5 Parallel Architectures for Programmable Video Signal Processing

Zhao Wu and Wayne Wolf

Princeton University, Princeton, New Jersey

1 INTRODUCTION

Modern digital video applications, ranging from video compression to content analysis, require both high computation rates and the ability to run a variety of complex algorithms. As a result, many groups have developed programmable architectures tuned for video applications. There have been four solutions to this problem so far: modifications of existing microprocessor architectures, application-specific architectures, fully programmable video signal processors (VSPs), and hybrid systems with reconfigurable hardware. Each approach has both advantages and disadvantages. They target the market from different perspectives. Instruction set extensions are motivated by the desire to speed up video signal processing (and other multimedia applications) by software solely rather than by special-purpose hardware. Application-specific architectures are designed to implement one or a few applications (e.g., MPEG-2 decoding). Programmable VSPs are architectures designed from the ground up for multiple video applications and may not perform well on traditional computer applications. Finally, reconfigurable systems intend to achieve high performance while maintaining flexibility.

Generally speaking, video signal processing covers a wide range of applications from simple digital filtering through complex algorithms such as object recognition. In this survey, we focus on advanced digital architectures, which are intended for higher-end video applications. Although we cannot address every possible video-related design, we cover major examples of video architectures that illustrate the major axes of the design space. We try to enumerate all the cutting-edge companies and their products, but some companies did not provide much detail (e.g., chip architecture, performance, etc.) about their products, so we do not have complete knowledge about some Integrated circuits (ICs) and systems. Originally, we intended to study only the IC chips for video signal processing, but reconfigurable systems also emerge as a unique solution, so we think it is worth mentioning these systems as well.

The next section introduces some basic concepts in video processing algorithms, followed by an early history of VSPs in Section 3. This is just to serve as a brief introduction of the rapidly evolving industry. Beginning in Section 6, we discuss instruction set extensions of modern microprocessors. In Section 5, we compare the existing architectures of some dedicated video codecs. Then, in Section 6, we contrast in detail and analyze the pros and cons of several programmable VSPs. In Section 7, we introduce systems based on reconfigurable computing, which is another interesting approach for video signal processing. Finally, conclusions are drawn in Section 8.

2 BACKGROUND

Although we cannot provide a comprehensive introduction to video processing algorithms here, we can introduce a few terms and concepts to motivate the architectural features found in video processing chips. Video compression was an early motivating application for video processing; today, there is increased interest in video analysis.

The Motion Pictures Experts Group (MPEG) (www.cselt.it) has been continuously developing standards for video compression. MPEG-1, -2, and -4 are complete, and at this writing, work on MPEG-7 is underway. We refer the reader to the MPEG website for details on MPEG-1 and -2 and to a special issue of *IEEE Transactions on Circuits and Systems for Video Technology* for a special issue on MPEG-4. The MPEG standards apply several different techniques for video compression. One technique, which was also used for image compression in the JPEG standard (JPEG book) is coding using the discrete cosine transform (DCT). The DCT is a frequency transform which is used to transform an array of pixels (an 8×8 array in MPEG and JPEG) into a spatial frequency spectrum; the two-dimensional DCT for the 2D array can be found by computing two 1D DCTs on the blocks. Specialized algorithms have been developed for computing the DCT efficiently. Once the DCT is computed, lossy compression algorithms will throw away coefficients which represent high-spatial frequencies, because those represent fine details which are harder to resolve by the human eye, particularly in moving objects. DCT is one of the two most computation-intensive operations in MPEG.

The other expensive operation in MPEG-style compression is block motion estimation. Motion estimation is used to encode one frame in terms of another (DCT is used to compress data within a single frame). As shown in Figure 1, in MPEG-1 and -2, a macroblock (a 16×16 array of pixels composed of four blocks) taken from one frame is correlated within a distance p of the macroblock's current position (giving a total search window of size $2p + 1 \times 2p + 1$). The reference macroblock is compared to the selected macroblock by two-dimensional correlation: Corresponding pixels are compared and the sum of the magnitudes of the differences is computed. If the selected macroblock can be matched within a given tolerance, in the other frame, then the macroblock need be sent only once for both frames. A region around the macroblock's original position is chosen as the search area in the other frame; several algorithms exist which avoid performing the correlation at every offset within the search region. The macroblock is given a motion vector that describes its position in the new frame relative to its original position. Because matches are not, in general, exact, a difference pattern is sent to describe the corrections made after applying the macroblock in the new context.

MPEG-1 and -2 provide three major types of frames. The I-frame is coded without motion estimation. DCT is used to compress blocks, but a lossily compressed version of the entire frame is encoded in the MPEG bit stream. A P-



Figure 1 Block motion estimation.

frame is predicted using motion estimation. A P-frame is encoded relative to an earlier I-frame. If a sufficiently good macroblock can be found from the I-frame, then a motion vector is sent rather than the macroblock itself; if no match is found, the DCT-compressed macroblock is sent. A B-frame is bidirectionally encoded using motion estimation from frames both before and after the frame in time (frames are buffered in memory to allow bidirectional motion prediction). MPEG-4 introduces methods for describing and working with objects in the video stream. Other detailed information about the compression algorithm can be found in the MPEG standard [1].

Wavelet-based algorithms have been advocated as an alternative to blockbased motion estimation. Wavelet analysis uses filter banks to perform a hierarchical frequency decomposition of the entire image. As a result, wavelet-based programs have somewhat different characteristics than block-based algorithms.

Content analysis of video tries to extract useful information from video frames. The results of content analysis can be used either to search a video database or to provide summaries that can be viewed by humans. Applications include video libraries and surveillance. For example, algorithms may be used to extract key frames from videos. The May and June 1998 issues of the *Proceedings of the IEEE* and the March 1998 issue of *IEEE Signal Processing Magazine* survey multimedia computing and signal processing algorithms.

3 EARLY HISTORY OF VLSI VIDEO PROCESSING

An early programmable VSP was the Texas Instruments TMS34010 graphics system processor (GSP) [2]. This chip was released in 1986. It is a 32-bit microprocessor optimized for graphics display systems. It supports various pixel formats (1-, 2-, 4-, 8-, and 16-bit) and operations and can accelerate graphics interface efficiently. The processor operates at a clock speed from 40 to 60 MHz, achieving a peak performance of 7.6 million instructions per second (MIPS).

Philips Semiconductors developed early dedicated video chips for specialized video processors. Philips announced two digital multistandard color decoders at almost the same time. Both the SAA9051 [3] and the SAA7151 [4] integrate a luminance processor and chrominance processor on-chip and are able to separate 8-bit luminance and 8-bit chrominance from digitized S-Video or composite video sources as well as generate all the synchronization and control signals. Both VSPs support PAL, NTSC, and SECAM standards.

In the early days of JPEG development, its computational kernels could not be implemented in real time on typical CPUs, so dedicated DCT/IDCT (discrete cosine transform–inverse DCT) units, Huffman encoder/decoder, were built to form a multichip JPEG codec [another solution was multiple digital signal processors (DSPs)]. Soon, the multiple modules could be integrated onto a single chip. Then, people began to think about real-time MPEG. Although MPEG-1 decoders were only a little more complicated than JPEG decoders, MPEG-1 encoders were much more difficult. At the beginning, encoders that are fully compliant to MPEG-1 standards could not be built. Instead, people had to come up with some compromise solutions. First, motion-JPEG or I-frame-only (where the motion estimation part of the standard is completely dropped) encoders were designed. Later, forward prediction frames were added in IP-frame encoders. Finally, bidirectional prediction frames were implemented. The development also went through a whole procedure from multichip to singlechip. Meanwhile, the microprocessors became so powerful that some software MPEG-1 players could support real-time playback of small images. The story of MPEG-2 was very similar to MPEG-1 and began as soon as the first single-chip MPEG-1 decoder was born. Like MPEG-1, it also experienced asymptotic approaches from simplified standards to fully compliant versions, and from multichip solutions to single chip solutions.

