AN ENHANCED DSP ARCHITECTURE FOR THE SEVEN MULTIMEDIA FUNCTIONS: THE MPACT 2 MEDIA PROCESSOR

Robert E. Owen, Consultant
Data/Time International
Saratoga, California US-95070
bowen@chromatic.com

Steven Purcell
Chromatic Research, Inc.
Sunnyvale, California US-94089
purcell@chromatic.com

Abstract - This paper reviews the architectural enhancements to the second generation of a VLIW media processor. The concept of a media processor is introduced and its application in an x86 family personal computer platform is described. The architectural choices made in the original Mpact media processor are explained as are how they were extended in the latest version based on design experience and changing requirements.

INTRODUCTION - THE MEDIA PROCESSOR

In an architectural sense, low-bandwidth DSP applications like digital audio and telecommunications, and high-bandwidth applications like video, graphics and image processing had travelled along different paths. Each had gone through alternate cycles of versatile highly programmable solutions to very application specific mostly hardwired solutions. With the progression to higher IC densities and speeds and the merging of their functions in applications like videoconferencing, it became apparent that a single high-performance programmable processor with one large media memory and various external codecs, as shown in Figure 1, could be designed for most of the widely used applications. Such a DSP processor became known generically as a media processor because it could do all the processing for communication and generation for the visual and auditory media, or so-called multimedia.

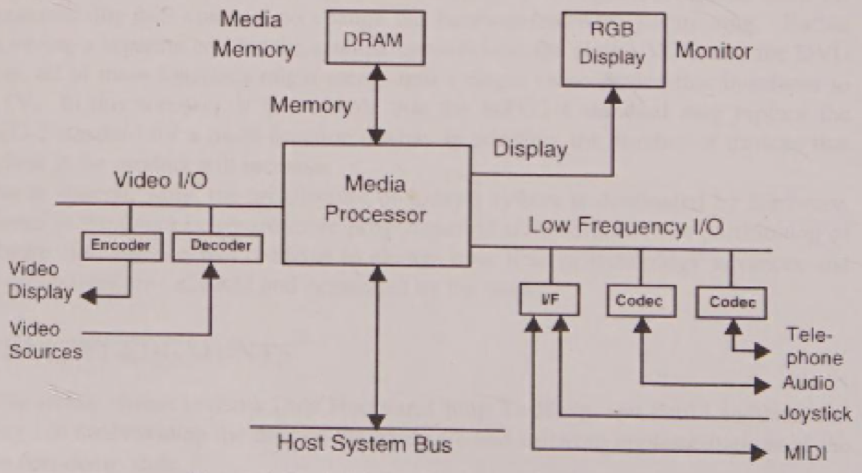

Figure 1. Basic Media Processor Configuration

0-7803-3806-5/97/$10.00 ©1997 IEEE

Generally there are thought to be seven multimedia functions: video compression/decompression through a video interface, interactive 2-D and 3-D graphics for a display interface, data/fax modem, telephony, and multichannel digital audio and music synthesis through a low frequency interface, and videoconferencing combining video, display and telephony.

Central to the media processor concept was that down-loaded programs could assure not only the easy transition from one function or functions to the next but were a requirement to allow a single design to meet the rapidly changing standards and feature enhancements of these applications.

In addition to the speed and densities of submicron IC fabrication technology, for a media processor to be commercially viable three other technologies were needed. The economy of one processor must be met with a corresponding economy in its high-bandwidth media memory, or else all cost efficiency would be lost. An additional large SRAM memory system, common with general purpose DSPs, would not acceptable. For broad usage a media processor needs a widely accepted high-bandwidth system bus. Lack of such a bus had long hampered video/graphics systems development. And not the least important by far was the need for uniform software applications interfaces. Adoption of the media processor would be slow indeed if unique software would need to be added for each of the seven functions for applications by different vendors on a variety of platforms.

THE x86 PC PLATFORM

The public PCI Bus and its associated AGP (Accelerated Graphics Port) standard with a minimum 132 MB/sec. bandwidth now provides a clear choice for a platform independent system bus. Likewise, the industry-standard Rambus memories provide a 500 MB/sec. minimum memory with the economies of dynamic RAMs and a low input/output pin count.

