

# The Sussex Multimedia Frame Buffer

Simon F. Pearce, Mike C. Bassett, Graham J. Dunnett and Paul F. Lister  
Centre for VLSI and Computer Graphics, University of Sussex, Falmer, Brighton, BN1 9QT

## Abstract

*This paper describes work at the University of Sussex in the field of multimedia video and high resolution display control. This paper is the result of research carried out for the SMILE (SPARC Macrocell and Interface Library Elements) project, one of the coordinated set of projects within Esprit III. The overall objective of SMILE is to develop, within the Open Microprocessor systems Initiative (OMI) framework, a family of SPARC based library cells, both 32-bit processor-core and application-oriented, to provide the basis for a next generation family of microprocessors/controllers that give the European systems industry a collective edge.*

*The University of Sussex is required to design a system that provides 2-D graphics functions to support full motion video, image transformation and high resolution display control. In this paper we describe the approach taken in employing classic memory interleaving techniques together with an innovative memory access coordination approach and dedicated video hardware to provide a high bandwidth frame store, optimised for live video support.*

## 1. Introduction

As part of the SMILE project the University of Sussex is required to support and enhance the display of multiple video sources in a Windows environment. Support is provided by dedicated video display hardware. Enhancement is in two forms: performance enhancement, assisting the decompression of the video streams, and image enhancement to improve the quality of the displayed images.

The main problems associated with live video are the quantity of data involved and the processing power required. Often, dedicated hardware is employed to provide the maximum degree of performance [1], but recent improvements in sub-micron technology have brought about a rapid increase in performance of the generic CPU [2]. The advent of low cost 100 MOPS processors has reduced the need for specialised hardware and brought about a shift in the data flow bottleneck. Improvements in system bus performance have shifted the bottleneck even lower. In fact, the performance of many modern processor-intensive applications is restricted only by the performance of the memory subsystem. For video applications in particular, most data processing activities are now centred close to the frame store where the video is decompressed and the images reconstructed.

## 2. Live Video Support

Simply supporting the display of a single live video stream presents a substantial problem. When we are dealing with an undefined number of these sources in a Windows environment the problems become extreme. Clearly, even with considerable processing power available, some form of hardware support must be provided.

### 2.1 Video Compression Processing

Compression is almost always a prerequisite to video transmission or storage because of the large quantity of information required to represent each frame. The three most prominent compression standards currently used are JPEG, MPEG and H.261 [3,4]. JPEG (Joint Photographic Experts Group) was originally developed for the reconstruction of still images from a storage medium but is often used for moving images. JPEG treats each frame of a video sequence as a still image, compressing each image independently of the other images in the sequence. This provides excellent error recovery for transmission [5] but the compression ratio is poor. Both MPEG (Motion Picture Experts Group) and H.261 where specifically developed for moving images and so provide much more impressive compression ratios. MPEG is highly asymmetric, requiring a great deal of processing power to compress images, but once compressed they are easy to decode. This high latency makes it unsuitable for real-time compression but its simple decompression algorithm makes it popular in information retrieval systems [6]. The H.261 standard was developed for video phone applications and is therefore ideal for the SMILE application.

The extremely high compression ratios achieved by H.261 and its suitability for transmission over fixed bandwidth mediums make it well-suited for live communication. Compression and decompression is achieved by applying an Inverse Discrete Cosine Transformation (IDCT) which converts the image into its frequency components. Although the IDCT process is extremely processor intensive, dedicated IDCT hardware is readily available [1]. To provide even larger compression ratios additional techniques are employed which also require high processing performance. Table 1 shows the results of research carried out by Texas Instruments to determine the RISC processing requirements of H.261 CIF (352 x 288 - PAL) [7]. 946 million RISC like operations per second are required to encode a single frame and 247 million RISC like operations per second are then required to decode it, of which a major part is used to reconstruct and filter the final image. These latter stages of decompression are performed within the frame store and so for optimal performance these bandwidth intensive activities should be tuned to the frame memory and performed as close to it as possible.

Function	RISC Like MOPS
Motion estimation - block matching	608 (51.0%)
Coding mode decisions	40 (3.4%)
Loop filtering (encode and decode)	110 (9.2%)
Pixel difference	18 (1.5%)
IDCT (encode)	74 (6.2%)
Inverse IDCT (encode and decode)	192 (16.1%)
Threshold/quantisation/zigzag scan	50 (4.2%)
Bit stream encode	17 (1.4%)
Reconstruction (encode and decode)	62 (16.1%)
Bit stream decode and inverse quantisation	22 (1.8%)
Total	1193 MOPS

**Table 1 Typical RISC processing requirements for H.261**  
(taken from [7])

## 2.2 Video Windows

Windowing systems are the accepted form of Graphical User Interface on most computer systems. They provide a well established control environment for image manipulation and are ideally suited for live video [8]. A video windowing system is required to provide the same functionality for video windows as expected for any other window, namely window traversal and transposition, window resizing, multiple overlapping windows and iconisation. In addition to the problem of displaying more than one video source simultaneously the problem becomes more acute with the requirement for computer generated images, including window border drawing, text, overlay and cursors on the same display. Although only a small number of the video streams may be displayed at any one time and a percentage of the displayed images may overlap, H.261 requires that the whole frames for each stream be available as a reference for following frames. This means that a large amount of off-screen memory must be made available for storing non-visible areas of video windows.