The late 1980s and early 1990s saw the announcement of several complex, programmable VSPs. Important examples include chips from Matsushita [5], NTT [6], Philips [7], and NEC [8]. All of these processors were high-performance parallel processors architected from the ground up for real-time video signal processing. In some cases, these chips were designed as showcase chips to display the capabilities of submicron very-large-scale integration (VLSI) fabrication processes. As a result, their architectural features were, in some cases, chosen for their ability to demonstrate a high clock rate rather than their effectiveness for video processing. The Philips VSP-1 and NEC processor were probably the most heavily used of these chips.

The software (compression standards, algorithms, etc.) and hardware (instruction set extensions, dedicated codecs, programmable VSPs) developments of video signal processing are in parallel and rely heavily on each other. On one hand, no algorithms could be realized without hardware support; on the other hand, it is the software that makes a processor useful. Modern VLSI technology not only makes possible but also encourages the development of coding algorithms—had developers not been able to implement MPEG-1 in hardware, it may not have become popular enough to inspire the creation of MPEG-2.

4 INSTRUCTION SET EXTENSIONS FOR VIDEO SIGNAL PROCESSING

The idea of providing special instructions for graphics rendering in a generalpurpose processor is not new; it appeared as early as 1989 when Intel introduced i860, which has instructions for Z-buffer checks [9]. Motorola's 88110 is another example of using special parallel instructions to handle multiple pixel data simultaneously [10]. To accommodate the architectural inefficiency for multimedia applications, many modern general-purpose processors have extended their instruction set. This kind of patch is relatively inexpensive as compared to designing a VSP from the very beginning, but the performance gain is also limited. Almost all of the patches adopt single instruction multiple data (SIMD) model, which operates on several data units at a time. Apparently, the supporting facts behind this idea are as follows: First, there is a large amount of parallelism in video applications; second, video algorithms seldom require large data sizes. The best part of this approach is that few modifications need to be done on existing architectures. In fact, the area overhead is only 0.1% (HP PA-RISC MAX2) to 3% (Sun UltraSparc) of the original die in most processors. Already having a 64-bit datapath in the architecture, it takes only a few extra transistors to provide pixel-level parallelism on the wide datapath. Instead of working on one 64-bit word, the new instructions can operate on 8 bytes, four 16-bit words, or two 32bit words simultaneously (with the same execution time), octupling, quadrupling, or doubling the performance, respectively. Figure 2 shows the parallel operations on four pairs of 16-bit words.

In addition to the parallel arithmetic, shift, and logical instructions, the new instruction set must also include data transfer instructions that pack and unpack data units into and out of a 64-bit word. In addition, some processors (e.g., HP PA-RISC MAX2) provide special data alignment and rearrangement instructions to accelerate algorithms that have irregular data access patterns (e.g., zigzag scan in discrete cosine transform). Most instruction set extensions provide three ways to handle overflow. The default mode is modular, nonsaturating arithmetic, where any overflow is discarded. The other two modes apply saturating arithmetic. In signed saturation, an overflow causes the result to be clamped to its maximum or minimum signed value, depending on the direction of the overflow. Similarly, in unsigned saturation, an overflow sets the result to its maximum or minimum unsigned value.



Figure 2 Examples of SIMD operations.

Vendor	Microprocessor	Extension	Release date	Ref.
Hewlett-Packard	HP PA-RISC 1.0 and 2.0	MAX1	Jan. 1994	11
		MAX2	Feb. 1996	12
Intel	Pentium and Pentium Pro	MMX	March 1996	14
Sun	UltraSPARC-I, -II, and -III	VIS	Dec. 1994	15
DEC	Alpha 21264	MVI	Oct. 1996	17
MIPS	MIPS R10000	MDMX	March 1997	18

Table 1 Instruction Set Extensions for Multimedia Applications

An important issue for instruction set extension is compatibility. Multimedia extensions allow programmers to mix multimedia-enhanced code with existing applications. Table 1 shows that all the modern microprocessors have added multimedia instructions to their basic architecture. We will discuss the first three microprocessors in detail.

4.1 Hewlett-Packard MAX2 (Multimedia Acceleration eXtensions)

Hewlett-Packard was the first CPU vendor to introduce multimedia extensions for general-purpose processors in a product [11]. MAX1 and MAX2 were released in 1994 and 1996, respectively, for 32-bit PA-RISC and 64-bit PA-RISC processors.

Table 2 lists the MAX2 instructions in PA-RISC 2.0 [12]. Having observed a large portion of constant multiplies in multimedia processing, HP added *hshladd* and *hshradd* to speed up this kind of operation. The mix and permute instructions are useful for subword data formatting and rearrangement operations. For example, the mix instructions can be used to expand 16-bit subwords into 32-bit subwords and vice versa.

Another example is matrix transpose, where only eight mix instructions are required for a 4×4 matrix. The permute instruction takes 1 source register and produces all the 256 possible permutations of the 16-bit subwords in that register, with or without repetitions.

From Table 3 we can see that MAX2 not only reduces the execution time significantly but also requires fewer registers. This is because the data rearrangement instructions need fewer temporary registers and saturation arithmetic saves registers that hold the constant clamping value.

4.2 Intel MMX (Multi Media eXtensions)

Table 4 lists all the 57 MMX instructions, which, according to Intel's simulations of the P55C processor, can improve performance on most multimedia applica-

Group	Mnemonic	Description
Parallel add	Hadd	Add 4 pairs of 16-bit operands, with modulo arith- metic
	Hadd,ss	Add 4 pairs of 16-bit operands, with signed saturation
	Hadd,us	Add 4 pairs of 16-bit operands, with unsigned satura- tion
Parallel subtract	Hsub	Subtract 4 pairs of 16-bit operands, with modulo arith- metic
	Hsub,ss	Subtract 4 pairs of 16-bit operands, with signed satura- tion
	Hsub,us	Subtract 4 pairs of 16-bit operands, with unsigned sat- uration
Parallel shift and add	Hshladd	Multiply 4 first operands by 2, 4, or 8 and add corre- sponding second operands
	hshradd	Divide 4 first operands by 2, 4, or 8 and add corre- sponding second operands
Parallel average	havg	Arithmetic mean of 4 pairs of operands
Parallel shift	hshr	Shift right by 0 to 15 bits, with sign extension on the left
	hshr,u	Shift right by 0 to 15 bits, with zero extension on the right
	hshl	Shift left by 0 to 15 bits, with zeros shifted in on the right
Mix	mixh,L mixh,R mixw,L mixw,R	Interleave alternate 16-bit [h] or 32-bit [w] subwords from two source registers, starting from leftmost [L] subword or ending with rightmost [R] subword
Permute	permh	Rearrange subwords from one source register, with or without repetition

Table 2MAX2 Instructions in PA-RISC 2.0

Source: Ref. 13.

tions by 50–100%. Compared to HP's MAX2, the MMX multimedia instruction set is more flexible on the format of the operand. It not only works on four 16bit words but also supports 8 bytes and two 32-bit words. In addition, it provides packed multiply and packed compare instructions. Using packed multiply, it requires only 6 cycles to calculate four products of 16×16 multiplication on a P55C, whereas on a non-MMX Pentium, it takes 10 cycles for a single 16×16 multiplication. The behavior of pack and unpack instructions is very similar to that of the mix instructions in MAX2. Figure 3 illustrates the function of two MMX instructions. The DSP-like PMADDWD multiplies two pairs of 16-bit

Kernel algorithm	16×16 Block match	8×8 Matrix transpose	3×3 Box filter	8×8 IDCT
Cycles	160 (426)	16 (42)	548 (2324)	173 (716)
Registers Speedup	14 (12) 2.66	18 (22) 2.63	15 (18) 4.24	17 (20) 4.14

Table 3 Performance of Multimedia Kernels With (and Without)MAX2 Instructions

Source: Ref. 13.

words and then sums each pair to produce two 32-bit results. On a P55C, the execution takes three cycles when fully pipelined. Because multiply-add operations are critical in many video signal processing algorithms such as DCT, this feature can improve the performance of some video applications (e.g., JPEG and MPEG) greatly. The motivation behind the packed compare instructions is a common video technique known as chroma key, which is used to overlay an object on another image (e.g., weather person on weather map). In a digital implementation with MMX, this can be done easily by applying packed logical operations after packed compare. Up to eight pixels can be processed at a time.