But only the x86 personal computer platform and its DirectX family of Applications Programming Interfaces (APIs) begin to provide the system software environment necessary to make a media processor feasible today. Figure 2 shows the natural hardware home for a media processor in an x86 system because with an attached

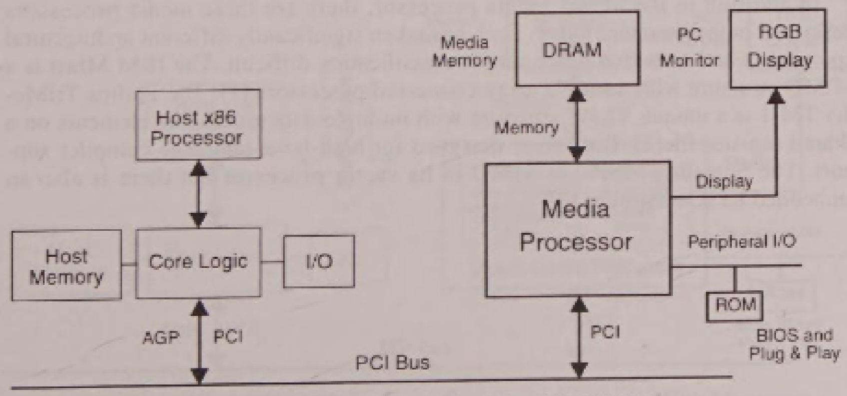

Figure 2. A Media Processor On The x86 PC Platform

BIOS ROM it can become the primary PC display monitor.

ARCHITECTURAL ISSUES

Even the relatively constrained systems architecture of Figure 2 leaves many architectural choices. A primary one is the relationship of the media processor to the host, ranging between an intimate instruction accelerator at one extreme and an isolated co-processor with discrete handing-off of data and instructions at the other. Important too is the use of MMX instruction streams in the host's floating-point processor.

Configurability is important because of the rapid change in devices and standards. Programmability can clearly accommodate algorithm changes but it is less clear about I/O and peripheral devices. The new AC'97 standardized codec and enhancements in the PCI bus are but only most recent examples of major I/O standards changes. I/O can be programmable, microprogrammable or rely on ASIC technology to reconfigure for changing needs.

Today's circuit densities allow processor architectural choices beyond just superscalar versus superpipelined or RISC versus CISC. Multiple processing units are required so their structure and control become more central. Whether to use SIMD or MIMD structures or some combination in a Very Long Instruction Word (VLIW) processor is a relevant question.

Software issues also heavily affect the hardware architecture. A major trend is to support high-level languages and to delegate to a compiler the difficult task of resource scheduling and multitasking control. A compiler-friendly structure generally requires more hardware support. Likewise real-time control can be simplified in the software at the expense of additional hardware complexity.

A separate large high-bandwidth media memory is central to the efficiency in the media processor concept. But it is only the middle portion of the memory hierarchy that extends from the processor-intimate data and instruction caches out to the host system memory and disk. Performance and silicon cost must be optimized in the memory structure choice for a broad range of possible system functions and operating scenarios.

In addition to the Mpact media processor, there are three media processors described in the literature today. Each has taken significantly different architectural approaches and ones that make simple classification difficult. The IBM Mfast is a MIMD structure with multiple array-connected processors [1]. The Philips TriMedia TM-1 is a unique VLIW structure with multiple data processing elements on a shared register file[2]. Both were designed for high-level language compiler support. The Samsung MSP-1 is SIMD in its vector processor but there is also an embedded RISC controller [3].

THE MPACT VLIW ARCHITECTURE

The Mpact family of media processors fit into the x86 system architecture as shown in Figure 3. They are an independent co-processor on the PCI bus that executes instructions from an internal cache on registered data, both of which are maintained from its own media memory. Functions are dispatched from the host as separate tasks, or through direct hardware emulation since it appears in the PCI configuration space as the primary VGA display monitor with its own bootstrap BIOS [4].

The processors are highly programmable with hardware functions tending towards the generic arithmetic except where specific configurations give unusually high performance gains like in motion estimation and 3-D rendering. Peripheral I/O processing which must be flexible but very fast is done in its own microprogrammable controller with its own writable control store. The programmed multimedia functions, mediaware, are down-loaded from host I/O sources.