Video image manipulation control must also be provided for brightness and contrast control, perhaps offered through a pop-up control panel. This control panel must also be able to adjust the sound level and provide an interface to create and destroy windows and reallocate video streams. Each video window should maintain information regarding all the participants of the video conference, and provide a menu based interface to change the source of the video stream being displayed in it.

## 2.3 Memory Requirements

As well as presenting a taxing processing problem, the large quantity of data involved with live video presents even greater memory problems.

### 2.3.1 Memory Capacity

As well as the usual screen space, font storage and scratch pad memory requirements associated with a normal frame store, the frame buffer memory must have the capacity to store a full decompressed frame for each active video stream. H.261 has two possible resolutions, CIF (352 x 288) and QSIF (176 x 144).

Supporting CIF would require at least 200 KBytes of YU memory space per stream in addition to the 5 Mbytes required for a high resolution display. If ten separate video streams were considered a sensible upper limit for a conferencing system then 1000 Mbytes of image memory space would be required.

### 2.3.2 Memory Bandwidth

H.261 specifies frame rates of around 30 fps. For a normal video conferencing application this generates compressed video streams of between 40 Kbit/s and 2 Mbit/s. We found that a compressed video stream of a typical video conferencing scene, showing the head and shoulders view of a relatively inanimate person, produced around 2 MByte/s of IDCT decompressed data per video stream. This has then to be reconstructed, filtered, scaled and converted to the RGB display format. This represents a further 100 MByte/s per stream. This is in addition to the bandwidth required to update the display. So for 10 video streams over 2 GByte/s of bandwidth is required.

## 3. The Sussex Frame Buffer Solution

### 3.1 System Overview

At the higher level, processor performance, local bus performance and even transmission network performance has excelled, but memory system performance has remained relatively static. In video applications it is not until the video stream is to be reconstructed that the low level processing, memory capacity and memory bandwidth requirements become large. Our approach has been to provide low level video support where it is most needed, in the frame buffer. Figure 1 shows the Sussex Frame Buffer Solution. It is made up of four distinct components, the Graphics Processor, Memory Controller, Display Controller and Bus interfaces.

### 3.2 Graphics Processor

The processing capability required by the video application is provided by four parallel graphics processors which we call Graphics Genies. The Genies are controlled by the Genie Access Controller which coordinates the allocation of processing tasks. Read and write requests as well as command information can originate from two sources, the local bus or the host bus. The host bus could be any one of the fast bus schemes that are currently available, our implementation is based on the P bus [9]. The local bus is based on an asynchronous generic MF interface for use by a local processor hosting the video application.

#### 3.2.1 Genie Access Controller

Figure 2 shows the internal structure of the Genie Access Controller. Graphics task access coordination is provided between the two buses and data tracking information is kept by the Access Coordination and Data Tracker. It is important for both buses to have equal access to the graphics facilities. Data written into the Graphics Processor is stored in one of two 256 word write FIFOs for processing. This allows full bus burst capability, minimising bus traffic and maximising system performance. All control access and command information is stored in this register bank. This will include read requests. All requested read data

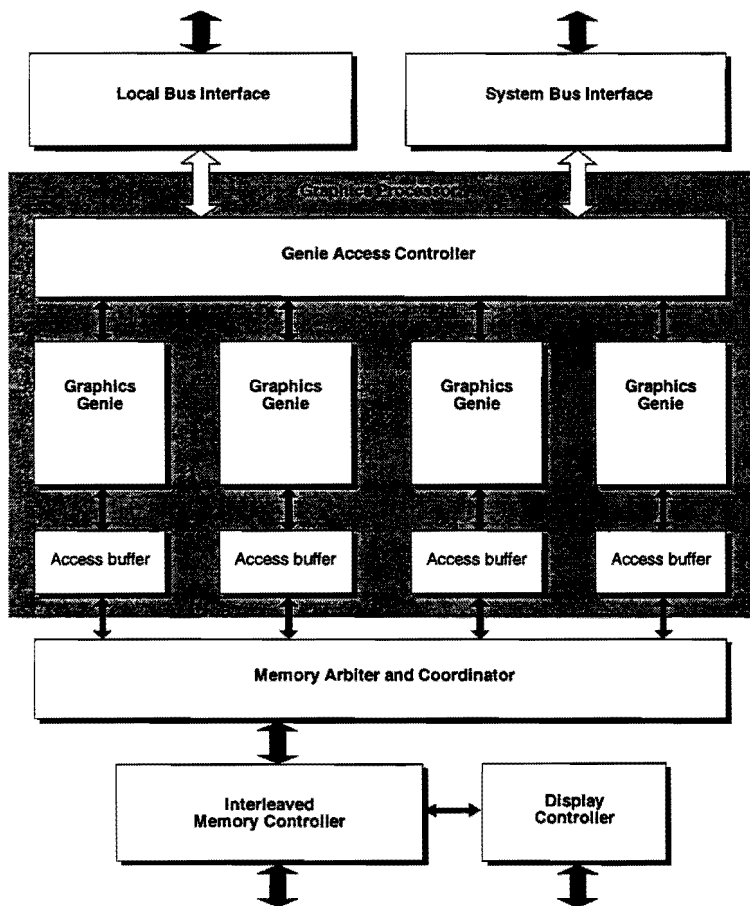


Figure 1 Architectural view of the Sussex Frame Buffer

transferred to the appropriate read FIFO and data origin information is stored in the register bank for data correlation. The Task Allocation Unit divides individual and multiple tasks equally between the Genies depending on their current status. Tasks involving video decompression are assigned stream identifiers, if that stream is not being displayed or that window is covered then the command is ignored. This serves to reduce unnecessary processing operations.