Unlike MAX2, MMX instructions do not use general-purpose registers; all the operations are done in eight new registers (MM0–MM7). This explains why the four packed logical instructions are needed in the instruction set. The MMX registers are mapped to the floating-point registers (FP0–FP7) in order to avoid introducing a new state. Because of this, floating-point and MMX instructions cannot be executed at the same time. To prevent floating-point instructions from corrupting MMX data, loading any MMX register will trigger the busy bit of all the FP registers, causing any subsequent floating-point instructions to trap. Consequently, an EMMS instruction must be used at the end of any MMX routine to resume the status of all the FP registers. In spite of the awkwardness, MMX has been implemented in several Pentium models and also inherited in Pentium II and Pentium III.

4.3 Sun VIS

Sun UltraSparc is probably today's most powerful microprocessor in terms of video signal processing ability. It is the only off-the-shelf microprocessor that supports real-time MPEG-1 encoding and real-time MPEG-2 decoding [15]. The horsepower comes from a specially designed engine: VIS, which accelerates multimedia applications by twofold to sevenfold, executing up to 10 operations per cycle [16].

Group	Mnemonic	Description
Data transfer, pack and	MOV[D,Q] ^a	Move [double,quad] to/from MM register Pack words into bytes with unsigned satu-
unpack	PACKUSWB	ration
	PACKSS[WB,DW]	Pack [words into bytes, doubles into words] with signed saturation
	PUNPCKH[BW,WD,DQ]	words, doubles] from MM register Unpack (interleave) low-order [bytes,
	PUNPCKL[BW,WD,DQ]	words, doubles] from MM register
Arithmetic	PADD[B,W,D]	Packed add on [byte, word, double]
	PADDS[B,W]	Saturating add on [byte, word]
	PADDUS[B,W]	Unsigned saturating add on [byte, word]
	PSUB[B,W,D]	Packed subtract on [byte, word, double]
	PSUBS[B,W]	Saturating subtract on [byte, word]
	PSUBUS[B,W]	Unsigned saturating subtract on [byte, word]
	PMULHW	Multiply packed words to get high bits of product
	PMULLW	Multiply packed words to get low bits of product
	PMADDWD	Multiply packed words, add pairs of prod- ucts
Shift	PSLL[W,D,Q]	Packed shift left logical [word, double, quad]
	PSRL[W,D,Q]	Packed shift right logical [word, double, quad]
	PSRA[W,D]	Packed shift right arithmetic [word, double]
Logical	PAND	Bit-wise logical AND
-	PANDN	Bit-wise logical AND NOT
	POR	Bit-wise logical OR
	PXOR	Bit-wise logical XOR
Compare	PCMPEQ[B,W,D]	Packed compare "if equal" [byte, word, double]
	PCMPGT[B,W,D]	Packed compare "if greater than" [byte, word, double]
Misc	EMMS	Empty MMX state

 Table 4
 MMX Instructions

^a Intel's definitions of word, double word, and quad word are, respectively, 16-bit, 32-bit, and 64-bit.

Source: Ref. 14.



Figure 3 Operations of (a) packed multiply-add (PMADDWD) and (b) packed compare-if-equal (PCMPEQW). (From Ref. 14.)

For a number of reasons, the visual instruction set (VIS) instructions are implemented in the floating-point unit rather than integer unit. First, some VIS instructions (e.g., partitioned multiply and pack) take multiple cycles to execute, so it is better to send them to the floating-point unit (FPU) which handles multiple-cycle instructions like floating-point add and multiply. Second, video applications are register-hungry; hence, using FP registers can save integer registers for address calculation, loop counts, and so forth. Third, the UltraSparc pipeline only allows up to three integer instructions per cycle to be issued; therefore, using FPU again saves integer instruction slots for address generation, memory load/ store, and loop control. The drawback of this is that the logical unit has to be duplicated in the floating-point unit, because VIS data are kept in the FP registers.

The VIS instructions (listed in Table 5) support the following data types: pixel format for true-color graphics and images, fixed16 format for 8-bit data, and fixed32 format for 8-, 12-, or 16-bit data. The partitioned add, subtract, and multiply instructions in VIS function very similar to those in MAX2 and MMX. In each cycle, the UltraSparc can carry out four 16×8 or two 16×16 multiplications. Moreover, the instruction set has quite a few highly specialized instructions. For example, *EDGE* instructions compare the address of the edge with that of the current pixel block, and then generate a mask, which later can be used by partial store (PST) to store any appropriate bytes back into the memory without using a sequence of read-modify-write operations. The ARRAY instructions are specially designed for three-dimensional (3D) visualization. When the 3D dataset is stored linearly, a 2D slice with arbitrary orientation could yield very poor locality in cache. The ARRAY instructions convert the 3D fixed-point addresses into a blocked-byte address, making it possible to move along any line or plane with good spatial locality. The same operation would require 24 RISC-equivalent instructions. Another outstanding instruction is PDIST, which calculates the SAD (sum of absolute difference) of two sets of eight pixels in parallel. This is the

Opcode ^a	Operands	Description
FPADD16/32(S)	Fsrc1, fsrc2, fdest	Four 16-bit or two 32-bit partitioned add or
FPSUB16/32(S)	Fsrc1, fsrc2, fdest	subtract
FPACK16	Fsrc2, fdest	Pack four 16-bit pixels into fdest
FPACK32	Fsrc1, fsrc2, fdest	Add two 32-bit pixels into fdest
FPACKFIX	Fsrc2, fdest	Pack two 32-bit pixels into fdest
FEXPAND	Fsrc2, fdest	Expand four 8-bit pixels into fdest
FPMERGE	Fsrc1, fsrc2, fdest	Merge two sets of four 8-bit pixels
FMUL8 \times 16(opt)	Fsrc1, fsrc2, fdest	Multiply four 8-bit pixels by four 16-bit constants
ALIGNADDR(L)	src1, src2, dest	Set up for unaligned access
FALIGNDATA	fsrc1, fsrc2, fdest	Align data from unaligned access
FZERO(S)	Fdest	Fill fdest with zeroes
FONE(S)	Fdest	Fill fdest with ones
FSRC(S)	fsrc, fdest	Copy fsrc to fdest
FNOT(S)	fsrc, fdest	Negate fsrc in fdest
Flogical(S)	fsrc1, fsrc2, fdest	Perform one of 10 logical operations (AND, OR, etc.)
FCMPcc16/32	fsrc1, fsrc2, dest	Perform four 16-bit or two 32-bit compares with results in dest
EDGE8/16/32(L)	src1, src2, dest	Edge boundary processing
PDIST	fsrc1, fsrc2, dest	Pixel distance calculation
ARRAY8/16/32	src1, src2, dest	Convert 3D address to blocked byte address
PST	fsrc, [address]	Partial store
FLD, STF	[address], fdest	8- or 16-Bit load/store to FP register
QLDA	[address], dest	128-Bit atomic load
BLD, BST	[address], dest	64-Bit block load/store

 Table 5
 Summary of VIS Instructions

^a S = single-precision option; L = little-endian option. *Source*: Ref. 15.

most time-consuming part in MPEG-1 and MPEG-2 encoders, which normally needs more than 1500 conventional instructions for a 16×16 block search; however, the same job can be done with only 32 *PDIST* instructions on Ultra-Sparc. Needless to say, VIS has vastly enhanced the capability and role of UltraSparc in high-end graphics and video systems.

4.4 Commentary

Instruction set extensions increase the processing power of general-purpose processors by adding new functional units dedicated for video processing and/or modifying existing architecture. All of the extensions take advantage of subword parallelism. The new instruction set not only accelerates video applications greatly but also can benefit other applications that bear the same kind of subword parallelism. The extended instruction sets get the processors more involved in video signal processing and lengthens the lifetime of those general-purpose processors.

