Video & Graphics Processors: 1997

John A. Watlington
MIT Media Laboratory

May 9, 1997

This report is an attempt to assess and contrast the current state of the art in digital video and graphics processing system architectures. Although this is not meant to be a comprehensive survey, I attempted to at least quickly review most of the commercially available major architectures. At present, there is a huge amount of interest in this area, as three new markets are perceived: systems for personal (set-top) terminals, PC graphics for game and network applications, and consumer video products based around digital video disks (DVD). Due both to their proliferation, and their very limited programmability, I did not include dedicated MPEG2 video decoders and graphics accelerators limited to 2D operations (GUI accelerators) in this survey.

The surveyed systems aggregated into three distinct classes, namely:

1. Video Signal Processors (VSPs) - A programmable CPU, usually with specialized processing elements as well.

2. Graphics Architectures - Specialized processing elements, usually with a fixed interconnect (a linear array.)

3. Structured Video Architectures - Containing both a video signal processor and possibly a dedicated pipeline. Currently represented by Talisman.

A section follows for each class, describing representative systems of that class. The report closes with a subjective Commentary section, containing both comments about architectural features and a look at future scalability.

1 Video Signal Processors

Many of the surveyed systems are characterized by a single programmable core, with an instruction set architecture optimized for the operations typically encountered in either graphics or video processing algorithms. Specialized processing units may be incorporated either into the programmable core, or as co-processors.

Increasingly, general purpose microprocessors are integrating instruction set extensions allowing the processing of multiple small datum packed into a larger word simultaneously [1] [2] [3] in order to gain a slight speed improvement (x2 or x3) in graphics tasks. The minimal costs of these *group* instruction extensions (increased instruction set complexity, and slightly increased carry propagation delay in the ALUs) compared with the speed improvement when processing typical images justify the extensions. While support for group instructions alone was generally not sufficient for inclusion in this survey, they are a common architectural feature among the video signal processors discussed.

System	TI 'C6201	Philips TM1000	MicroUnity MediaProc	Chromatics MPact2	Samsung MSP	Units
Inst issue	8	5	1	2	1	inst
Inst Width	256	220	32	72	32	total bits
Inst Cache	512K	256K	256K	18K	16K	bits
Data Cache	-	128K	256K	-	40K	bits
Data Ram	512K	-	-	-	-	bits
Registers	32x32	126x32	320x64	1Kx72	31x32	bits
Data Width	32	32	64	72	32	bits
Func. Units	8	22	4	4	1	
Floating Pt	No	Yes	No	Yes	Yes	
Spec. Units	-	2	1	2	2	

Table 1: Video Signal Processors Architecture Comparison

1.1 Texas Instruments VelociTI TMS320C6201

The TMS320C6201 [4] is the first implementation of a new VLIW architecture, VelociTI [5] [6], just available from TI's DSP group. While it contains no functional units dedicated to video or graphics, it's raw performance makes it interesting. The processing core (currently running at 200 MHz) consists of two each of four types of processing elements: logical (and arithmetic), arithmetic, data addressing, and multiply. All processing elements operate on 32b data except for multiply, which only operates on 16b input values. Up to eight instructions (one per processing element) may be issued in a single cycle. The processing units are grouped into two sets of four, each of which is coupled to a 16x32b register bank. Ten read ports and five write ports are provided per bank. A cross-linked register read port provides direct communication between the two processing clusters, which share a common instruction issue unit.

No data cache is provided (although the architecture allows for one). Instead, directly addressable data RAM is provided on-chip, organized as four interleaved banks of 128 Kbits (8Kx16b) each. Each processing cluster has a separate 32b data path to the data RAM, and two simultaneous 32b accesses may occur if different banks are accessed. A 32b memory interface which can address 64 MBytes of the architecture's 4 GBytes is provided, along with a 16b host interface which is limited to addressing the internal register, data and 512 Kbits of instruction RAM. The memory interface is capable of running at clock rates equal to the processor core, and is designed for synchronous DRAM, synchronous burst SRAM, or async. SRAM. Two channels of DMA are provided on-chip for efficiently moving data around the system.

The instruction set architecture is spartan (esp. in the context of this survey). In the TMS320 tradition, it contains concessions to typical signal processing needs: saturation instead of numeric overflow, a normalize instruction (`norm`), support for circular addressing, and extended precision (using two 32b registers for a 40b value). There is no support for group instructions, or floating point. The execution of every instruction is conditional. Three bits of each 32b fixed-size opcode indicate a guard register, and a fourth bit indicates whether the execution is conditional upon the guard being equal or not-equal to zero.