The processors are designed for a mix of assembly language and compiler coding. Data caching and data organization is meant to be under tight software control without rigid protection hardware. The memory elements in the memory hierarchy are shown shaded in Figure 3.

The operating software is partitioned as shown in Figure 4 between the host x86 and the media processor. Note that individual functions, like digital audio, are shared between the two processors. Media functions are normally executed primarily on the Mpact processor, but with simultaneous operations an increasing amount will use spare cycles on the host.The MMX instructions are utilized on the host if they are available. All of this is governed by the host resident resource manager.

Operation of functions on the media processor is controlled by two real-time operating systems kernels with the faster one for the more time-critical audio related tasks. Tasks are I/O device independent, as are the majority of the I/O software. Only a lower hardware dependent layer must change along with I/O controller microcode to add a new codec to the system.

Individual multimedia tasks are invoked by the combination of high-level appli-

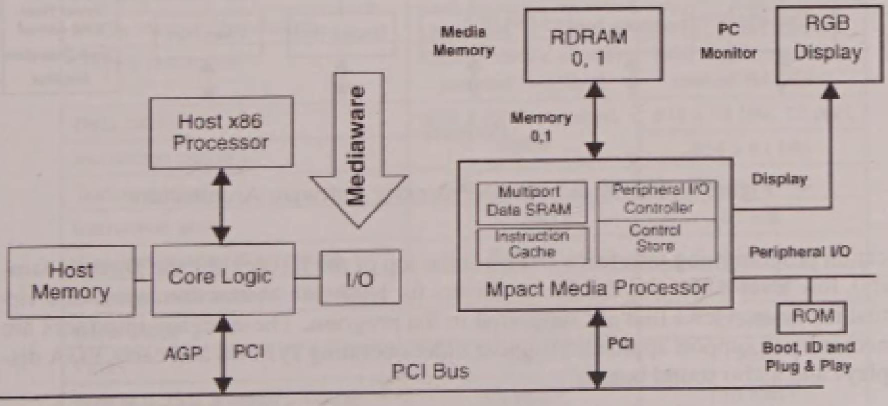

Figure 3. The Mpact Media Processor System Architecture With Memories

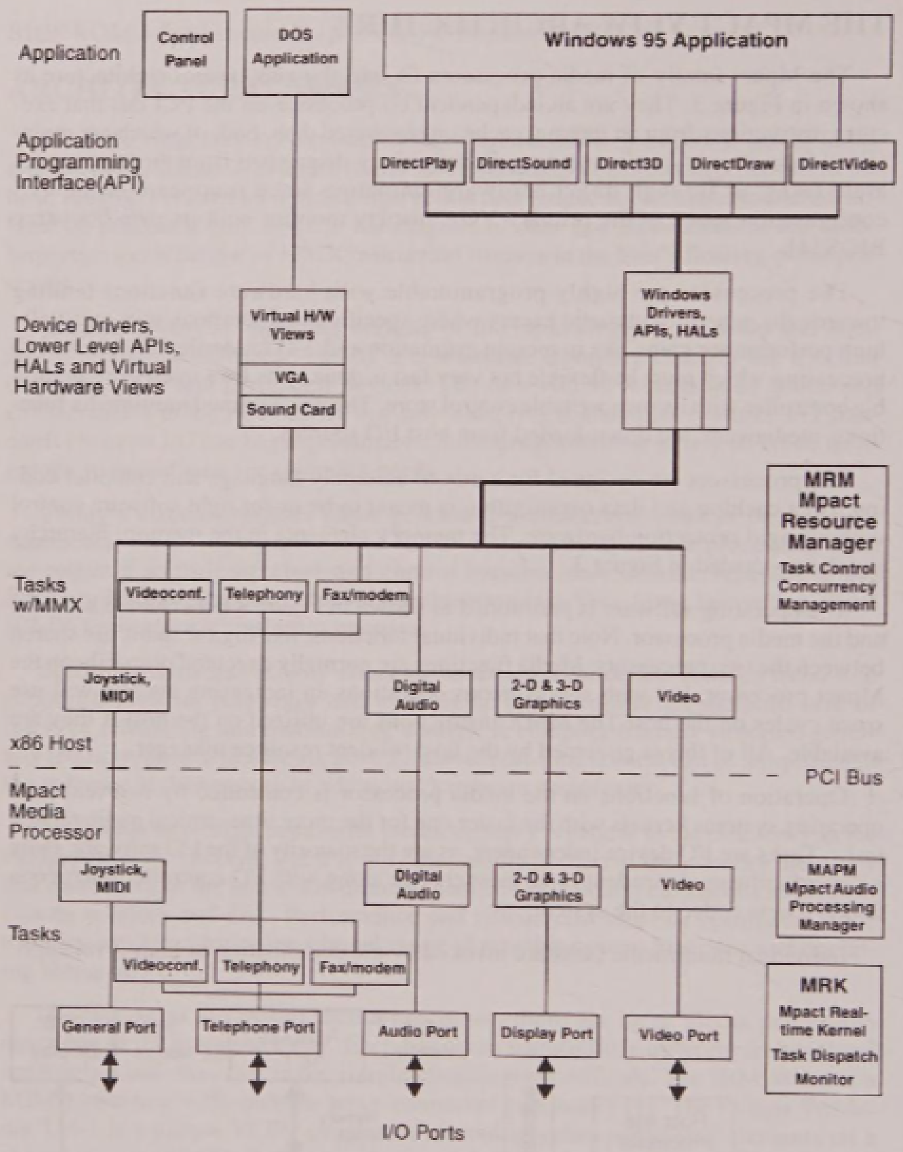

Figure 4. The Mpact Media Processor Software Architecture

cation programming interfaces (APIs) at the top of the figure (like the DirectX family), low level APIs, and the device drivers for hardware abstraction layers and virtual hardware views that are supported in the program. These legacy interfaces are necessary to support applications under older operating systems and early VGA displays and audio sound boards.

LESSONS FROM THE MPACT/3000 MEDIA PROCESSOR