5 APPLICATION-SPECIFIC PROCESSORS

Although some of today's modern microprocessors are powerful enough to support computation intensive video applications such as MPEG-1 and MPEG-2, it is still worthwhile to design dedicated VSPs that are tailored for a specific applications. Many dedicated VSPs are available now (see Table 6). They display a variety of architectures, including the array processor [19], pipelined architecture [20], and the application-specific processor (ASIC) [21]. Application-specific processors are often used in cost-sensitive applications, such as digital cable boxes and DVD players. Because these processors are highly optimized for limited functionality, they usually achieve better performance/cost ratio for application-specific systems than multimedia-enhanced microprocessors or programmable VSPs; hence, they will continue to exist in some cost-sensitive environments.

Most dedicated VSPs have been designed for MPEG-1 and MPEG-2 encoding and decoding. By adopting special-purpose components (e.g., DCT/IDCT unit, motion estimation unit, run-length encoder/decoder, Huffman encoder/ decoder, etc.) in a heterogeneous solution, dedicated VSPs can achieve very high performance at a relatively inexpensive cost.

5.1 8 \times 8 VCP and LVP

The 8×8 (" 8×8 " is a product name) 3104 video codec processor (VCP) and 3404 low bit-rate video Processor (LVP) have the same architecture, which is shown in Figure 4. They can be used to build videophones capable of executing all the components of the ITU H.324 specification. Both chips are members from 8×8 's multimedia processor architecture (MPA) family. The RISC IIT is a 32-bit pipelined microprocessor running at 33 MHz. Instead of using an instruction cache, it has a 32-bit interface to external SRAM for fast access. The RISC processor also supervises the two direct memory access (DMA) controllers, which provide 32-bit multichannel data passage for the entire chip. The embedded vision processor (VPe) carries out all the compression and decompression operations as well as preprocessing and postprocessing functions required by various applications. The chips can also be programmed for other applications, such as I-frame encoding, video decoding, and audio encoding/decoding for MPEG-1. The

Vendor	Product(s)	Application(s)	Architecture	Peak perform.	Technology
8 × 8	3104 (VCP) 3404 (LVP)	H.324 videophone MPEG-1 I-frame en- coder MPEG-1 decoder	Multimedia processor architecture (MPA), DSP-like engine	33 MHz	240-PQFP, 225-BGA, 5 V, 2 W
Analog Devices	ADV 601 ADV 601LC	4:1 to 350:1 real-time wavelet compression	Wavelet kernel, adaptive quantizer, and coder	27–29.5 MHz	120-PQFP, 5 V, low cost
	ADV 611] ADV 612]	Real-time compression/ decompression of CCIR-601 video at up to 7500:1	Wavelet kernel plus pre- cise compressed bit rate control	27 MHz	120-LQFP
C-Cube	$DV^{x} 5110$ $DV^{x} 6210$	MPEG-2 main profile at main level encoder	DV ^X multimedia architec- ture	100 MHz >10 BOPS	352-BGA, 3.3 V
	CLM 4440 CLM 4725 CLM 4740	MPEG2 authoring en- coder MPEG-2 storage encoder MPEG2 broadcast en- coder	CL4040 Video RISC Pro- cessor 3 (VRP-3) loaded with different microcode	60 MHz 5.7 BOPS	240-MQUAD, 3 W
	AViA 500] AViA 502]	MPEG-2 audio/video decoder	Video RISC processor- based architecture		160-PQFP, 3.3 V, 1.6 W
ESS Technology	ES3308	MPEG-2 audio/video decoder	RISC processor and MPEG processor	80 MHz	208-PQFP, 3.3 V, <1 W
IBM	MPEGME31 MPEGME30 (chipset)	MPEG-2 main profile at main level encoder	RISC-based architecture loaded with different microcode	54 MHz	304-CQFP, 0.5 μm, 3.3 V, 3.0–4.8 W
	MPEGCS22 MPEGCD21	MPEG-2 audio/video decoder	RISC-based architecture loaded with different microcode		160-PQFP, 0.4/0.5 μm, 3.3 V, 1.4 W

Table 6 Summary of Some Dedicated VSPs

InnovaCom	DV Impact	MPEG-2 main profile at main level encoder		54 MHz	304-BGA, 4.5 W
LSI Logic	VISC (chipset)	MPEG-2 main profile at main level encoder	MIPS-compatible RISC core	54 MHz	208-QFP, 0.5 µm, 3.3 V
	L64002 L64005	MPEG-2 audio/video de- coder	Customized RISC engine	27 MHz	160-PQFP, 3.3 V
	L64020	DVD decoder			
Matsushita	VDSP2]	MPEG-2 main profile at	SIMD DSP-core and	100 MHz	261-PGA
	COMET ∫	main level encoder	motion-estimation processor	80 MHz	144-PGA
Mitsubishi	DISP II (chipset)	MPEG-2 main profile at main level encoder	RISC-processor-based architecture		393-PGA, 257-PGA, 152-QFP
NTT	ENC-C	MPEG-2 simple profile		81 MHz	208-CQFP
	ENC-M	at main level encoder			304-CQFP
Philips	SAA6750H	Single-chip low-cost MPEG-2 encoder	Motion estimator plus preprocessing	27 MHz	0.5 µm, 198 mm ²
	SAA7201	MPEG-2 audio/video/ graphics decoder	Video decoder, audio decoder, and graphics unit	27 MHz	160-PQFP, 3.3 V
	SAA4991	Motion-compensated field-rate conversion	Top-level processor and coprocessors for inter- polation, motion esti- mation and vector	33 MHz 10 BOPS	0.8 μm, 1 million transis- tor, 84-PLCC, 5 V, 1.8 W
Sony	CXD1922Q	MPEG-2 main profile at main level encoder	DSP controller and coprocessors	27 MHz (DSP)	208-PQFP, 0.4 μm, 4.5 million transistor
	CXD1930Q	MPEG-2 audio/video de- coder	RISC, audio DSP and video processor	27 MHz	208-PQFP, 0.4 μm, 3.3 V
Vision Tech	MVision 10	MPEG-2 main profile at main level encoder	MIMD massively paral- lel scalable processor	40.5 MHz	304-CQFP, 0.5 μm, 5.2 million transistor



Figure 4 Architecture of 8×8 VCP and LVP. (From Ref. 22.)

microprogram is stored in the 2K \times 32 on-chip ROM; the 2K \times 32 SRAM provides alternatives to download new code.

The RISC processor can be programmed using an enhanced optimizing C compiler, but further information about the software developing tools is not available. Targeting at low bit-rate video applications, both VCP and LVP are lowend VSPs which do not support real-time applications such as MPEG-1 encoding.

5.2 Analog Devices ADV601 and ADV601LC

Unlike other VSPs which target DCT, the ADV601 and ADV601LC [23] target wavelet-based schemes, which have been advocated for having advantages over classical DCT compression. Wavelet-basis functions are considered to have a better correlation to the broad-band nature of images than the sinusoidal waves used in DCT approaches. One specific advantage of wavelet-based compression is that its entire image filtering eliminates the block artifacts seen in DCT-based schemes. This not only offers more graceful image degradation at high compression ratios but also preserves high image quality in spatial scaling, even up to a zoom factor of 16. Furthermore, because the subband data of the entire image is available, a number of image processing functions such as scaling can be done with little computational overhead. Because of these reasons, both JPEG 2000 and the upcoming MPEG-4 incorporate wavelet schemes in their definition.

Both the ADV601 and ADV601LC are low-cost (the 120-pin TQFP ADV601LC is, at this writing, \$14.95 each, in quantities of 10,000 units) realtime video codecs that are capable of supporting all common video formats, in-



Figure 5 Block diagram of Analog Devices ADV601 (ADV601LC). (From Ref. 23.)

cluding CCIR-656. It has precise compressed bit-rate control, with a wide range of compression ratios from visually lossless (4:1) to 350:1. The glueless video and host interfaces greatly reduce system cost while yielding high-quality images. As shown in Figure 5, the ADV601 consists of four interface blocks and five processing blocks. The wavelet kernel contains a set of filters and decimators that process the image in both horizontal and vertical directions. It performs forward and backward biorthogonal 2D separable wavelet transforms on the image. The transform buffer provides delay line storage, which significantly reduces bandwidth when calculating wavelet transforms on horizontally scanned images. Under the control of an external host or digital signal processor (DSP), the adaptive quantizer generates quantized wavelet coefficients at a near-constant bit-rate regardless of scene changes.