The instructions are grouped by the compiler or optimizing assembler into execution groups to be issued in the same cycle. The LSB of each opcode is dedicated to indicating the last instruction in an execution group. One or more of these groups are packed into an eight

Unit	Number	Unit	Number
Int. ALU	5	Load/Store	2
DSP ALU	2	DSP Mult.	2
Shifter	2	Branch	3
Int/FP Mult	2	FP ALU	2
FP Compare	1	FP Sq.Root	1
Constant	5		

Table 2: TM1000 Functional Units

instruction (256b) fetch group. An execution group may not overlap a fetch group boundary. The instruction RAM (which may be configured as a cache) uses a 256b bus to provide a fetch group to the CPU every cycle.

1.2 Philips TriMedia TM1000

Unlike the TMS320C6201, the Philips TriMedia TM1000 [7] [8] [9] integrates Media specific co-processors and a VLIW core with extensions for media processing. It was designed to be a PCI based media co-processing system in a PC. Based on the LIFE-1 VLIW architecture developed at Philips Research Labs, Palo Alto, in 1987, it started development as a product in 1994, and the TM1000 (previously named the TM-1) shipped in early 1996. A second generation is now in development.

The VLIW core contains a large number [1] of functional units, connected to a 128x32b register bank through 15 read ports and 5 write ports. A listing of the functional units is provided in Table 2. Up to five instructions may be issued in a single 10 nS cycle. The instruction set is large (197), having been extended to support the specialized functional units and floating point. Group instructions are supported, operating on four 8b or two 16b values in a 32b word, as well as packing and unpacking instructions. Specialized instructions are provided for performing convolution and vector distance (i.e. motion estimation) calculation. As in the TMS320C6201, the execution of almost every instruction is conditional upon a guard register.

The VLIW core is connected to a 128 Kbit data cache (8-way associative) using two 32-bit buses. A separate 256 Kbit instruction cache (also 8-way associative) uses a 220b bus to provide five instructions per cycle to the CPU. The instruction stream is stored and cached in a compressed format, and decompressed to provide the 220b instructions only upon being fetched. The data and instruction caches share a single 32b main data bus (the Data Highway) with all the co-processors and peripherals on the chip. The Data Highway connects to both a 32b PCI bus interface (master/slave), and a memory interface (32b) to off-chip synchronous DRAM. The architecture address space of 4 GBytes is fully supported throughout the system.

The Image Co-processor is a pipeline of specialized processors designed to perform typical image manipulations (at 50 Mpixels/sec. peak) independently of the VLIW core. It reads its parameters and image data from SDRAM memory using the Data Highway, and writes its output either back to SDRAM or to a destination on the PCI bus. A set of FIFOs (6 x 512b) are provided at the input to the co-processor, feeding a 5-tap polyphase 1D FIR filtering unit. The filtering unit processes a single 8b channel at a time, using multiple passes to perform operations on multiple color channels. A YUV/RGB converter is next, followed by an alpha-

[1] Philips' count of 27 functional units includes 5 constant units which basically serve as ports for accessing immediate values in the instruction stream. See Table 2.

System	TI 'C6201	Philips TM1000	MicroUnity MediaProc	Chromatics MPact2	Samsung MSP	Units
Clock	200	100	300	120	50	MHz
Chip/ICache	6.4	3.2	38.4	8.6?	6.4	Gbits/sec
ICache/Proc	50	22.4	38.4	8.6	1.8	Gbits/sec
Proc/Reg	196	64	154	104	154?	Gbits/sec
Reg/DCache	12.8	6.4	77	-	51	Gbits/sec
DCache/Chip	6.4	3.2	38.4	104	6.4	Gbits/sec
Chip/System	3.2	4.2	35	10.4	4.2	Gbits/sec
Reg Size	1	4	10	73	1+	Kbits
DCache Size	512	128	256	-	40	Kbits

Table 3: Video Signal Processors Bandwidth Hierarchies

blending unit (if used, a separate background image is also input), and an output formatting stage. The Image Co-processor is microprogrammed, allowing it to be reconfigured for different data formats and functionality.

A Variable Length Decoder, designed to decode MPEG and MPEG2 system bitstreams, is also provided on-chip. Like the Image co-processor, it contains DMA controllers for reading and writing data from the SDRAM. Both co-processors synchronize with the VLIW core by interrupting it. Other peripherals incorporated on-chip are CCIR-601/656 video input and output (the video output incorporates one last alpha-blended overlay), digital audio I/O, and two serial interfaces (I^2C and V.34/ISDN).

1.3 MicroUnity MediaProcessor

The use of a single programmable CPU core to perform ALL operations in a system is a cornerstone of the MicroUnity MediaProcessor architecture. From the perspective of this survey, the interesting architectural features of the MediaProcessor are:

- The use of group extensions to the instruction set, supporting 2x32, 4x16, or 8x8 groups.
- The support for arbitrary bit shuffling & shifting.
- The relatively high I/O bandwidth supported.

Several good short introductions to the MediaProcessor are now available [10] [11], so none will be given here.