The initial member of the Mpact family is the Mpact/3000 media processor [5]. It is a 3000-MIPS fixed-point processor with a single Rambus media memory with 500 MB/sec. bandwidth. With its mediaware it generally exceeds the performance in the seven multimedia functions of the high sales-volume solutions at a substantially reduced system size and cost. A cost- and performance-improved version is already available, the Mpact R/3600 media processor, that has an integral 170-MHz, 24-bit display RAMDAC and a higher clock rate for 3600 MIPS processing.

With the highest volume applications performance needs met, the architecture was reviewed and expanded to a second generation for the highest performance applications. Experience had shown that video and the 2-D and 3-D graphics for games applications tended to set the high performance standards for PCs. The initial goal was to improve these functions by a factor of at least two. Since all of the mediaware was developed internally, the full experience of code development could be utilized to improve the hardware. The Mpact 2 architecture was the result [6].

THE MPACT 2/6000 VLIW ARCHITECTURE

The first implementation of the Mpact 2 architecture is the Mpact 2/6000 shown in Figure 5. It has a VLIW central processing unit (CPU) operating out of a multiport data cache shared with four I/O controllers and two media memory controllers. In common with the original Mpact architecture it has a 72-bit data path of eight 9-bit bytes and the instruction set is a superset of the original, but as Table 1 shows many other aspects were changed.

Table 1. Architectural Feature Summary

Function	Mpact/3000	Mpact 2/6000
CPU operation rate, fixed-point	3000 MOPS	6000 MOPS
CPU operation rate, floating-point	-	500 MFLOPS
CPU execution units	5	6
PCI Bus bandwidth	132 MB/sec.	264 MB/sec.
Media memory bandwidth	500 MB/sec.	1200 MB/sec.
Display resolution	1280 x 1024 x 18 with external RAMDAC	1600 x 1200 x 24 with internal RAMDAC
Data cache	512 x 72 bits shared, 8 port	512 x 72 bits, 13 port
Instruction cache		256 x 81 bits
Texture cache	-	256 x 72 bits
Instruction word	72 bits	81 bits
Fixed-point data (9-bit bytes)	8 bytes	8 bytes
Floating-point data (36-bit words)	-	2 words
Core Clock	62.5 MHz	125 MHz
Pins	240	352
Size in 0.35 μ 3-metal process	64 mm.^{2}	115 mm.^{2}