5.3 C-Cube DV^x and Other MPEG-2 Codecs

The C-Cube DV^x 5110 and DV^x 6210 [24] were designed to provide singlechip solutions to MPEG-2 video encoding at both main- and high-level MPEG-2 profiles (see Table 7) at up to 50 Mbit/sec. Main profile at mail level (MP@ML) is one of the MPEG-2 specifications used in digital satellite broadcasting and digital video disks (DVD). SP@ML is a simplified specification, which uses only I-frames and P-frames in order to reduce the complexity of compression algorithms.

The DV^x architecture (Fig. 6), which is an extension of the C-Cube Video RISC Processor (VRP) architecture, extends the VRP instruction set for efficient MPEG compression/decompression and special video effects. The chip includes two programmable coprocessors. A motion estimation coprocessor can perform hierarchical motion estimation on designated frames with a horizontal search

		Level			
Profile	Parameter	Low (CIF)	Main (CCIR 601)	High 1440 (HDTV 4:3)	High (HDTV 16:9)
Simple (I- and P-frames only, 4.2.0)	Image size Frame rate Bit rate		720 × 576 (480) 25 (30) Hz 15 Mbit/sec		
Main (4:2:0)	Image size Frame rate Bit rate	352 × 288 (240) 25 (30) Hz 4 Mbit/sec	720×576 (480) 25 (30) Hz 15 Mbit/sec	1440×1152 (960) 50 (60) Hz 60 Mbit/sec	1920 × 1152 (960) 50 (60) Hz 80 Mbit/sec
SNR scalable (4:2:0)	Image size Frame rate Bit rate	352 × 288 (240) 25 (30) Hz 3 Mbit/sec	720×576 (480) 25 (30) Hz 10 Mbit/sec		
	Image size Frame rate Bit rate	352 × 288 (240) 25 (30) Hz 4 Mbit/sec	720 × 576 (480) 25 (30) Hz 15 Mbit/sec		

Table 7Profiles and Levels for MPEG-2 Bit Stream

Spatially scalable	Image size	720×576 (480)				
(4:2:0)	Frame rate		25 (30) Hz			
	Bit rate		15 Mbit/sec			
	Image size		$1440 \times 1152 \ (960)$			
	Frame rate		50 (60) Hz			
	Bit rate		40 Mbit/sec			
	Image size		$1440 \times 1152 \ (960)$			
	Frame rate	50 (60) Hz				
	Bit rate		60 Mbit/sec			
High (4:2:2,	Image size	352×288 (240)	720×576 (480)	960×576 (480)		
4:2:0)	Frame rate	25 (30) Hz	25 (30) Hz	25 (30) Hz		
	Bit rate	4 Mbit/sec	20 Mbit/sec	25 Mbit/sec		
	Image size	720×576 (480)	$1440 \times 1152 \ (960)$	$1920 \times 1152 \ (960)$		
	Frame rate	25 (30) Hz	50 (60) Hz	50 (60) Hz		
	Bit rate	15 Mbit/sec	60 Mbit/sec	80 Mbit/sec		
	Image size	720×576 (480)	$1440 \times 1152 \ (960)$	$1920 \times 1152 \ (960)$		
	Frame rate	25 (30) Hz	50 (60) Hz	50 (60) Hz		
	Bit rate	20 Mbit/sec	80 Mbit/sec	100 Mbit/sec		

Note: No shading = base layer; light shading = enhancement layer 1; dark shading = enhancement layer 2. *Source*: Ref. 25.



Figure 6 C-Cube DV^X platform architecture block diagram. (From Ref. 24.)

range of ± 202 pixels and vertical range of ± 124 pixels. A DSP coprocessor can execute up to 1.6 billion arithmetic pixel-level operations per second. The IPC interface coordinates multiple DV^x chips (at the speed of 80 Mbyte/sec) to support higher quality and resolution. The video interface is a programmable high-speed input/output (I/O) port which transfers video streams into and out of the processor. MPEG audio is implemented in a separate processor.

Both the AViA500 and AVia502 support the full MPEG-2 video main profile at the main level and two channels of layer-I and layer-II MPEG-2 audio, with all the synchronization done automatically on-chip. Their architectures are shown in Figure 7. In addition, the AViA502 supports Dolby Digital AC-3 sur-



Figure 7 Architecture of AviA500 and Avia502. (From Ref. 24.)

round-sound decoding. The two MPEG-2 audio/video decoders each require 16 Mbit external DRAM.

These processors are sold under a business model which is becoming increasingly common in the multimedia hardware industry but may be unfamiliar to workstation users. C-Cube develops code for common applications for its processors and licenses the code chip customers. However, C-Cube does not provide tools for customers to write their own programs.

5.4 ESS Technology ES3308

As we can see from Figure 8, the ES3308 MPEG-2 audio, video, and transportlayer decoder [26] from ESS Technology has a very similar architecture to $8 \times$ 8's VCP or LVP. Both chips have a 32-bit pipelined RISC processor, a microcode programmable low-level video signal processor, a DRAM DMA controller, a Huffman decoder, a small amount of on-chip memory, and interfaces to various devices. The RISC processor of ES3308 is an enhanced version of MIPS-X prototype, which can be programmed using optimizing C compilers. In an embedded system, the RISC processor can be used to provide all the system controls and user features such as volume control, contrast adjustment, and so forth.

5.5 IBM MPEG-2 Encoder Chipset

The IBM chipset for MPEG-2 encoding [27] consists of three chips: an I-frame (MPEGSE10 in chipset MPEGME30, MPEGSE11 in chipset MPEGME31) chip, a Refine (MPEGSE20/21) chip, and a Search (MPEGSE30/31) chip. These chips can operated in one-, two-, or three-chip configurations, supporting a wide range



Figure 8 ES3308 block diagram.

of applications economically. In a one-chip configuration, a single "I" chip produces I-frame-only-encoded pictures. In a two-chip configuration, the "I" and "R" chips work together to produce IP-encoded pictures. Finally, in a threechip configuration, B-frames are generated for IPB-encoded pictures. The chipset offers expandable solutions for different needs. For example, I-frame-only bit streams are good enough for video editing, IP-encoded bit streams can reduce coding delay in video conferencing, and IPB-encoded bit streams offer a good compression ratio for applications like DVD. Furthermore, the chipset is also able to generate a 4:2:2 MPEG-2 profile at the main level. The encoder chipset has an internal RISC processor powered by a different microcode. IBM is releasing the microcode for the variable bit-rate (VBR) encoder. Little information is available from IBM about the architecture of the internal RISC processor and they do not offer tools for microcode-level development.

5.6 Philips SAA6750H, SAA7201, and SAA4991

The Philips SAA6750H [28] is a single-chip, low-cost MPEG-2 encoder which requires only 2 Mbytes of external DRAM. The chip includes a special-purpose motion estimation unit. It is able to generate bit streams that contain I-frames and P-frames. The designers claimed that "the disadvantage of omitting the B-frames can almost completely be eliminated using sophisticated on-chip pre-processing" and "at 10 Mbit/s, the CCIR picture quality is comparable with DV coding, while at 2.5 Mbit/s the SIF picture quality is comparable with Video CD" [28].

The SAA7201 [29] is an integrated MPEG-2 audio and video decoder. In addition, it incorporates a graphics decoder on-chip, which enhances region-based graphics and facilitates on-screen display. Using an optimized architecture, the AVG (audio, video, and graphics) decoder only requires $1M \times 16$ SDRAM, yet more than 1.2 Mbits (2.0 Mbits for a 60-Hz system) is available for graphics. The internal video decoder can handle all the MPEG-compliant streams up to the main profile at the main level, and the layer-1 and layer-2 MPEG audio decoder supports mono, stereo, surround sound, and dual-channel modes. The on-chip graphics unit and display unit allow multiple graphics boxes with background loading, fast switching, scrolling, and fading. Featuring a fast CPU access, the full bit-map can be updated within a display field period.