MicroUnity has ceased trying to fabricate its own chips in 0.5 μm BiCMOS (with a 1 GHz clock rate), and is now solely targetting CMOS implementations [12]. The numbers given in the tables are for the Chronus CMOS implementation of the MediaProcessor. The specialized processing element referred to in Table 1 is the unit supporting "extended mathematics": Galois field multiply and polynomial multiply/divide.

1.4 Chromatics MPact2

Chromatics has been shipping the Mpact1 [13] [14] (several versions) since Sept. '96. They have now announced the second generation in the architecture, Mpact2 [15] [16] which has more off-chip memory bandwidth, new fab. technology w. faster clock rate, larger data RAM with

System	TI 'C6201	Philips TM1000	MicroUnity MediaProc	Chromatics MPact2	Samsung MSP	Units
Fab. Tech.	CMOS	CMOS	CMOS	CMOS	CMOS	
Line Size	0.25	0.35	0.6	0.35	0.5/0.35	μm
Metal Lyrs	5	4	3	3	?	
Clock	200	100	300	120	50/100	MHz
Voltage	2.5	3.3	3.3	3.3	3.3	V
Package	BGA	BGA/QFP	BGA	QFP	?	
Pins	352	240	441	304	128/256	pins
Power	4.2	4	?	?	4	W
Area	270 (?)	?	250(?)	?	?	mm^2
Ext. Mem.	SDRAM & SBSRAM	SDRAM	SDRAM	RDRAM	SDRAM	

Table 4: Video Signal Proc. Technology Comparison

Unit	Ports In	Ports Out	Floating Point ?	Description
ALU1	3	2	Yes	Shift & Align, Juggle
ALU2	2	1	Yes	Logic, Arith, supports FFT butterfly
ALU3	6	2	Yes	Logic, Arith, 3-input ops on 144b words
ALU4	2	2	Yes	Wallace tree for Multiply
ALU5	1	1	No	Motion Estimation
ALU6	1	1	No	Graphics Pipeline

Table 5: Mpact2 Functional Units

more ports, and the addition of an instruction cache and a specialized processing pipeline for 3D graphics.

The Mpact2 processor contains a VLIW core capable of issuing one or two instructions packed into a 72b word per cycle. Instead of separate data cache and registers, Mpact2 uses a single 73 Kbit (8Kx72b) bank of RAM with 6 read and 6 write ports. This central multiport memory is accessed through an 11-port crosspoint by the six functional units, a 32b PCI interface, a CCIR601/656 video I/O interface, random peripherals, and two Rambus interfaces. Rambus specifies 9b memory devices (for parity purposes) — the Mpact architecture uses the ninth bit for additional precision, giving data sizes of 9b, 18b, 36b, and 72b.

The two specialized functional units (see Table 1.4) are targeted at accelerating 3D graphics and motion estimation (vector distance). The graphics unit is a 35-stage scan conversion pipeline, capable of rendering 50Mpixels/sec. The pipeline performs 18b z-buffered compositing, Gouraud shading, perspective and texture mapping. An 18 Kbit texture memory is provided as part of this unit. The motion estimation unit is capable of computing the vector distance between two 128-element vectors per cycle (8b elements.)

The programming model/instruction set architecture of the Mpact2 is proprietary — Chromatics develops all firmware (currently providing drivers for accelerating Microsoft products through DirectX.)

1.5 Samsung Media Signal Processor

The Samsung Media Signal Processor [17] [18] consists of a conventional 32b RISC core (ARM7) coupled with a custom vector processor. The ARM7 instruction set architecture contains explicit support for up to 16 co-processors. Dedicated synchronization signals and test instructions are provided to signal the completion of a co-processor instruction to the ARM core. Both the core and the vector co-processor share a cache subsystem (40 Kbits of data cache and 16 Kbits of instr. cache.) A 64b bus connects the cache subsystem, a 32b PCI bus interface, and an optional memory controller connected to external SDRAM (32b data bus).

Little has been published about the vector processor. It is described as a SIMD architecture, and from the performance figures cited [2] it probably contains sixteen 32b processing elements. It supports group instructions, and like the Chromatics Mpact it supports a 9b data type, although only internally. The vector processor is supplied with data through two 256b buses from the shared cache subsytem. A separate MPEG2 bitstream processor is also provided on-chip.

1.6 Honorable Mention

There are several processors which deserve to be included in the above group, but which for one reason or another (mostly time) weren't described or considered. Four of these, the Mitsubishi D30V [19], the Fujitsu MMA [20], the Sony Video DSP [21] and the C-Cube Video RISC 3, are similar to the processors described above.

A fifth, the Texas Instruments TMS320C80 [22], is a five-way MIMD architecture that has been available for several years but has not seen widespread market acceptance. This is probably due to the difficulty in parallelizing application code for execution on the 'C80. Witness TI's introduction of the 'C6201, which isn't appreciably more powerful, but easily supports the available parallelism through a simple source code recompile.