To achieve an overall performance gain of two, critical bottlenecks had to all be cut in half. Doubling the CPU, on-chip memory and controller clock rates from 62.5 to 125 MHz was relatively easy with the 0.35μ CMOS process used. Data and instruction flows to the CPU and controllers had to be increased at least proportionately. On-chip memory cache speeds doubled but keeping them current meant doubling the media memory bandwidth. Rambus memories have currently increased to 600 MB/sec. so two separate memories were used for concurrent operation. This plus clocking separate from the CPU allows the bandwidth to more than double to 1200 MB/sec. with a cost of only an additional 19 input/output pins.

Effective memory bandwidth was further increased by adding a separate CPU instruction cache of 256 words and increasing the number of data cache ports from eight to thirteen. Further, the reduced latency of current Rambus memories makes the average effective bandwidth closer to the maximum burst rate given. The internal data bus remained 792 bits wide (11 words) with an aggregate bandwidth of 18 GB/sec.

Large display and video buffers must still be transferred with the host so the 66-MHz extension to the PCI bus is now supported. The media processor can function as a master or target on the bus with the Accelerated Graphics Port (AGP) support provided along with distributed DMA for the legacy VGA and other direct hardware registers.

A 230-MHz, 24-bit RAMDAC was integrated on-chip to support the full resolution 1600 x 1200 color displays now possible with the greater processing. External video bus bandwidth was effectively doubled by making it full-duplex. Total system cost was further reduced by integrating the MIDI UART and extending the low-bandwidth peripheral I/O bus from 40 to 83 pins, thereby reducing external interface glue logic. Direct connections to AC'97 codecs are provided.

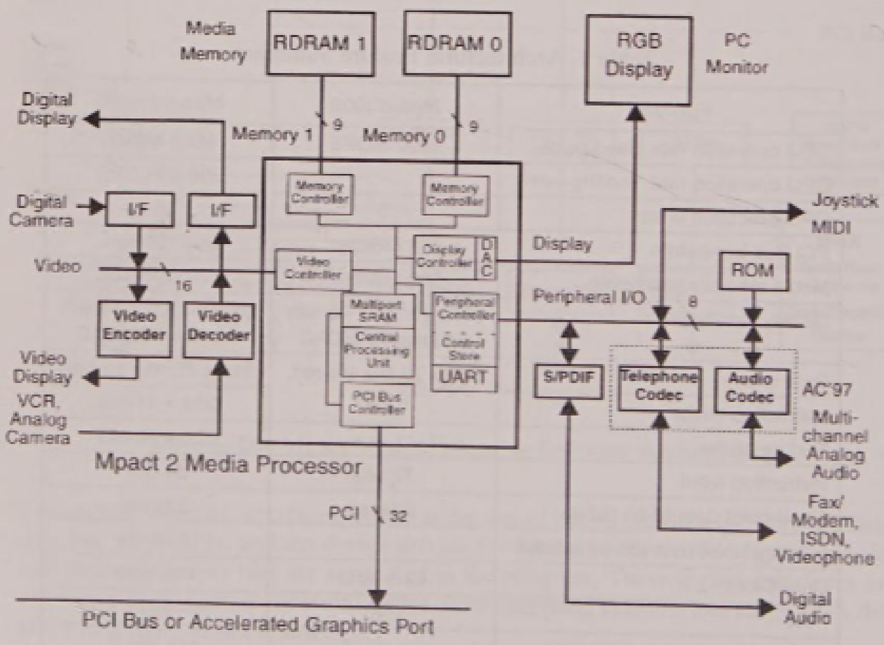

Figure 5. The Mpact 2/6000 Simplified Block Diagram

Figure 7 shows the detailed architecture. The central processing unit is composed of six execution units: four arithmetic/logic unit (ALU) groups, a motion estimation unit and a 3-D graphics unit. The CPU is a very long instruction word (VLIW) processor where each instruction word is nine 9-bit bytes. Each word is composed of two sub-instructions that are 3 to 5 bytes in length each. All CPU instructions are fetched from the 2-kilobyte instruction cache that is kept current from the RDRAM media memory. Instruction cache efficiency is high because of vector and block repeat instructions.

