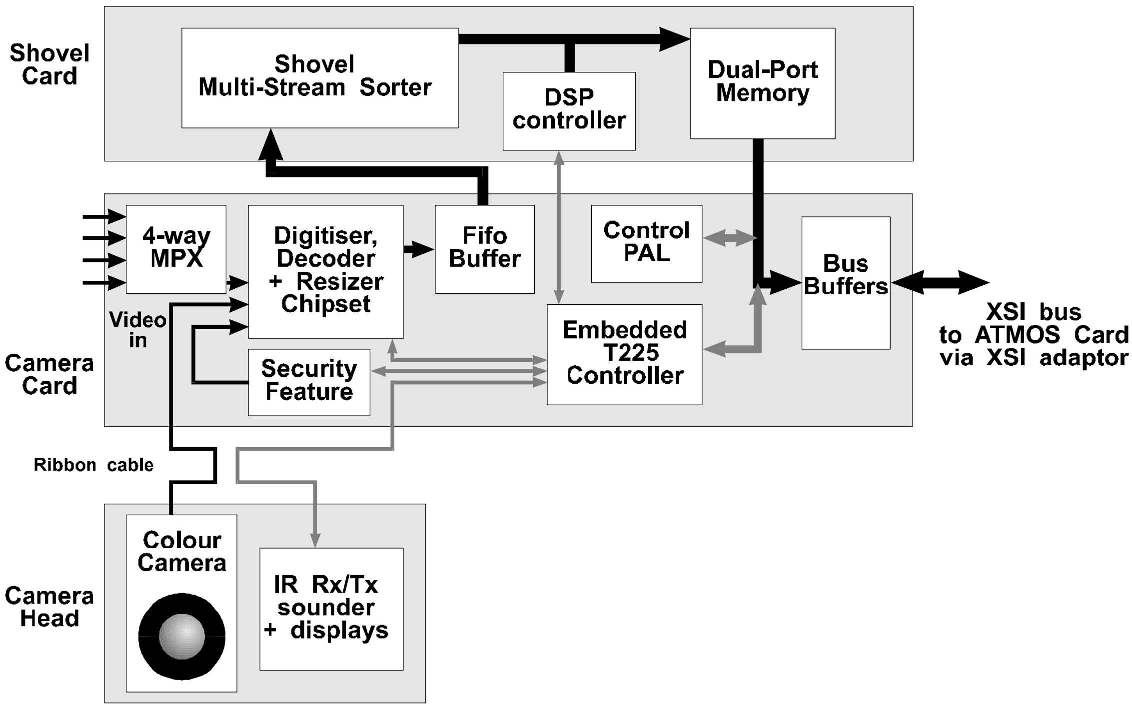

Figure 7: Block Diagram of the Medusa Video Brick

be achieved. The output of the Digital Video Scaler is fed via some FIFOs to the daughter-board connector.

FIFO Buffers

Extra FIFO buffering is necessary at this point because the output FIFO built into Digital Video Scaler is only 16 pixels deep. Data can be ready to be read as often as every lμs, which is hard to service with a processor-driven system.

3.7 Shovel Card Hardware Description

The Shovel Card is controlled by a DSP, an Analog Devices ADSP 2111 which was chosen for its combination of on-chip memory and host interface port. This port eliminates the need for a boot ROM; the boot image is written there by the host processor, in this case, the Camera Card controller processor. It also provides a convenient way of handshaking control data between the processors for changes in image size and format.

Bit-Laning

Video from the Camera Card FIFOs enters the Shovel Card via bit-laning hardware which allows colour or monochrome pixel formats to be handled correctly.

Shovel Multi-Stream Sorter

The Shovel Card is named for the 'shovel' or fly-by DMA circuit which is used by the DSP to move video pixels from input to output. The DMA sequence includes decimation which results in three separate images from each video field. These images have linear dimensions in the ratio 1 : 0.5 : 0.25, allowing users a greater choice of image size.

Dual-Port memory

Segments (typically 8 image lines) of video are buffered in the dual-port memory, and are available to be read by the Atmos Card for on-shipment on the ATM network.