2 Graphics Architectures

Another set of systems examined were those which used specialized processors connected in a dedicated pipe architecture. While evolved from the traditional graphics pipeline, these systems are generally capable of limited video processing, viewing the pixel rasters as texture mapped polygons.

2.1 Silicon Graphics Infinite Reality

The Silicon Graphics Infinite Reality graphics subsystem [23] [24] was designed for a different market than the other systems assesed: one in which performance is more important than cost. The Infinite Reality is a second generation of the Reality Engine architecture [25], which not only takes advantage of technological improvements but also increases the range of scalability.

The Infinite Reality is available on the Onyx2 workstations [26]. These systems contain from 1 to 24 MIPS R10K processors, each with 512 Kbits of primary inst. cache, 512 Kbits of primary data cache, and 32 Mbits of secondary cache. The main system memory is configurable from 512 Mbits up to 64 Gbits. From one to eight rendering pipelines are available on a workstation. Each rendering pipeline contains four Geometry Engines, and may be configured with one to four Raster Managers. From two to eight display "channels" may be provided per graphics pipeline, each generating an RGB signal at up to 1920x1200 60 fps non-interlaced.

[2] At 100MHz, the vector processor is supposedly capable of 6.4 billion 8b integer operations per sec., 3.2 billion 16b int. ops/sec, or 1.6 billion 32b floating point ops/sec [18].

	Chromatics MPact2	Talisman Escalante	GLINT & PerMedia	Infinite Reality	Units
Geometry	1	0.8	1.0	11	MTriangles/sec
Rasterizing	50	150	30	780	MPixels/sec

Table 6: Comparative Graphics Performance

Geometry Subsystem The Geometry Engines (GE) are the subsystems responsible for performing polygon-to-triangle decomposition, geometry transformations and screen space projection. Other functions subsumed by the GE is image manipulation: rotations, warps, interpolation, decimation, filtering, and statistics measurement. Each GE utilizes a custom processor consisting of three SIMD processing elements, consisting of a dedicated register file and a floating point multiplier and arithmetic unit. The processing elements share a common multiport SRAM, and all GEs in a system share a 2.9 Gb/s interface to main system memory. The processor operates at 90 MHz, and is controlled using a 195b micro-instruction. The micro-instructions are compressed, yielding average 2.7:1 reduction in size with minimal (1.5%) performance impact [24].

The transformed and projected triangles must be distributed to the appropriate Raster Manager (which use image space subdivision to provide parallelism.) This is done using a shared 3.2 Gb/s linear interconnect, the Triangle bus.

Raster Subsystem From one to four Raster Manager boards perform the triangle scan conversion, texture mapping, and rasterization. A single copy of the texture memory is stored on each Raster Manager board. Using 128 SDRAM devices (2048b wide) provides a texture read bandwidth of around 15 Gb/s per board and a total texture capacity of 128 – 512 Mb. The frame buffer memory is distributed for maximum read/write bandwidth: up to 320 rasterizing processors are used, each connected to a dedicated 256Kx32b SGRAM. The peak frame buffer access rates are in excess of 600 Gb/s.

2.2 Honorable Mention

A number of integrated 3D graphics pipelines have appeared recently. One architecture is the Permedia (rasterizer) & GLINT Delta (triangle setup) combination from 3Dlabs [27], of which the Texas Instruments TVP4010 [28] is a licensed implementation.

My favorite, however, would have to be the Nintendo64. It contains an 94MHz R4200 processor, coupled with a dedicated graphics pipeline (the Reality Co-processor) from Silicon Graphics. Using two Rambus RDRAMs (32 Mb total) for memory, the Reality Co-processor renders 30 fps at a resolution of 640x480 using perspective projection, z-buffering, multi-resolution texture (MIP) mapping, environment mapping, and a form of anti-aliasing. And all at a price point of $200/complete system.

3 Structured Video Architectures

Microsoft has proposed a reference architecture for graphics and multimedia, named Talisman [29] [30], which is intended to provide a high level of performance with a minimum of memory and hardware. Talisman is based on four concepts:

- **Composited Image Layers** - The scene database is decomposed into layers of non-interpenetrating objects, which are rendered independently then spatially remapped (if necessary) and composited for display. Talisman carries this further by specifying that the compositing is to be done in small strips of the image, as they are required for display, thus eliminating the requirement for a frame buffer.

- **Compression** - Both memory and I/O bandwidth are minimized through the compression of all off-chip image data, including textures. Uncompressed image data is represented using 32b (RGB plus alpha). Compressed data is represented using TREC, a JPEG-like algorithm using DCT coded 8x8 blocks, but without ADPCM coding of the DC coeff. There is no difference of representation in the Talisman architecture between textures, image layers, and images — TREC is ubiquitous unless image data is actively being used in a computation.