### 3.2.2 Graphics Genie

The arithmetic functionality of the Graphics Processor is provided by the array of four Graphics Genies. These devices form the heart of the Sussex Frame buffer Architecture. As Figure 3 demonstrates, the basic structure of a Genie is that of a multi-source BitBLT unit, with three address generators and an Arithmetic Unit. The Genie is more powerful than a normal BitBLT unit for two reasons, first it offers exceptional arithmetic performance and second it is capable of performing complex internal multi-pass calculations.

#### Control

The Command Interpreter decodes the graphics commands and along with the state-machine controls the flow of data through the Genie. Optimum performance is achieved by bursting data directly into the internal Block Buffer either from a system bus

or from the memory. This data can then be operated upon internally with no further memory cycles or bus operations until the calculation is complete. Up to 64 words can be loaded in one burst.

#### Arithmetic

The Arithmetic unit on-board each Genie incorporates a 32 bit adder/subtractor, a 32 bit floating-point multiplier, two comparators, a 64 bit accumulator and a set of logical operators. Once data has been loaded into the Genie the arithmetic unit can offer the equivalent of 50 MFLOPS peak (1 FLOP/clock cycle with a 50 MHz clock), 200 MFLOPS total, 133 MFLOPS sustainable, allowing for memory access and arbitration delay. The microcoded architecture offers a complete range of video support functions including YUV to RGB, image scaling, image filtering, image clipping, line drawing, colour expand and a whole range of block transfer operations used for motion estimation, area fill, area copy and text BLT. The Genie also offers the facility to perform calculations on data directly from the internal memory or system bus but the performance for multi-pass operations is greatly reduced.

#### Memory

Each Genie interfaces to the memory through eight multipurpose buffers. Each of these buffers holds an array of information for its associated memory bank. This information includes up to four double words of write data and associated addresses and four read addresses. These registers can be used as a temporary store for arithmetic