3.8 Camera Card Bus Interface and Controller

XSI Bus Interface

The Camera Card had bidirectional bus buffers and address decode logic for connection to the XSI bus. There is also some interrupt marshalling so that only one XSI interrupt line is used.

Controller Processor

The Camera Card is managed by a T225 16-bit Transputer. This processor was chosen for its fast interrupt response and ability to handle many threads without the need for an operating system. Its tasks are:

- Update the Digital Video Scaler parameters every video field.
- Receive and decode Infra-Red signals.
- Control the video input selector.
- Drive the user-feedback displays and sounder on Camera Head.
- Bootstrap the DSP on the Shovel Card.
- Provide control for changes in image size and format.

4 Software Issues

4.1 Camera Card firmware

All the software on the Camera Card is written in Occam [9]. This language's ability to handle many concurrent tasks securely and with fast context switch, makes it ideal for an embedded application of this type.

Occam operates entirely by message-passing between processes. It was therefore natural to adopt this model for communication between the Camera Card and the Atmos Card. The physical link between the two processors is an Inmos Link Adaptor chip which presents a byte-wide bidirectional port to the Atmos Card's ARM processor. The messages passing along this link are handled directly in the Occam software with channel-in and channel-out language primitives. (The other side of the link, on the Atmos Card, requires the use of an interrupt handler routine and shared-memory message queue.)

Management of the Camera card is embedded in one seqential process which handles incoming messages one at a time and sends back replies when appropriate. Most of the low-level control (which in other languages would be in device-drivers) is in a number of concurrent parallel processes which communicate with each other and with the management process via Occam message-passing channels. This is the natural way to express functionality in Occam, because the parallelism in the language is supported by a hardware scheduler, yielding fast code with very fast (sub-microsecond) context-switching.