- **Chunking** - In this mechanism, more properly identified as "virtual buffering" [31], the rendered image space is divided into a number of equal sized regions (32x32 in Talisman), and all included objects are rendered into a region at one time. This requires a sorting/clipping stage to determine which objects to render into a particular virtual buffer. It was determined that clipping to an enclosing volume four times the size the virtual buffer (and thus reusing the clip over several regions) was sufficient. [29]

- **Multi-pass Rendering** - By requiring that the output of the renderer be available for use as input to a later rendering stage (either as texture or background) Talisman is capable of supporting a number of lighting, shadow, and environmental effects.

The Talisman architecture is justified by its authors as supporting incremental, high quality rendering at a minimal system cost. What isn't touted is the ability to effortlessly integrate rendered and real images in the system. The composited image layers and multi-pass rendering are typical of a structured video system — Talisman implements the processing pipeline for compressed structured video proposed by Bove, *et al* [32].

3.1 Escalante

Part of the Talisman architecture proposal is a reference implementation, Escalante (code-named Touchstone)[30] [29] [33], aimed at the high end of the consumer PC market. This PCI-based implementation consists of four major functional blocks (each in a separate package). One or two Rambus DRAMs (8–16 Mbits) are used, incorporated into a single functional block, to meet all system memory needs.

The Media Processor The first functional block in the system is a programmable "Media DSP", responsible for video codecs, audio processing, and graphics front-end processing (geometry transformations and lighting processing.) Any of the Video Signal Processors introduced above is capable of filling this position. The reference design suggest either the Samsung MSP or the Philips TriMedia processor as suitable examples.

The Polygon Object Processor The second functional block in the system is a Polygon Object Processor (POP), responsible for rendering the transformed polygons passed to it by the Media processor. The polygon rendering pipeline supports texture mapping (with anisotropic texture filtering), anti-aliasing, and z-buffered hidden surface removal. It renders a single 32x32 pixel virtual buffer at a time, which are then compressed and stored in system memory for later

Unit	RAM (bits)	Area ($M\lambda^2$)	Unit	RAM (bits)	Area ($M\lambda^2$)
Rambus I/F		169	Memory I/F	12K	58
Clip & Scan Convert	57K	764	Decompression	16K	195
Texture Addr.		290	Texture Cache	71K	356
Compositor	137K	654	Compression	32K	477
Testability		215	Routing		318
I/O Pads		708			
			Total	325K	4,200

Table 7: Escalante Polygon Object Processor Block Areas

Unit	RAM (bits)	Area ($M\lambda^2$)	Unit	RAM (bits)	Area ($M\lambda^2$)
Layer Prefetch	4K	220	Decompressor	25K	685
Image “cache”	71K	600	Filtering		134
Composite Ctl.		85	Testability		215
Routing		152	I/O Pads		555
			Total	100K	2,640

Table 8: Escalante Image Layer Compositor Block Areas

fetch by the Image Layer Compositor. The POP is fabricated in a 0.35 μm 4 level metal CMOS process, and packaged in a 304 pin QFP. A summary of the area consumed by the different processing blocks is provided in Table 3.1.

In order to reduce memory costs, Talisman consolidates the different large memory buffers in the system into a single external memory subsystem. This subsystem, which uses dual Rambus channels to provide a peak memory bandwidth in excess of 10 Gb/s, is integrated into the POP. It is thus conveniently located between the two other blocks in Talisman that require external memory: the Media Processor and the Image Layer Compositor.

The Image Layer Compositor The Image Layer Compositor (ILC) is responsible for fetching image layers from memory, decompressing them, bilinearly filtering them (if necessary), then outputting them in depth order to the compositor in order to generate a strip of the video output. It is implemented using a 0.35 μm four level metal 3.3V CMOS process, and packaged in a 304 pin QFP. The maximum filtering and compositing throughput is 320 Mpixels/sec.

In order to handle the latency involved in fetching and decompressing an image layer, two traverses of the display list are made. The first fetches objects from memory and decompresses it into a 64 Kbit image “cache” (really just a temporary storage buffer, since nothing ever gets re-used) The second traversal of the display list results in the bilinear interpolation of the “cached” decompressed image data and it’s writing to an alpha-blending compositor (located on another chip !) As shown in Table 3.1, the decompressor and staging RAM occupy the majority of the chip. The image filtering and composite control (including the second display list traversal) occupy only 8% of the silicon.

Compositing DAC The final stage of the ILC, the alpha-blending, is actually located on a separate die, probably due to area constraints. A video DAC is incorporated to lighten the package count. The ILC passes (in reverse depth order) four 32b (RGB plus Alpha) pixels at a time to the compositing DAC. The alpha-blended compositing is performed into a double buffered 32 scan-line buffer, using a single 8b 32 scan-line alpha buffer. Thus the compositing DAC contains some simple arithmetic logic and 1,792 bits of memory per scan line pixel. For the Escalante target resolution (1344 hor.), the compositor requires 2.5 Mbits of buffer memory.