The Philips SAA4991 WP (MELZONIC) [30] is a motion-compensation chip, designed using Phideo, a special architecture synthesis tool for video applications developed by Philips Research [31]. This chip can automatically identify the original frame transition and correctly interpolate the motion up to a field rate of 100 Hz. In addition, it also performs noise reduction, vertical zoom functions, and 4:3 to 16:9 conversion. Four different types of SRAM and DRAM

totaling 160 Kbits are embedded on-chip in order to deliver an overall memory bandwidth of 25 Gbit/sec.

5.7 Sony Semiconductor CXD1922Q and CXD1930Q

The CXD1922Q [32] is a low-cost MPEG-2 video encoder for a real-time main profile at the main level. The on-chip encoding controller supports variable bitrate encoding, group-of-pictures (GOP) structure, adaptive frame/field MC/DCT (motion compensation–DCT) coding and programmable quantization matrix tables, and so forth. The chip uses multiple clocks for different modules (a 67.5-MHz clock for SRAM control; a 45-MHz clock for motion estimation and motion compensation, which has a wide search range of -288 to +287.5 pixels in horizontal and -96 to +95.5 pixels in vertical; a 22.5-MHz clock for variable-length encoding block; a 13.5-MHz clock for front-end filters; and a 27-MHz clock for the DSP core), yet it only consumes 1.2 W.

The CXD1930Q [33] is another member of Sony Semiconductor's Virtuoso family. It incorporates the MPEG-1/MPEG-2 (main profile at main level) video decoder, MPEG-1/MPEG-2/Dolby Digital AC-3 audio decoder, programmable preparser for system streams, programmable display controller, subpicture decoder for DVD and letter box, and some other programmable modules. The chip targets low-cost consumer applications such as DVD players. The embedded RISC processor in the CXD1930Q is able to support real-time multitasking through Sony's proprietary nano-OS operating system.

5.8 Other Dedicated Codecs

InnovaCom DVImpact [34] is a single-chip MPEG-2 encoder that supports main profile at main level. This chip has been designed from the perspective of the systems engineer; a multiplexing function has been built in so as to relieve the customer's task of writing interfacing code. Although the detailed architecture is not available, it is not difficult to infer that the kernel must be a RISC processor plus a powerful motion estimator, like the ones used in C-Cube's DV^x architecture.

The LSI Logic Video Instruction Set Computing (VISC) encoder chipset [35] consists of three ICs: the L64110 video input processor (VIP) for image preprocessing, the L64120 advanced motion estimation processor (AMEP) for computation-intensive motion search, and the L64130 advanced video signal processor (AVSP) for coding operations such as DCT, zigzag ordering, quantization, and bit-rate control. Although the VIP and AVSP are required in all the configurations, users can choose one to three AMEPs, depending on the desired image quality. The AMEP performs a wide search range of ± 128 pixels in both horizon-

tal and vertical directions. All the three chips need external VRAMs to achieve a high bandwidth. Featuring the CW4001 32-bit RISC (which has a compatible instruction set with MIPS) core, the VIP, AMEP, and AVSP can be programmed using the C/C++ compilers for MIPS, which greatly simplifies the development of the firmware.

Mitsushita Electric Industrial's MPEG-2 encoder chipset [36] consists of a video digital signal processor (VDSP2) and a motion estimation processor (COMET). To support MPEG-2 main profile encoding at main level, two of each are required; but an MPEG-2 decoder can be implemented with just one VDSP2. Inside the VDSP2, there are a DRAM controller, a DCT/IDCT unti, a variablelength-code encoder/decoder, a source data input interface, a communication interface, and a DSP core which further include four identical vector processing units (VPU) and one scalar unit. Each VPU has its own ALU, multiplier, accumulator, shifters and memories based on the vector-pipelined architecture [37]. Therefore the entire DSP core is like a VLIW engine.

Mitsubishi's DISP II chipset [38] includes three chips: a controller (M65721), a pixel processor (M65722) and a motion estimation processor (M65727). In a minimum MPEG-2 encoder system, a controller, a pixel processor, and four motion-estimation processors are required to provide a search range of 31.5×15.5 . Like some other chipsets, the DISP II is also expandable. By adding four more motion-estimation processors, the search range can be enlarged to 63.5×15.5 .

The MVision 10 from VisionTech [39] is yet another real-time MPEG-2 encoder for the main profile at the main level. It is a single-chip Multiple Instruction Multiple Data (MIMD) processor, which requires eight $1M \times 16$ extended data out (EDO) DRAMs and four $256K \times 8$ DRAM FIFOs. Detailed information about the internal architecture is not available.

5.9 Summary of MPEG-2 Encoders

Digital satellite broadcasting and DVD have been offering great market opportunities for MPEG-2. MPEG-2 encoders are important for broadcast companies, DVD producers, nonlinear editing, and so forth, and it will be widely used in tomorrow's video creation and recording products (e.g., camcorders, VCRs, and PCs). Because they reflect the processing ability and represent the most advanced stage of dedicated VSPs, we summarize them in Table 8.

5.10 Commentary

Dedicated video codecs, which are optimized for one or more video-compression standards, achieve high performance in the application domain. Due to the complexity of the video standards, all of the VSPs have to use microprogrammable

		Chin	Supported	Encoding bit rate	Var.	Group of	Search range (pixels)		External memory
Vendor	Product(s)	count	main level	(Mbit/sec)	bit rate	pictures	Н	V	for MP@ML
C-Cube	DV ^x 5110	1	SP, MP	2-15	Yes	I, IP, IBP, IBBP	±202	±124	8 MB DRAM
	DV ^x 6210	2	SP, MP, 4:2:2	2-50	Yes	I, IP, IBP, IBBP	±202	±124	16 MB DRAM
	CLM 4725	7	SP, MP	3-15	Yes		$\pm 100^{P}$	$\pm 28^{\text{P}}$	14 MB DRAM
							±52°	±24°	
IBM	MPEGME	3	SP, MP,	1.5 - 40	Yes	I, IP, IBP	±64	±56	5–6 MB DRAM
	30/31		4:2:2						256 KB SRAM
Innova Com	DV Impact	1	SP, MP			I, IP, IBP, IBBP, IBBBP	±64	±64	9 MB DRAM
LSI Logic	VISC	5	SP, MP	2-15			± 128	± 128	10-14 MB VRAM
Matsushita	VDSP2 &	4-6	SP, MP	Up to 15			$\pm 48^{P}$	$\pm 48^{P}$	14 MB DRAM
	COMET		*				±32 ^в	±32 ^в	
Mitsubishi	DISP II	6 or 10	SP, MP	1-20			$\pm 64^{P}$	±16	5.5 MB SDRAM
			,				±32 ^в		
NTT	ENC-C &	2	SP	1.8-15			± 48.5	± 24.5	512 KB VRAM
	ENC-M								2 MB SDRAM
Sony	CXD1922Q	1	SP, MP	Up to 25	Yes	I, IP, IBP, IBBP	± 288	±96	8 MB SDRAM
Vision Tech- nology	Mvision10	1	SP, MP	1–24			±50	±34	16 MB DRAM 1 MB FIFO

Table 8Summary of MPEG-2 Encoders

RISC cores. It is important for the dedicated VSPs to be configurable or programmable to accept different compression standards and/or parameters.

6 PROGRAMMABLE VSP

Although MPEG is an important application, it is only one of many video applications. Therefore, it would be extremely helpful to develop highly programmable VSPs that can support a whole range of applications. Video applications are continuously becoming more complex and diverse. While still working hard on MPEG-2, people have already proposed MPEG-4, and MPEG-7 is on the schedule. Apparently, dedicated VSPs cannot keep pace with the rapid evolution of new and some existing video applications.

Although usually less expensive, dedicated VSPs may not be the overall winner in a comprehensive system. We might need quite a few different dedicated VSPs in a complicated system which must support several different multimedia applications such as video compression/decompression, graphics acceleration, and audio processing. Furthermore, the development cost of dedicated VSPs is not inexpensive, as the designers must hand-tune many parts to achieve the best performance/cost ratio. Because of the large potential market demand, programmable VSPs seem to be relatively inexpensive.