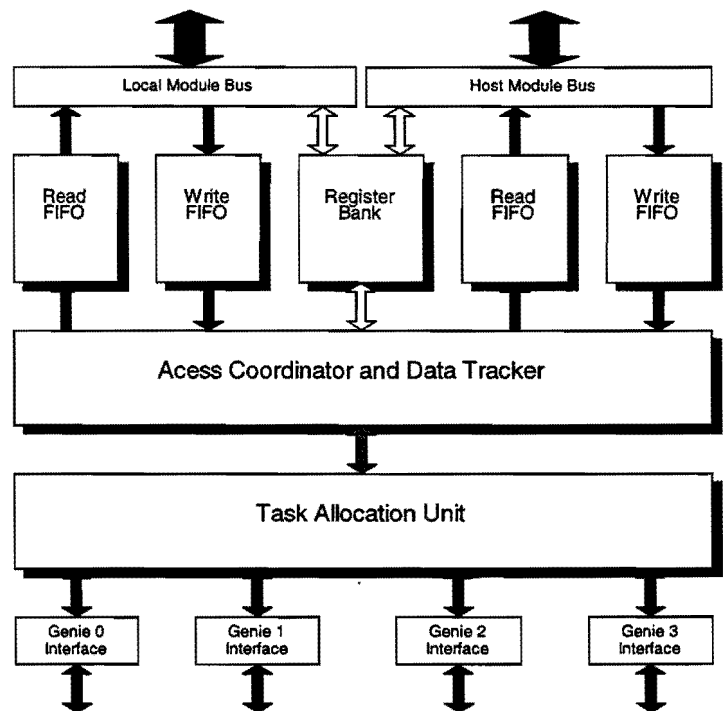


Figure 2 Genie Access Controller Architecture

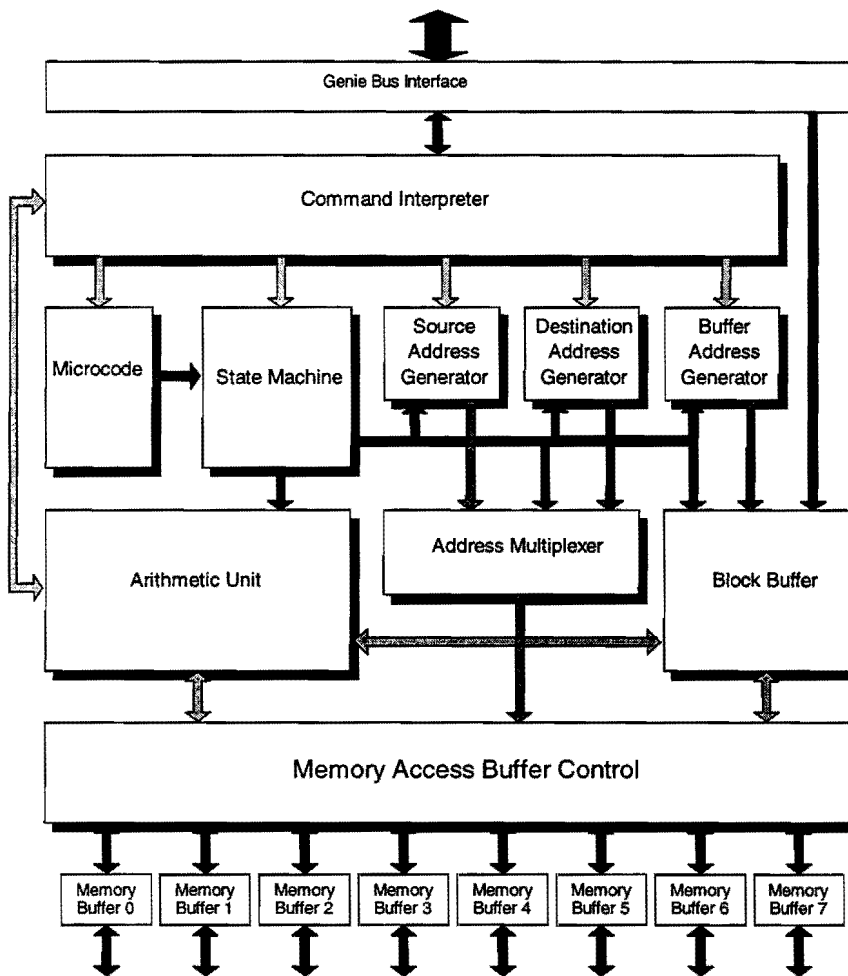


Figure 3 Graphics Genie Architecture

operations. Each buffer also holds memory access information, including page mode identifiers, memory access mode indicator and priority counters. Each buffer has an 8 bit priority counter which is incremented each time a memory slot is missed (see Memory Controller, section 3.4). The memory access mode identifier may include special read-modify-write or block write operations as well as normal access operations.

### 3.3 Display Controller

It was decided that a maximum display resolution of 1280 by 1024 would provide sufficient screen space for the number of video streams required for a useful video conference with minimum performance degradation. True colour depth with overlay/window-type at 72 Hz refresh brings the display refresh bandwidth up to 360 MByte/s. It is not acceptable for this bandwidth to be taken away from the available frame buffer bandwidth provided for the video application, so it was decided to use dual port video memory (VRAM) for screen space. This effectively isolates the display refresh circuitry from the display update circuitry. The Display Controller Architecture is illustrated in Figure 4. The Timing Sequencer coordinates display memory accesses and formats the pixel clock to produce the correct character clock. The CRT controller provides vertical and

horizontal frame synchronisation and blank signals for the display resolution chosen. The CI controller also provides a horizontal line counter which is used by the Memory Controller to load display refresh data into the shift register of the VRAM. The CRT controller generates the required VRAM shift and enable clocks, and the RAMDAC load signals. An interrupt is used to signify that the display is in vertical retrace, the facility is used by applications that wish to update the screen space while the contents are not being displayed.

Up to eight pixels can be loaded into the RAMDAC simultaneously to provide the correct display refresh rate. The Brooktree Bt4630 supports this facility. This form of pixel multiplexing limits the possible memory configurations that can be used (see later).

### 3.4 Memory Controller

The Sussex Frame Buffer memory controller satisfies the high performance memory requirements of the live video application by using a combination of classic memory interleaving techniques and an intelligent access control strategy. Figure 1 shows how the Memory Controller fits into the Sussex Frame Buffer Architecture. The Memory Arbitrator and Coordinator provides the intelligent memory access strategy, while the Interleaved Memory Controller component provides the low level dynamic memory control signals. Figure 5 shows the memory system.

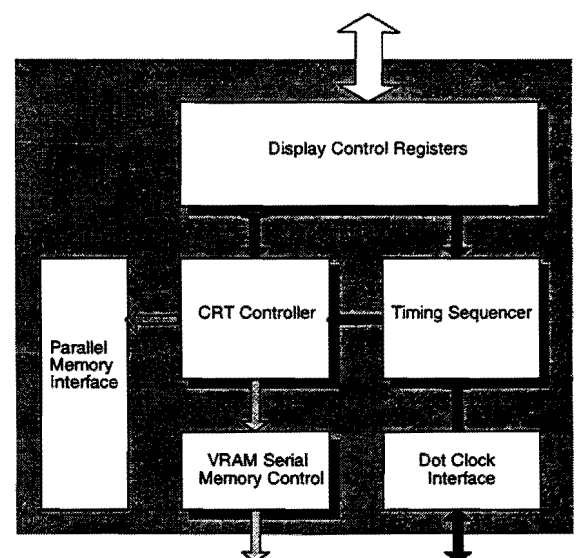


Figure 4 Display Controller Architecture

### 3.4.1 Memory Configuration

The configuration of the memory directly effects the performance of the system. The memory interface is the largest port on the chip so it is this that effects most directly the cost of the chip. With the memory playing such an important role in the design of the chip it is important to get it right. The maximum data bus size that could possibly be provided while still maintaining a reasonable overall pin count was found to be 64 bits. The rest of the required memory bandwidth is made up by the level of memory interleaving.