3.2 Other Talisman Implementations

Other implementations of Talisman are expected. At the extreme low end, the entire architecture may be implemented as software running on a conventional microprocessor, taking advantage of virtual buffers to improve data cache performance. The specialized Image Layer Composition hardware may be replaced by a conventional frame buffer, into which the alpha-blended, composited strips of video data are stored.

While Video Signal Processors are explicitly included into the Escalante reference design (as the Media Processor), a complete implementation could consist solely of the VSP and memory. Actually, Talisman could be implemented on any system described in this survey. Even the Infinite Reality contains a method of feeding the output of renderer back to it's input, critical to a layered approach. Newer VSPs which include co-processors capable of implementing parts of the Talisman architecture are especially suitable. The Chromatics Mpact2, for example, has an Image Co-processor analogous to the Image Layer Compositor.

4 Commentary

4.1 VLIW

The noticeable industry trend toward very long instruction word (VLIW) processors is not surprising. Earlier programmable DSPs with a small number (two or three) of functional units were very conscious of code size, utilizing CISC instruction sets which supported commonly used parallel issues as single instructions (*e.g.* `multiply-add`.) As the number of functional units integrated into a processor is increased, the need for multiple instruction issue becomes critical.

A VLIW approach to multiple instruction issue is favored for two reasons. First, the overhead of an alternative "super-scalar" (run-time instruction parallelizin') approach is significant. And second, as Philips describes it: "Defining software compatibility at the source code level" [7] is not a problem for video signal processors, whose software life cycle more closely resembles an embedded controller than a mainstream microprocessor. In order to address the related expansion of program code, compression of the instruction stream such as that used on the Philips TriMedia [8] or the Infinite Reality GE [24] is promising. Roll-your-own CISC !

The advantages and disadvantages of group, or vector, instructions have already been mentioned above. The other noteworthy instruction set architecture feature commonly found on the VSPs is conditional execution of each operation (first seen in the IBM 604, 1952 [34].) This prevents disruption of the instruction fetch pipeline (long, even without instruction compression). It can be carried too far, however — the instructions stream overhead of supporting eight different guard registers (in the TMS320C6201) seems exhorbitant.

Units	Area ($G\lambda^2$)	Relative Area
Memory Interface	0.22	4%
Compression/Decompression	1.4	27%
Clip & Scan Convert	0.76	15%
Texture Map & Composite	1.3	26%
Display Generation	1.0	20%
Interblock Routing	0.47	8%
Total Usable Area	5.1	

Table 9: Escalante (POP+ILC) Relative Block Costs

4.2 Specialized Co-processors

The specialized co-processors integrated onto the VSPs were varied. The one common co-processor was an MPEG2 system bit-stream variable length decoder, found on the Philips TriMedia, the Chromatics Mpact, and the Samsung MSP. This reflects the fundamental difficulty of handling a datatype which traditional processors aren't prepared to process. Since a stream of bits, with variable length fields, is central to most efficient communications channels, a processing element or co-processor for parsing/manipulating it should become ubiquitous. Hopefully programmable architectures for manipulating bit-streams will become more common, supplanting the fixed protocol architectures presently encountered.

Other co-processors present on surveyed VSPs were:

- The Image Co-processor on the Philips TriMedia TM1000
- The polygon rendering pipeline on the Chromatics Mpact2
- The SIMD floating point vector processor on the Samsung MSP.

4.3 Memory Costs

The amount of memory and memory bandwidth required by a high performance video or graphics is a problem. The Rambus solution addresses the bandwidth quite well throught the use of advanced signalling techniques, but doesn't reduce the amount of memory required. Compressing all data before communications or storage, as proposed by Talisman, reduces both the size and bandwidth requirements. The tradeoff is the amount of processing power/area required (27% of the total in the case of Escalante.) As the relative cost of processing/memory decreases, the compression approach should become more common. And one can always attempt to reduce memory requirements through changing the overall algorithm.

4.4 The Retirement of the Frame Buffer

The disappearance of the frame buffer in the Talisman architecture is notable, but not new. Many earlier graphics architectures have used display lists to generate the images one or more scan lines at a time, either in order to support fast window or object (sprite) manipulations, or to eliminate the need for frame buffer memory. The viability of its replacement, the Image Layer Compositor (ILC) hinges on the ready availability of multi-billion λ^2 devices at consumer prices, and on two other architectural aspects of Talisman:

1. The pre-sorting required for dynamic display already mandated in order to support virtual buffers.
2. The compression mandated for all image layers in order to conserve memory reduces the memory bandwidth required for fetching the data for display — the typical killer for list-based display generators. Even with compression, Microsoft estimates a display read bandwidth of 1 Gb/s [3]

Nonetheless, decompressing the image data, then performing a bilinear filtering and alpha blending of the image layers in the process of generating the display exhibits a level of sophistication in the ILC that is relatively unique. The cost (in silicon area) is substantial: (the percentage shown for Display Generation in Table 4.3 doesn't include the 2.4 Mbits of compositing buffer required. All told, the area cost is roughly comparable to the 3.3 $G\lambda^2$ required for an equivalent frame buffer but the performance is vastly superior.