Instructions cause multiple operations within an ALU group and even multiple operations in multiple groups. A single instruction word can cause 8 single-byte operations to be executed on each of the four ALU groups for sustained rates of 4.0 billion operation per second (BOPS). Peak rates can reach 6.0 BOPS. Floating-point operations are shared between the four ALU groups with a sustained rate of 500 million floating-point operations per second (mflops) on the single-precision word format.

Each ALU group operates on double words of 8 bytes which can be 1, 2, 4 or 8 operands. ALU1 is a shift and align group which uses three inputs for extensive crossbar operations to produce two results. ALU2 is a general purpose ALU group with two inputs and a single output result except for special FFT butterfly instructions which produce a sum and difference result. ALU3 is a general purpose ALU group without the butterfly operations but it is augmented with three inputs for ternary operations that produce two results. ALU4 uses Booth encoders in a Wallace tree structure to produce various precisions of multiply and multiply-add operations in combination with ALU3 which outputs the results.

The motion estimation unit consists of some 400 arithmetic elements which produce an additional 40 BOPS performance for MPEG motion estimation.

The new thirty-five stage pipelined 3-D graphics unit for pixel rendering operates on data from the 2 kilobyte texture cache and immediate instruction operands. A new pixel is started every three instruction cycles and operates in parallel with other executing units. This pipeline, shown in Figure 6, which added only ten percent to the die area, can render at a one million triangle per second rate. This along with the new floating-point operations provide high-end 3-D graphic performance with an addition of about 30% to the instruction set size.

Input Polygon	Generate Spans	Generate Pixels	Perspect-ive Divide	Generate Texture Address	Access Texture Cache	Texel Format	Filter	Blend / Modulate Ptexel + Diffuse + Specular	FOG	Cluster-ing
3	3	3	10	3	1	1	2	2	2	5