#### Memory Capacity

16 Mbytes of memory was found to be adequate, 5 Mbytes to meet the display requirements and the rest to meet the intermediate YUV storage requirements of 10 Video streams plus fonts, pull-down menus and scratchpad space. The chip can address an additional 16 Mbytes of DRAM which can be used for future expansion (increased screen resolution, additional video channels).

#### Memory Bandwidth

The memory bandwidth for each bank can be estimated using the following formula:

$$\frac{N \times (8\text{bytes})}{(7 + ((N - 1) \times 2)) \times Tclk}$$

N = no. of page mode accesses  
Tclk = Memory clock period

This formula takes into account the reduced memory access time and overhead associated with a stream of page mode accesses, the longer the continuous stream the closer to the maximum memory bandwidth we can get. Using a nominal 70-80 ns access DRAM (40 ns page cycle time) with a memory controller clock frequency of 50 MHz, a peak bandwidth of 200 MByte/s per bank is available, of which 173 MByte/s can be maintained, assuming an average of 16 sequential page mode accesses per bank. This gives an overall memory bandwidth of 1.38 GByte/s. Of course no memory system is expected to be able to sustain this kind of performance, but a large percentage can be utilised with an intelligent memory access strategy.

### 3.4.2 Memory Arbitrator and Coordinator

Providing a large peak memory bandwidth capability is one thing, but sustaining it is something quite different. At Sussex we have developed an architecture that formats any memory access requirement into that which provides the best performance. This formatting is supplied, with the aid of the Graphics Genie data presentation scheme, by the Memory Arbitrator and Coordinator. Figure 6 shows the general flow of bandwidth performance. If adjacent memory accesses are on different interleaved memory banks then the bandwidth capability is doubled. If the adjacent memory accesses on the same bank are on the same page then the potential bandwidth is tripled again or even quadrupled. It can be summarised therefore that the maximum potential bandwidth can be achieved if all accesses are on separate banks and all accesses are on the same page each time that bank is accessed. This is in fact exactly what the Memory

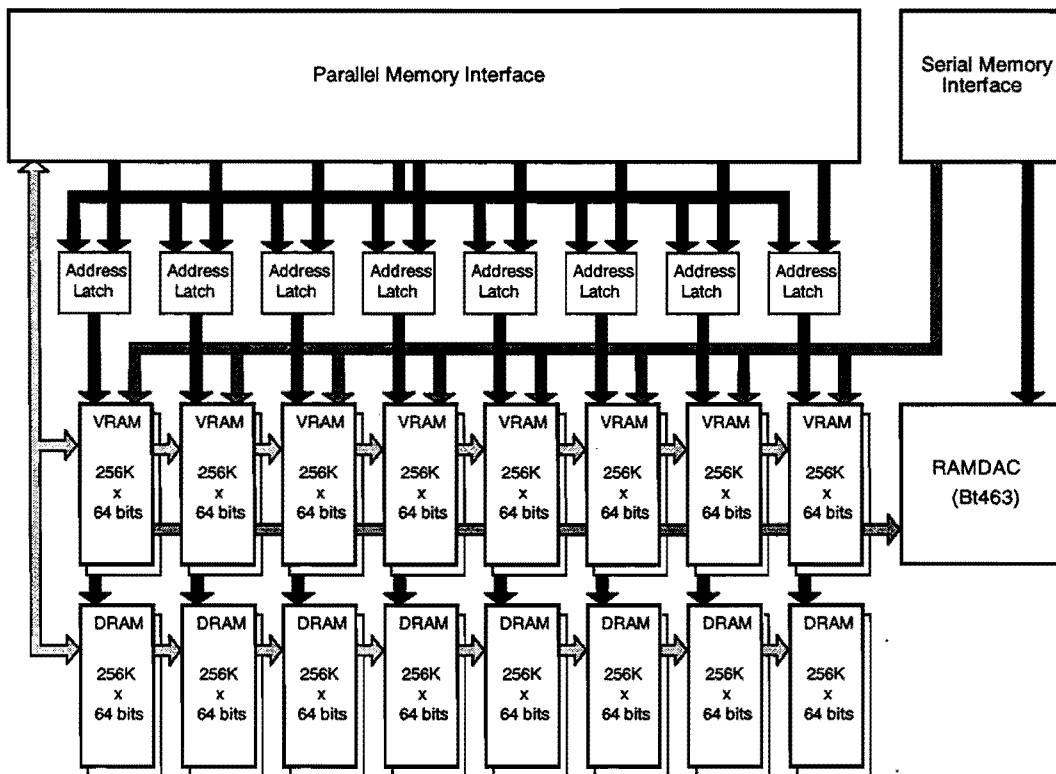


Figure 5 Memory System

Arbiter and Coordinator unit in the Sussex frame buffer tries to do. The Memory Arbiter and Coordinator unit divides the potential memory access time in to fixed length time slots. Each Graphics Genie presents the Memory Arbitrator with all the information it needs to determine which memory slot is allocated to which Genie in order to attain the maximum memory performance. Thus, by ensuring that a memory transaction is performed on every available memory time slot and assuming an average of 16 consecutive page bursts are performed between rows, the full 1.38 GByte/s of memory bandwidth can be sustained.