While there is no explicit frame buffer in the Talisman system, the memory containing the image layer data (output from the polygon rendering stage) does decouple the display generation from the image rendering. This point isn't developed upon in the architecture, as they have accepted a relatively low target video output resolution (1344x1024 at 75fps), attainable with a single chip solution. Decoupling allows the display resolution to be scaled spatially [4] by simply scaling the ILC and not the entire system. Since the image layer composition is easily parallelizable (the data already having been partitioned into virtual buffers) multiple ILCs could be employed. A redesign to distribute the memory among all the system chips, instead of concentrating it all on the rendering engine, as in Escalante, would be necessary. I expect to see more systems taking this approach, as the cost of processing silicon (relative to the cost of memory bandwidth and silicon) drops.

References

[1] Alex Peleg, Sam Wilkie, and Uri Weiser. Intel MMX for Multimedia PCs. *Communications of the ACM*, 40(1), January 1997.

[2] R. Lee. Subword Parallelism with MAX-2. *IEEE Micro*, 16(4):51–59, August 1996.

[3] M. Tremblay, J. M. O'Connor, V. Narayanan, and H. Liang. VIS speeds new media processing. *IEEE Micro*, 16(4):51–59, August 1996.

[4] Texas Instruments, Inc., Houston, TX. *TMX320C6201 Data Sheet*, Feb. 1997. Literature Ref. SPRU051A, Also at http://www.ti.com/sc/docs/dsps/products/c6x/index.htm.

[5] Texas Instruments, Inc., Houston, TX. *TMS320C62xx Technical Brief*, Jan. 1997. Literature Ref. SPRU197, Also at http://www.ti.com/sc/docs/dsps/products/c6x/index.htm.

[6] Texas Instruments, Inc., Houston, TX. *TMS320C62xx CPU and Instruction Set Reference Guide*, Jan. 1997. Literature Ref. SPRU189A, Also at http://www.ti.com/sc/docs/dsps/products/c6x/index.htm.

[3] Assuming a display size of 1024x768, 75 Hz refresh rate, an ave. compression factor of 5, and an ave. num. layers of 1.7. Simply reading an uncompressed pre-composited image out of a frame buffer would require 1.35 Gb/s.

[4] The disparity between the temporal display refresh rate, which is a characteristic of the display technology (typically 60fps or greater for raster scanned CRT displays), and the display content update rate (which need not be greater than 30 fps) is not utilized by Talisman. Although this disparity argues in favor of retaining the frame buffer, it is typically small.

[7] Philips Electronics TriMedia Group, Sunnyvale, CA. *TM1000 Data Sheet*, 1997. Also at http://www.trimedia.philips.com/prod.html.

[8] Selliah Rathnam and Gerrit A. Slavenburg. An Architectural Overview of the Programmable Multimedia Processor, TM-1. In *Proceedings of COMPCON96*, pages 319–326. IEEE, Feb. 1996.

[9] Gerrit A. Slavenburg, Selliah Rathnam, and Henk Dijkstra. The Trimedia TM-1 PCI VLIW Mediaprocessor. In *Proc. of Hot Chips 8: A Symposium on High Performance Chips*, Stanford, CA, Aug. 1996. Slides at http://infopad.eecs.berkeley.edu/HotChips8/.

[10] Craig Hansen. Architecture of a Broadband Mediaprocessor. In *Proceedings of COMPCON96*. IEEE, Feb. 1996. Also at http://www.microunity.com/white.htm.

[11] Craig Hansen. MicroUnity's MediaProcessor Architecture. *IEEE Micro*, pages 34–41, August 1996.

[12] MicroUnity Frequently Asked Questions. At http://www.microunity.com/faqpg.htm.

[13] Chromatics Research, Inc., Sunnyvale, CA. *Mpact Media Processor Preliminary Data Sheet*, Jan. 1997. Available at http://www.mpact.com/.

[14] P. Foley. The Mpact Media Processor. In *Proceedings of COMPCON96*, pages 311–319. IEEE, Feb. 1996.

[15] Chromatic Reveals Key Elements of Second-Generation Mpact Media Processor Architecture. Chromatics Research, Inc., Press Release, Oct. 1996. Available at http://www.mpact.com/press/.

[16] Steve Purcell. MPact2 Architecture. Invited Talk at IS&T/SPIE 3021-05: Multimedia Hardware Architectures 1997, San Jose, CA, Feb. 12 1997.