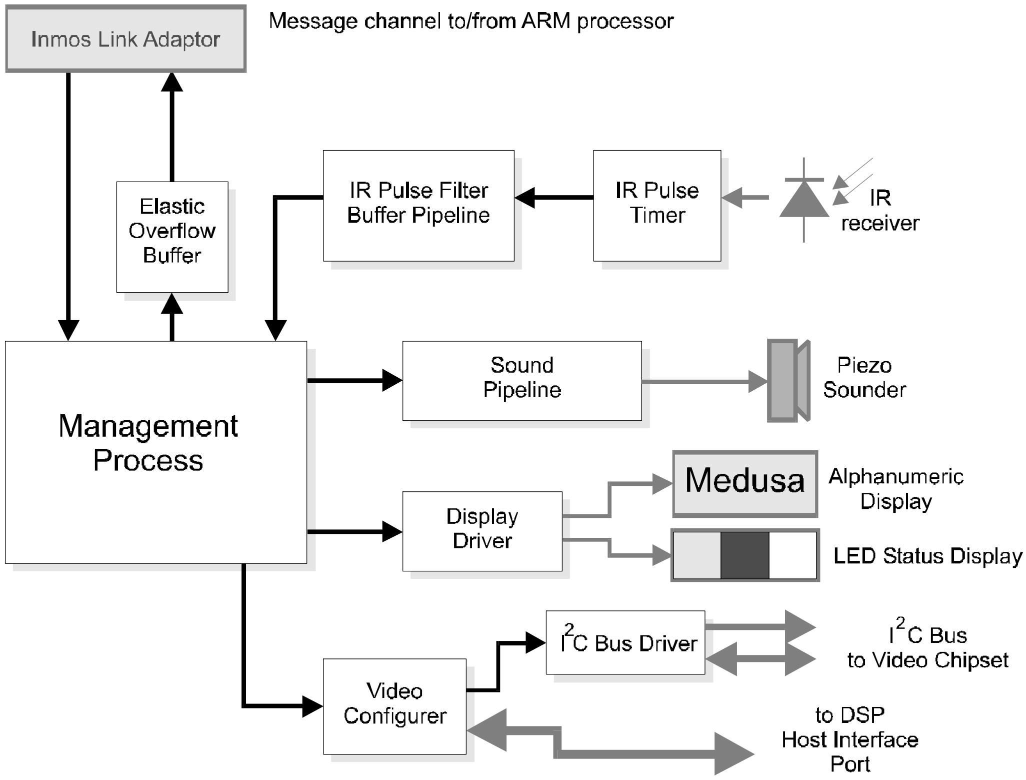

Figure 8: Simplified Process map for Camera Card Firmware

To give a flavour of the way the Camera Card firmware works, a simplified process diagram is shown in figure 8. Many simple processes have been omitted for clarity; this is because the general form of processes tends to fall into a pattern, much as code does written in any other language. For example, an elastic buffer 'process' would usually be constructed by means of two or more communicating parallel processes.

4.2 Shovel Card firmware

The Shovel Card is named for the function it performs, namely moving video pixels. This it does with the aid of hardware assist, dubbed the 'Shovel', which allows the DSP to perform simultaneous read from a FIFO buffer and write to a Dual-Port RAM. The DSP does groups of read cycles, generating a sequence of a addresses in the process. The Shovel hardware causes the read and write strobes which read from the FIFO and write directly to the Dual-Port RAM.

The firmware which runs on the DSP on the Shovel Card is written in Assembler. This was necessary as the main task of moving video pixels needed to be done at maximum speed. There is an amount of housekeeping software but the most of the time is spent in a loop which copies pixels in groups of 8. The loop is shown diagrammatically below in figure 9.

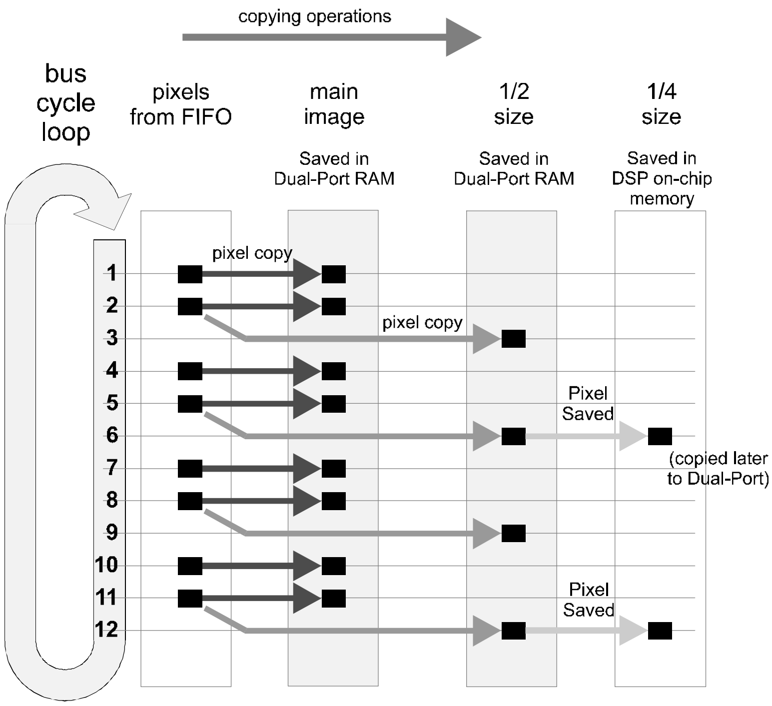

Figure 9: Shovel software loop

The DSP software and the Shovel hardware create three different sizes of image in the three buffer areas of the Dual-Port RAM. The largest image is passed through unchanged, whilst the other two are decimated by a factor of two and four respectively, tripling the number of concurrent image sizes available to the user.

With hindsight this is not quite the free lunch it might appear to be. Although the ARM processor on the Atmos Card is relieved of the task of decimating the video, there is just over 30% extra data to copy from the Dual-Port RAM.

4.3 Medusa software

The software on the Atmos Card is written in 'C' with some time-critical parts coded in assembler. At the lowest level, there is a substantial device-driver which services the video interrupts from the Camera Card/Shovel Card and copies video segments into buffer memory.