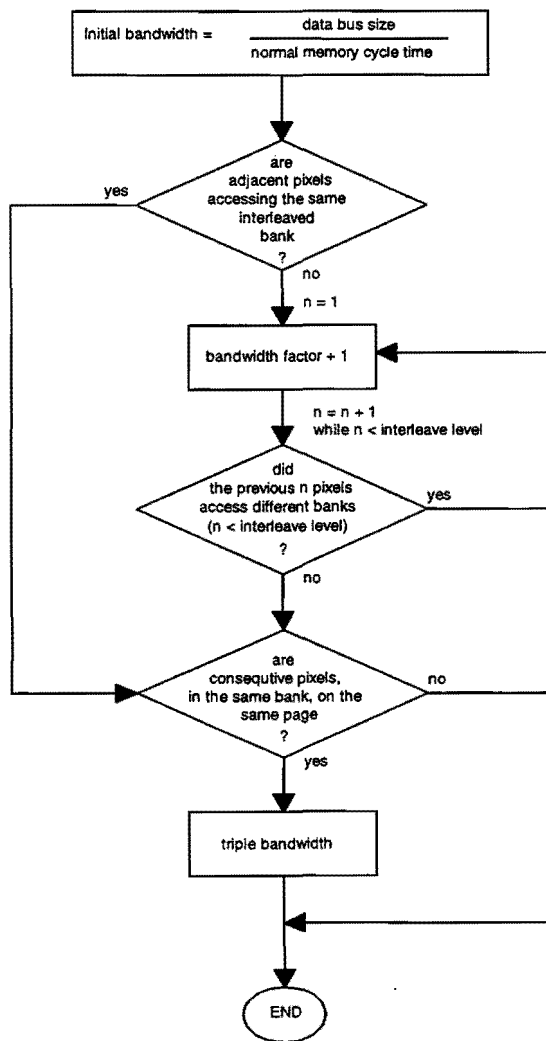


Figure 6 Memory performance algorithm

### 3.4.3 Interleaved Memory Controller

A distributed frame buffer scheme was considered but found to be unacceptable on the grounds of cost. Memory interleaving was found to be the most economical solution, providing high performance with minimum pin-out and chip count. The Memory controller supports a large number of functions that speed up

memory accesses, like the Graphics Genie, these are provided using microcode. Figure 7 shows the architecture of the Interleaved Memory Controller.

### Memory Access Modes

The Memory Controller supports a large number of memory access modes including, read-modify-write, early and late write and read both for normal operation and fast page mode operation. The Memory Controller also supports VRAM specific modes such as block write, masked write, serial transfer and serial transfer and a range of memory refresh modes.

### Memory Refresh Control

Although the display refresh controller allocates specific slots for memory refresh on each line retrace, the display controller can be turned off, so the Memory controller provides a number of additional functions used to assist memory refresh. These include DMA refresh which refreshes the entire memory. A timer counter is provided by the Memory Controller to signal the system processor when it is time for a DMA memory refresh. Many Dynamic memories have a built-in refresh address counter, usually for when a CAS-before-RAS refresh operation is performed, the Memory Controller provides its own memory refresh counter for other types of refresh.

### Interleaved Memory Control

The Memory Controller latches the full non-multiplexed address of each memory bank in its own array of external tri-state latches. The Controller then provides the appropriate signals to multiplex the address onto the dynamic memory in the correct fashion. By using these fast external latches the Memory controller does not have to hold the address while it is multiplexed.

## 4. Implementation Considerations

The University of Sussex is planning to produce a prototype of the Sussex frame buffer chip. The prototype will be large due to the high pin-count. Table 2 shows the pin requirements of each interface, a total of 262 signal pins will be required. We estimate that a further 30 power and ground pins will be necessary. The large number of buses will need many ground pins to eliminate ground bounce.

The die-size area for the prototype will also be large. However, the high pin requirements of the design mean that we are already looking at a large package so the die area is not a problem. European Silicon Structures have a 0.7µm technology which offers high integration, and is thus suitable for implementing this design. ES2 offer a 299 pin-grid array package capable of accepting a 15.51 mm by 15.51 mm die. This package is ideal for this project. Our current estimate of the complexity of the prototype is 150,000 gates. The speed of the 0.7µm technology is also suitable for the design.