[17] Samsung Target Industry Standard with MSP: A Second Generation Multi-Media Signal Processor. Samsung Semiconductor Press Release, Aug. 1996. Available at http://www.sec.samsung.com/News/1996news.html.

[18] L. T. Nguyen, M. Mohamed, H. Park, Y. Pai, R. Wong, A. Qureshi, P. Psong, F. Valesco, H. D. Truong, and C. Reader. Multimedia Signal Processor (MSP) Summary. In *Proc. of Hot Chips 8: A Symposium on High Performance Chips*, Stanford, CA, Aug. 1996. Also at http://infopad.eecs.berkeley.edu/HotChips8/.

[19] Edgar Holmann, Toyohiko Yoshida, Akira Yamada, and Yukihiko Shimazu. VLIW Processor for Multimedia Applications. In *Proc. of Hot Chips 8: A Symposium on High Performance Chips*, Stanford, CA, Aug. 1996. Slides at http://infopad.eecs.berkeley.edu/HotChips8/.

[20] Shunsuke Kamijo. The MMA: A Long-Instruction-Word Multimedia Processor. Presented at Microprocessor Forum Conf. '96, Oct. 1996.

[21] E. Iwata, K. Seno, M. Aikawa, M. Ohki, H. Yoshikawa, Y. Fukuzawa, H. Hanaki, K. Nishibori, Y. Kondo, H. Takamuki, T. Nagai, K. Hasegawa, H. Okuda, I. Kumata, M. Soneda, S. Iwase, and T. Yamazaki. A 2.2GOPS Video DSP with 2-RISC MIMD, 6-PE SIMD Architecture for Real-Time MPEG2 Video Coding/Decoding. In *Proceedings of the Int'l. Solid State Circuits Conf. (ISSC97)*, San Francisco, CA, February 1997.

[22] Texas Instruments, Inc., Houston, TX. *TMS320C8x System Level Synopsis*, 1995. Literature Ref. SPRU113B, Also at http://www.ti.com/sc/docs/dsps/products/c8x/index.htm.

[23] John S. Montrym, Daniel R. Baum, and David L. Dignam. InfiniteReality: A Real-Time Graphics System. In *Proc. SIGGRAPH '97*, ACM/SIGGRAPH Computer Graphics Annual Conference Series, Los Angeles, CA, Aug. 1997. ACM.

[24] Brian McClendon and John Montrym. InfiniteReality Graphics: Power Through Complexity. In *Proc. of Hot Chips 8: A Symposium on High Performance Chips*, Stanford, CA, Aug. 1996. Slides at http://infopad.eecs.berkeley.edu/HotChips8/.

[25] Kurt Akeley. RealityEngine Graphics. In *Proc. SIGGRAPH '93*, ACM/SIGGRAPH Computer Graphics Annual Conference Series. ACM, July 1993.

[26] Onyx2: Technology overview. Silicon Graphics, Inc., Available at http://www.sgi.com/Products/hardware/graphics/technology/.

[27] Neil Trevett. PerMedia and GLINT Delta: New Generation Silicon for 3D Graphics. In *Proc. of Hot Chips 8: A Symposium on High Performance Chips*, Stanford, CA, Aug. 1996. Slides at http://infopad.eecs.berkeley.edu/HotChips8/.

[28] Texas Instruments, Inc., Houston, TX. *TVP4010 Technical Characteristics*, 1997. Available at http://www.ti.com/sc/docs/msp/multimed/tech.htm.

[29] Jay Torborg and James T. Kajiya. Talisman: Commodity Realtime 3D Graphics for the PC. In *Proc. SIGGRAPH '96*, ACM/SIGGRAPH Computer Graphics Annual Conference Series, New Orleans, LA, August 1996. ACM. Also at http://www.research.microsoft.com/siggraph96/talisman.htm.

[30] Martin Randall. Talisman: Multimedia for the PC. *IEEE Micro*, pages 11–19, March-April 1997.

[31] N. Gharachorloo, S. Gupta, R. F. Sproull, and I. E. Sutherland. A Characterization of Ten Rasterization Techniques. In *Proc. SIGGRAPH '89*, ACM/SIGGRAPH Computer Graphics Annual Conference Series. ACM, August 1989.

[32] V. Michael Bove, Jr., Brett D. Granger, and John A. Watlington. Real-Time Decoding and Display of Structured Video. In *Proc. IEEE Int'l. Conf. on Multimedia Computing and Systems '94*, Boston, MA, May 1994.

[33] Emmett Kilgariff and Martin Randall. Touchstone: A Fresh Approach to Multimedia for the PC. In *Proc. of Hot Chips 8: A Symposium on High Performance Chips*, Stanford, CA, Aug. 1996. Also at http://www.sei.com/touchstn/touchstn.htm.

[34] Gerrit A. Blaauw and Frederick P. Brooks. *Computer Architecture: Concepts and Evolution*, page 361. Addison-Wesley, Reading, MA, 1997.