This data is then processed (if desired) and shipped onto the ATM network by Medusa software modules from the Medusa software library. These program fragments are made as a sort of software Lego which can be plugged together without the programmer having to hand-code the data handling. Key features are network-transparent data transfer (even via Ethernet) and exposure of 'Attributes' which can be used to observe and control the modules. Medusa software also runs on other SAMs (Smart ATM Modules) and on workstations as part of the Medusa software infrastructure.

At boot-up the Atmos Card runs a bootstrap loader which broadcasts for a bootserver (usually a workstation) to send it boot data. This data will include all the code for the Atmos card as well as data to boot the Camera Card and Shovel Card. There are no ROMs to replace when altering the software or exchanging hardware. The code run on the Atmos Card includes the Atmos operating system microkernel, which provides process-switching and message-passing.

5 Cameras of the future

Digital solutions bolted onto a largely analogue world are always at a disadvantage in terms of cost and complexity. This will change as digital interfaces become standardised and more widespread.

Computer video will demand its own breed of cameras with digital-only outputs. In the real world where bandwidth costs money, both inside and outside the computer's box, the key issues will be image quality and image compression.

There is a basic problem here, in that good compression of video requires inter-frame coding, but this can introduce too much delay for live person-to-person video. A case in point is MPEG compression which in a full implementation contains bidirectionally predicted frames. Allowable compression delay is further severely constrained where synchronisation must be maintained with live audio channels.

Good video compression also encodes as much picture detail as possible in as few bits as possible, making the extraction of *part* of the information as onerous as extracting the full thing. Where a recipient requires only a small low-quality image, he or she will be unwilling to either pay the shipping charges for the full sized one, or to burn up their computer decoding it.

The Medusa Video Brick showed that even a modest choice of image qualities (6 in this case) allows a user great flexibility in configuring their use of bandwidth. In the future, digital cameras will need to be able to supply video at many qualities/sizes simultaneously.

Such schemes are already well-known. Pyramid coding is used on the Photo-CD but is not efficient enough for motion video. It is likely that more advanced schemes such as sub-band coding and fractal encoding will begin to look increasingly attractive for computer video once suitable compression/decompression hardware becomes available.

DC

References

[1] *Private communication, Prof MV Wilkes.* November 1994.

[2] Alan Chaney. Xsi bus specification. Technical report, Olivetti Research Ltd, 1994..

[3] Derek Mcauley David Greaves. Private atm networks. Technical Report 92-7, Olivetti Research Ltd, 1992.

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

[5] Dr Andy Hopper Dr Roy Want. Active badges and personal interactive computing objects. Technical Report 92-2, Olivetti Research Ltd, 1992. Also published in IEEE Transactions on Consumer Electronics Feb 1992.

[6] Dr Andy Hopper et al Dr Roy Want. The active badge location system. Technical Report 92-1, Olivetti Research Ltd, 1992. Also published in ACM Transactions on Information Systems Jan 1992.

[7] Ed. Gray Girling. Atmos 111 reference manual. Technical report, Olivetti Research Ltd, 1995.

[8] Andy Harter and Frazer Bennett. Low bandwidth infra-red networks and protocols for mobile communicating devices. Technical Report 93-5, Olivetti Research Ltd, 1993.

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

[10] Dr Andy Hopper. Pandora - an experimental system for multimedia applications. Technical Report 90-1, Olivetti Research Ltd, 1990.

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

[12] GJF Jones K Sparck Jones SJ Young MG Brown, JT Foote. Video mail retrieval by voice, an overview of the cambridge/olivetti retrieval system. *ACM Multimedia Conference*, 1994. Presented at the Multimedia DBMS Workshop.

[13] Frank Stajano. Writing tcl programs in the medusa applications environment. Technical Report 94-7, Olivetti Research Ltd, 1994.

[14] Andy Hopper Stuart Wray, Tim Glauert. The medusa applications environment. Technical Report 94-3, Olivetti Research Ltd, 1994.