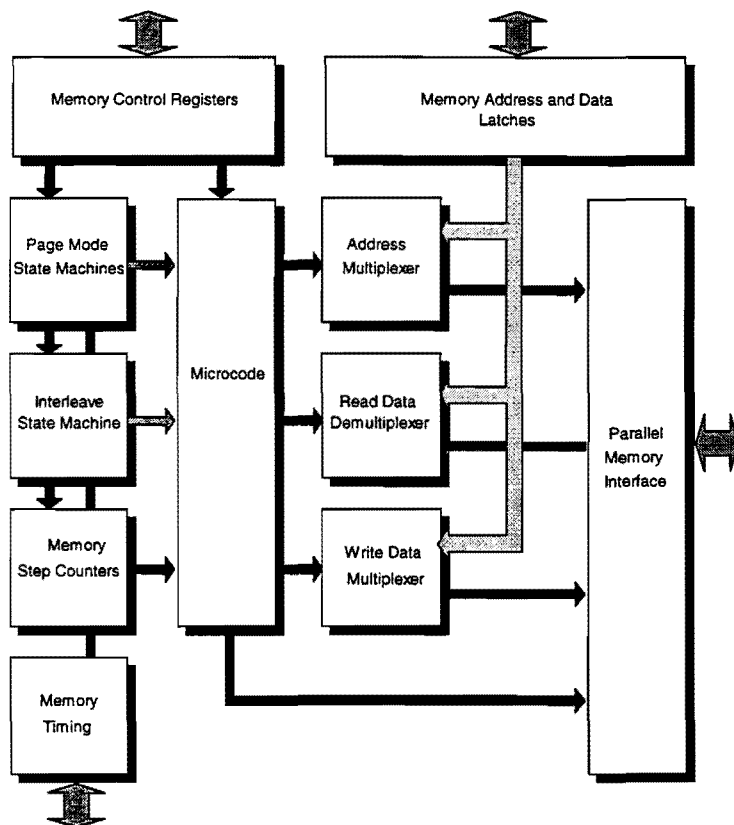


Figure 7 Interleaved Memory Controller

The prototype will be integrated into a prototype board. Our estimate of board area is 16000 mm<sup>2</sup> suggesting that the system will fit onto 1, or perhaps 2 PC boards. This confirms the suitability of the design to the PC marketplace.

### 5. Multimedia System Architecture

In addition to the frame buffer hardware solution, our approach to solving the video display problem is also based on a combined software mechanism which enables the flexibility of software control, with the acceleration provided by hardware support. Figure 8 shows how the Sussex multimedia frame buffer fits in a multimedia system. The architecture consists of a fast local bus and modular bus peripherals. The local video bus provides a flexible and powerful medium for multimedia communication.

Current bus peripherals consist of video compression/decompression, image transformation and display modules. The diagram shows an intermediate performance level implementation of the multimedia architecture. The Sussex multimedia board incorporates the video system processor as well as the frame buffer and display memory. The system CPU is in overall control of the multimedia system, communicating with the system architecture via the System bus interface. The system CPU provides high level commands initiated by windowing software running in system memory space.

### 5.1 Video System Processor

The video system processor is a powerful general purpose device similar to the system CPU (e.g. DEC Alpha, MIPS processors, etc.). The main system CPU is able to download the video processing software into the program memory of the video processor. As well as providing high level control of the video streams, the video system processor performs initial entropy and zigzag decoding, and inverse quantisation. Dedicated IDCT units are provided for primary decompression before the video streams are downloaded into the frame buffer. The general flexibility of the Sussex multimedia architecture and its reliance on software as well as hardware enables the system performance to be scaled by the amount of processing power provided and the associated hardware modules used to accelerate areas requiring high data throughput. Thus, the cost of the system can be directly controlled by the user for the application.

Interface	Pin description	Pin count
Memory	Data	64
	Address	16
	Bank	1
	RAS	8
	CAS	8
	OE/TR	8
	Write enable	8
	Special functs	16
Clock	Clock sel	4
	Dot clock	1
	Memory clock	8
PCI Bus	Address/data	37
	Interface cntrl	6
	Error report	2
	System	2
Local Bus	Data	32
	Address	24
	Addr strobe	1
	Read/write	1
	Dtack/Rdtack	1
Display	Sync	2
	Blank	1
	Serial clk	8
	Load	1
System	Clock	1
	Reset	1
TOTAL		262

Table 2 Frame buffer pin-out

### 5.2 Multimedia Network Controller

Multimedia data is received over the multimedia network by the multimedia network controller or from the video and audio capture module. Research multimedia networks provide bandwidths of 100 Mbit/s and above.

### 5.3 Video and Audio Encoder

A commercial video and audio encoding module was chosen because real-time compression can not be achieved in software on a conventional multimedia platform and is unlikely to be practical using hardware assisted software. It is also unlikely that more than one encoded video stream is required per platform, therefore a dedicated encoding chip set is most suitable as it provides a low maintenance, high power solution. The disadvantages of this approach are more rigid limits on algorithm support (although the microcode programmability of the C-Cube chip set would allow some flexibility) and the limit of one video stream. The advantage is a commercially proven encoding engine that has been highly optimised for the task of encoding a single video stream (C-Cube and SGS Thomson have achieved real-time encoding).

### 5.4 Video and Audio Output

After decompression the audio data is played out through the audio DAC, supervised and synchronised to the video sequences by the video system processor. The processor determines what post decompression operations are to be performed by the frame buffer and so is able to estimate the delay involved, ensuring that audio and video are output together.

## 6. Conclusions

The Sussex frame buffer solution is highly integrated. All image processing, display control, and memory functionality is provided in one custom circuit.

The graphics processor provides image scaling and decompression support for up to 10 simultaneous H.261 live video streams at 30 fps in a Windows environment. The parallel architecture can

provide a peak performance of 200 MFLOPS and can sustain 1 MFLOPS. This offers advantages over the use of general purpose processors. For example, the graphics processor does not require its own memory subsystem, cache controller, interrupt processor etc. Further, a graphics processor has hard wired control and does not need programming in software. For these reasons our design offers superior performance per unit of board area and superior cost per MFLOP when compared with general purpose processors.