Consequently, the need for greater functionality, as well as increased cost and time-to-market pressures, will push the video industry toward programmable VSPs. The industry has already seen a similar trend in modem and audio codecs. More and more, new systems incorporate DSPs instead of dedicated controllers.

Generally speaking, all of the VSPs are programmable to some degree; some of them have multiple powerful microprocessors, some have a RISC core and several coprocessors, and others only have programmable registers for system configuration. Our definition of "programmable" excludes the last category. Due to the complexity of video encoding algorithms, dedicated encoders have to use a processor core and many special-purpose functional units optimized for various parts of the algorithm, such as a motion-estimation unit, a vector quantization unit, a variable-length code encoder, and so on. By loading different microcodes into the core processor, the chip is able to generate different data formats for different standards. As a matter of fact, most of the dedicated video encoders support several standards. A demonstrative example is the DV^x architecture developed by C-Cube mentioned earlier. The kernel of this architecture is a 32-bit embedded RISC CPU; it also contains two programmable coprocessors. However, the architecture is designed and optimized for MPEG-2 encoding and decoding, not for general video applications.

What we are interested in is highly programmable VSPs that are more flexible and adaptable to new applications. These chips would be somewhat similar to general-purpose processors in terms of functionality and programmability. However, the difference is that the VSPs are dedicated to video signal processing. Many video applications belong to the category of scientific calculation and have some good properties such as regular control flow and symmetry in data processing. For a variety of video applications, including H.263, MPEG-1, MPEG-2, and MPEG-4, the whole video frame is divided into several blocks and then the data processing procedure for each block is exactly the same. This kind of symmetry carries a huge amount of parallelism. A high-performance programmable VSP must have a well-defined parallel architecture that is capable of exploring the potential parallelism.

All of the VSPs in Tables 9 and 10 are programmable to some extent. They all have at least one internal microprocessor, on which programs or microcodes are running. However, not all of the manufacturers provide developing tools for users to implement and test their own applications. Some only provide firmware necessary to support end applications.

6.1 Chromatic Research Mpact2 Media Processor

Media processors are different from traditional VSPs in that they are not only dedicated to accelerate video processing but are also capable of improving other multimedia functions (e.g., audio, graphics). The Mpact media processor [40] is a low-cost, multitasking, supercomputerlike chip that works in conjunction with an x86/MMX processor to provide a wide range of digital multimedia functions. In an Mpact-based multimedia system, specialized software called Mpact mediaware runs on both the Mpact chip and x86 processor, delivering multimedia functions such as DVD, videophone, video editing, 2D/3D graphics, audio, fax/ modem, and telephony.

The block diagram of the data path of Mpact2 R/6000 is shown in Figure 9. Two major enhancements were made to Mpact2 over the initial Mpact architecture to support 3D graphics: a pipelined floating-point unit and a 3D-graphics rendering unit (ALU group 6) and its associated texture cache. In each cycle, the Mpact2 can start a pair of floating-point add operations and a pair of floating-point multiply operations, yielding a peak performance of 500 MFLOPS (mega floating-point operations per second) at 125 MHz. The dedicated 3D-graphics rendering unit is a 35-stage scan-conversion pipeline which can render 1 million triangles per second. The Mpact2 also has a VLIW core, for which each instruction word is 81 bits long and contains two instructions which may cause eight single-byte operations to be executed on each of the four ALU groups (groups 1–4). The Mpact2 data paths are all 72 bits wide. Data exchanges are done via a crossbar on a 792-bit interconnection bus, which can transfer eleven 72-bit results simultaneously at an aggregate bandwidth of 18 Gbyte/sec.

In multimedia applications, external bandwidth is as critical as internal throughput. Chromatic chose Rambus RDRAMs because they provide a high bandwidth at a very low pin count. The Mpact2 memory controller supports two

Vendor	Processor(s)	Application(s)	Architecture	Peak perform.	Technology
Chromatic Mpact2 R/6000 Research		MPEG-1 encoding MPEG-1 & -2 decoding Windows GUI acceleration 2D/3D graphics H.320/H.324 videophone Audio, FAX/modem	VLIW (SIMD) 6 ALU groups, 2 Rambus RAC channels, and 5 DMA bus controllers	125 MHz 6 BOPS	0.35 μm, 352- BGA, 3.3 V
MicroUnity	Cronus	Communicating and pro- cessing at broad-band rates	Single instruction group data, 5 threads	300 MHz	0.6 μm, 441- BGA, 3.3 V
Mitsubishi	D30V	MPEG-1 & -2 decoding Dolby AC-3 decoding H.263 codec 2D/3D graphics; modem	Two-way VLIW RISC core	250 MHz 1 BOPS	0.3 μm, 135- PGA, 2V
NEC	V830R/AV	MPEG-1 encoding MPEG-1 & -2 decoding	Superscalar RISC core and SIMD multimedia exten- sion	200 MHz 2 BOPS	0.25 μm, 208- PQFP, 2.5 V, 2 W

Table 9Summary of Some Programmable VSPs by Vendor

Philips	TriMedia TM-1000	MPEG-1 encoding MPEG-1 & -2 decoding 2D/3D graphics H.32x videophone V.34 modem	VLIW core with 27 function units and 5 issue slots; coprocessors: video I/O, audio I/O, timer, image coprocessor, etc.	100 MHz 4 BOPS	0.35 μm, 240- BGA, 3.3 V, 4 W
Samsung	MSP-1	MPEG-1 & -2 decoding H.324 codec 2D/3D graphics AC-3 decoding, wave-table Modem, telephony	ARM7 RISC core and vector processor in shared memory	100 MHz 6.4 BOPS 8-bit int. 1.6 GFLOPS	0.5/0.35 μm, 128/256 pin, 3.3 V, 4 W
Texas Instru- ments	TMS320-C80 (MVP)	MPEG-1 encoding/decoding H.261 codec JPEG; 2D/3D graphics Image and vector processing	MIMD architecture with 4 in- teger DSPs and 1 floating- point RISC	50 MHz 2 BOPS 100 MFLOPS	305-CPGA, 3.3 V
	TMS320-C6201	Modems; servers Wireless based stations Multi-channel telephony	VLIW DSP core with 8 func- tional units and dual data- path	200 MHz 1.6 BOPS	0.25 μm, 352- BGA, 2.5 V, 4.2 W
	TMS320-C6701	High-performance fixed- and floating-point digital signal processing	TMS320C6201 added float- ing-point units	167 MHz 1 GFLOPS	0.18 μm, 352- BGA, 1.8 V

Processor(s)	Func. units	Issue slots	Data width (bit)	Register file	Inst. width (bit)	Inst. \$/RAM	Data \$/RAM	Float. point	Develop. tools
Mpact2	8	2	72	512 × 72	81	20 Kb	18-Kb texture cache	Yes	No
Cronus	5	1	128	$5 \times 64 \times 64$	32	256 Kb	256-Kb cache/buffer	No	C/C++, assembly
D30V	7	2	32	64×32	64	256 Kb	256 Kb RAM	No	
V830R/AV	V 3	1	32	32×32	16/32	128 Kb	128-Kb cache	No	
	M 6	1	64	32×64	32				
TriMedia TM-1000	27	5	32	128×32	220	256 Kb	128-Kb cache	Yes	C/C++, assembly
MSP-1	2	2	32	31×32	32	16 Kb	40-Kb cache	Yes	MSP tools
TMS320-C80	5	1	32	31×32	32	32 Kb	32-Kb cache	Yes	C, assembly
(MVP)	4×5	4×4	32	$4 \times 44 \times 32$	64	4 imes 16 Kb	272-Kb RAM		•
TMS320-C6201	8	8	32	32×32	256	512 Kb	512-Kb data RAM	No	C, assembly
TMS320-C6701	8	8	32	32×32	256	512 Kb	512-Kb data RAM	Yes	C, assembly

 Table 10
 Summary of Some Programmable VSPs by Processor



Figure 9 Architecture of Mpact2 datapath. (From Ref. 41.)

9-bit 300-MHz Rambus channels of RDRAM media memory (1.2 Gbyte/sec), which store both instructions and data. All of the other Mpact2 I/O ports have also been improved from the first generation: The PCI interface now operates at 66 MHz, enabling a peak transfer rate of 264 Mbyte/sec; the display interface incorporates a 220-MHz RAMDAC on-chip; and the digital video interface becomes fully duplex.