Figure 6. The 35 Stages Of The Pilelined 3-D Graphics Unit

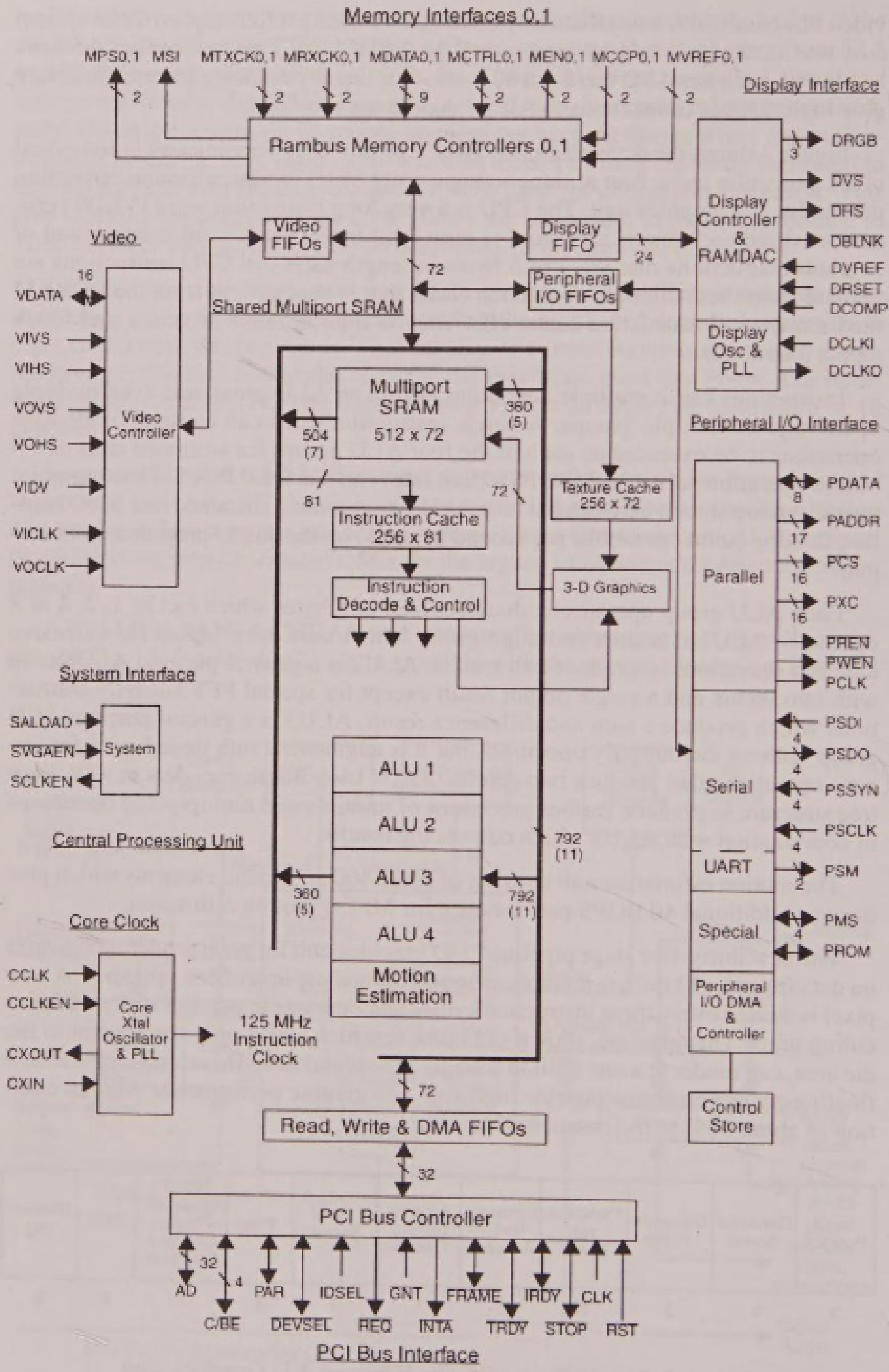

Figure 7. Mpact 2/6000 Internal Block Diagram

CONCLUSIONS

The original Mpact architecture has successfully grown to its second generation. This has happened during a period of steadily increasing demand for performance, constant change in the multimedia PC interface and function standards along with a constant need to reduce system costs, size and power. During this same period the advancements in device and memory technologies proceeded steadily but at a slower pace.

From a processor architecture standpoint, it confirms that a multiprocessor, multiple execution unit VLIW structure can be scaled upward while building on an established base of hardware designs and instruction set compatible function software.

From a system architecture standpoint it continues to confirm the concept of the media processor. It is regularly debated that enhancements to the host processor will eliminate the need for a separate media processor. Raw MIPS performance comparisons are usually made. But the availability of media processors has shown that there is a true requirement for tightly interrelated concurrent functions for video/graphics, digital audio and transmission coding/decoding. Each of these is not only performance demanding but needs the rapid real-time control the host can not provide and can only be achieved in a separate processor that is autonomous in a real-time sense.

REFERENCES

[1] G.G. Pechanek et al., "A Machine Organization And Architecture For Highly Parallel, Scalable Single Chip DSPs", *Proc. of 1995 DSP^{X} Technical Program*, 1995. pp. 42-50.

[2] S. Rathnam, G. Slavenburg, "An Architectural Overview of the Programmable Multimedia Processor TM1", *Proc. Compcon 96*, IEEE CS Press, 1996, pp. 319-326.

[3] Y. Yao, "Samsung Launches Media Processor", *Microprocessor Report*, MicroDesign Resources, Inc., August 26, 1996, pp. 1, 6-9.

[4] P. Foley, "The Mpact Media Processor Redefines the Multimedia PC", *Proc. Compcon 96*, IEEE CS Press, 1996, pp. 311-318.

[5] D. Epstein, "Chromatic Raises the Multimedia Bar", *Microprocessor Report*, MicroDesign Resources, Inc., October 23, 1995, pp. 23-27.

[6] S. Purcell, "Mpact 2 Media Processor, Balanced Times 2 Performance", *Proc. SPIE, Multimedia Hardware Architectures 1997*, Vol. 3021, SPIE, 1997, pp. 102-108.