The Display Controller unit can support display resolutions 1280 by 1024 by 32 bit pixels at 72 Hz or higher resolutions with reduced colour or frame frequency. The Display Controller provides all VRAM serial timing signals and interfaces directly to a RAMDAC with no glue logic.

The Memory Controller provides a peak memory bandwidth of 1 GByte/s. At full capacity the frame buffer can sustain an estimated 86.5% of this bandwidth using intelligent memory arbitration. This performance is achieved using 8 interleaved banks of cheap 70-80 ns dynamic memory with a 50 MHz memory clock. The single Sussex frame buffer solution represents a cost effective alternative to a distributed frame buffer, with reduced pin-out and chip count.

The general purpose architecture and microcoded functionality of our solution offers a large degree of flexibility and although its foremost application area is live video support, the functionality and performance it offers and the large memory bandwidth provides are the prerequisite requirements of any image processing application. An example alternative application might be in the support of 3D graphics, with line drawing, blending, antialiasing and filtering functions provided alongside a large frame buffer bandwidth and high resolution, full colour display.

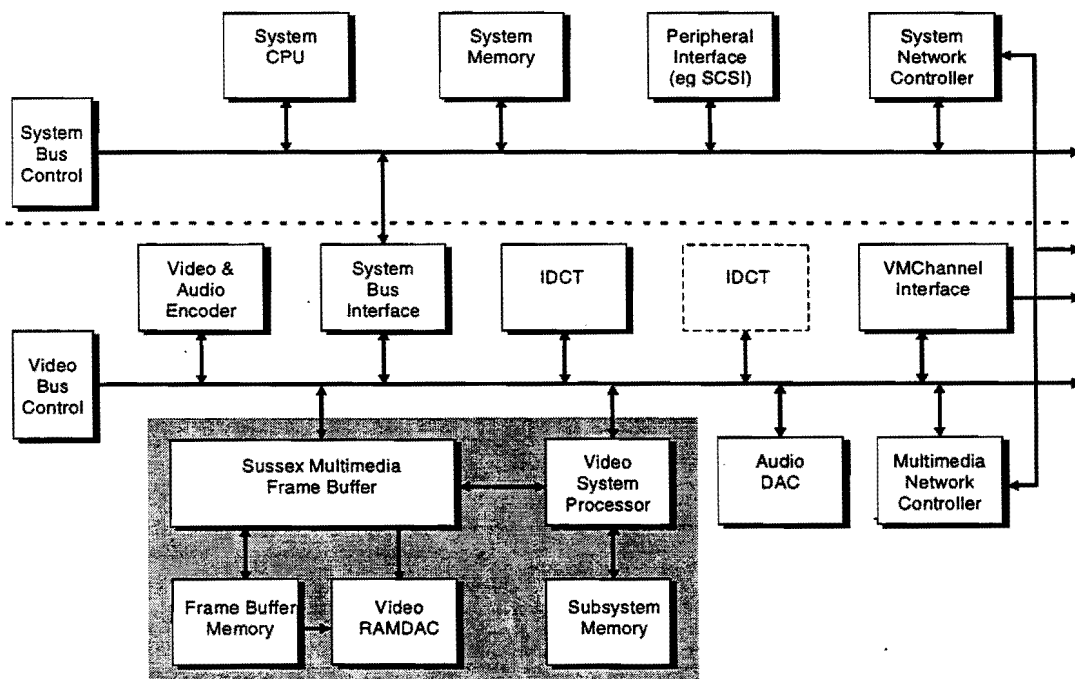


Figure 8 Sussex Multimedia System Architecture



The design of this system is well underway. Major parts of the Graphics Genie, Display Controller and Memory Controller have been written in synthesisable VHDL code using a variety of compilers. Using different VHDL compilers for the project has enforced design portability which is a primary concern of OMI. The design is expected to be available both as an ELI library macrocell and as a commercial ASIC.

#### References

1. "CCITT Video Compression Data Book," LSI Logic, September
2. T. Halfhill, "New RISC Chips For Windows NT," BYTE, vol.18, no.13. December, 1993
3. "Coding of Moving Pictures and Associated Audio," Committee Draft for standard ISO11172: ISO/MPEG 90/176 (1990)
4. "Recommendation H.261 - Video CODEC for Audio Visual Services at  $p \times 64$  Kbit/s"
5. Z.-Y. Shae and M.-S. Chen, "Mixing and Playback of JPEG Compressed Packet Videos," IBM Research Report, RC 16068, Also presented at GLOBCOM'92 (1990)
6. P. Lougher, D. Shepherd, "The design of a storage server for continuous media", Computer Journal, vol.33, no.1, 1993
7. K. Guttag, R.J. Gove and J.R. Van Aken, "A single-Chip Multiprocessor For Multimedia: The MVP," I.E.E.E. computer graphics & applications
8. H. Ichihara, T.Arikawa, "Multimedia control for desktop teleconferencing," Proc. IEEE 6th International Workshop on Telematics (IWT'91), Sept.,1991
9. "PCI Local Bus Specification," Intel Internal Copy, rev 2.0, April 30, 1993