The Mpact2 media processor runs an optimized real-time multitasking kernel to simultaneously execute several Mpact mediaware modules. This kernel, along with the programming model and instruction set, is completely proprietary and developed in-house only. Besides the drivers for accelerating multimedia applications, Chromatic does not provide any tools for users to write their own microcode.

6.2 MicroUnity's Broad-Band Media Processor

"A broadband media processor is a general-purpose processor system with sufficient computing resources to communicate and process digital, audio, data, and radio frequency signals at broadband rates (more than 1.5 Mbit/s)" [42]. The MicroUnity broad-band media processor is intended to be the sole processor in client or terminal systems. As such, it is a general-purpose microprocessor, including a memory management unit. Figure 10 shows the architecture of the media processor. In the first BiCMOS implementation, the 1-GHz clock drives a 512-Gbit/sec computing bandwidth and a 128-Gbit/sec memory bandwidth. An integrated SDRAM interface supports a peak bandwidth of 3.2 Gbit/sec. The media channel also offers a great communication bandwidth as high as 32 Gbit/ sec, which can be used to construct a multiprocessor system. A very simple packet



Figure 10 Structure of MicroUnity broad-band media processor. (From Ref. 43.)

control protocol is used in the multiprocessor system, with several types of hardware support such as low-latency memory-mapped I/O.

The MicroUnity media processor exploits parallelism from two perspectives: group instructions and multithreading. The group instructions specify operations on four 128-bit register pairs, totaling a bandwidth of 512 bits/instruction. This architecture, referred to as Single Instruction Group Data (SIGD), is almost exactly the same as the instruction set extensions introduced in Section 6, and a very similar concept can also be found in Texas Instruments' MVP, where the splittable ALUs can be reconfigured dynamically. There is some difference, however, in the size and number of operands. Although most other general-purpose microprocessors work on two 64-bit source operands, MicroUnity's media processor can take up to three source register pairs, each 128 bits long (a register pair consists of two 64-bit registers and can be used to represent different data granularities from two 64-bit words to 128 single bits). In order to deal with unaligned or mixed-precision data, the broad-band media processor also provides switching instructions which can shift, shuffle, extract, expand, and swizzle operands as well as other kinds of manipulation. These switching instructions are much more powerful than any other processors. Other instructions include control instructions, which can effectively reduce branch overhead. For example, branchgateway instruction fetches 128 bits from memory into a pair of registers (code and data pointers, respectively) while checking translation lookaside buffer (TLB) for access control; then it jumps to the code pointer, storing a result link in its space. This is extremely helpful for active message in message passing based multiprocessor systems. In addition, MicroUnity's media processor provides extended math operations such as multiply over 8-bit Galois fields [i.e. GF(256)]. This makes the processor not only suitable for source coding (e.g., MPEG-2) but also useful for channel coding (e.g., Reed–Solomon error-correcting code). All of the instructions are summarized in Table 11.

Another key feature of the architecture is that it supports multithreading. Like superscalar, multithreading is another technique to exploit instruction-level parallelism. A thread is an encapsulation of the control flow in a program. Multi-threaded programs have multiple threads running through different code paths concurrently, so multithread processors interleave a number of threads in a single execution pipeline. Although in a fine-grain view, only one instruction from one thread is running in a single pipeline, this technique can hide dependency. Instead of waiting for a stalled instruction, which would cause a pipeline bubble, multi-threaded processors switch to another thread, which is independent of the previous one. By the time the processor is back on the thread that had dependency constraints, the next instruction in the thread can be issued immediately as the dependency would have gone, maximizing pipeline utilization. In MicroUnity's media processor, five hardware-based threads of control share a common datapath, instruction cache, data cache, DRAM interface, and a set of I/O interfaces.

MicroUnity provides an open programming platform. Their software development environment includes assembler, C/C++ compiler, source-code debugger, profiler, and media and communications software libraries for various standards. In addition, the company also has a real-time micro kernel for client devices and a 64-bit Open Software Foundation UNIX for server applications [44].

6.3 Mitsubishi D30V

The D30V [45] is not really a VSP but an optimized processor core for video signal processing. By integrating a few other components, an MPEG-2 decoder can be implemented on a single chip. The architecture of the D30V core is shown in Figure 11.

There are three execution units in the D30V processor core: a memory unit, an integer unit, and a branch unit (inside the instruction decode unit). In addition to program sequencing control, the memory unit is also able to crunch data in its ALU and shifter. It supports several data types from byte (signed and unsigned) to 64-bit word. Its load and store instructions can operate on multiple operands using packing and unpacking. Other features of the memory unit include postincrement/decrement and modulo addressing, which ease programming. The integer unit contains a 32-bit ALU, a 32-bit shifter, a 32×32 multiplier, and two 64-bit accumulators. The 64-bit accumulators support 64-bit arithmetic, and all of the other functional units operate on both 32-bit and 16-bit operands. The integer unit has been optimized for video applications. It has subword and halfword operations to further exploit parallelism, and a few added video operations

Category	Instructions	Optional features	Interval-issue-latency (cycles)
Storage (8, 16, 32, 64, or 128 bits) and synchronization	Load 8, 16, 32, 64, or 128 bits, little- or big-endian	Unsigned, aligned, immediate	2-1-2
(64 bits)	Store 8, 16, 32, 64, or 128 bits, little- or big-endian	Aligned, immediate	4-1-0
	Store add, compare, or multi- plex 64 bits	Immediate, -and-swap	8-7-7
Branch (64 bits)	Branch and-equal, and-not-		2-1-1 for pipelined,
	equal, less, or less-equal-zero		2-1-4 for unpipelined
	Branch equal, not-equal, less, or greater-equal		2-1-1
	Branch floating-point equal, not-equal, less, or greater- equal (16, 32, 64, or 128 bits)		
	Branch	Immediate, -and-link	2-1-1
	Branch gateway	Immediate	2-2-1
	Branch down or back		2-1-1
Fixed point (64 bits) and group	Add or subtract	Immediate, overflow	1-1-1
$(128 \times 1, 64 \times 2, 32 \times 4,$	Multiply	Unsigned, -and-add	1-5-7 for 32-bit,
$16 \times 8, 8 \times 16, 4 \times 32$, or 2×64 bits)	Divide	Unsigned	1-20-22 for 64-bit multiply, 1-23-25 for 64-bit multiply- and-add,
			and-add, 1-2-4 for others

	AND, OR, AND-NOT, OR- NOT, XOR, XNOR, NOR, or NAND	Immediate	1-1-1
	Shuffle, deal, or swizzle		1-1-2
	Compress or expand	Unsigned, immediate	1-1-2
	Extract	Unsigned, immediate	1-2-3
	Deposit or withdraw immediate	Unsigned, merge	1-1-2
	Shift or rotate right or left	Unsigned, immediate, overflow	1-1-2
	4- or 8-Way multiplex	Shuffle, transpose	1-1-2
	Select bytes		1-1-2
	Set or sub, equal, not-equal, less, or greater-equal	Unsigned, immediate	1-1-2
	Multiplex		1-1-1
	AND sum of bits		1-1-3
	Log most significant bit		1-1-2
	Galois-field multiply, polyno- mial multiply-divide, 8 or 64 bits		1-4-5
Floating-point scalar (16, 32, 64, or 128 bits) and group	Add, subtract, multiply, or divide	Near, truncate, floor, ceiling, or exact	
$(8 \times 16, 4 \times 32, \text{ or } 2 \times 64)$ bits)	Multiply-and-add or -subtract	Near, truncate, floor, ceiling, or exact	
	Square-root, sink, float, or deflate	Near, truncate, floor, ceiling, or exact	
	Absolute, negate, inflate	Exception	
	Set equal, not-equal, less, greater-equal	Exception	

Source: Ref. 42.



Figure 11 Mitsubishi D30V processor core. (From Ref. 46.)

such as variable length saturation instruction, join instruction, add sign instruction, and so on. The branch unit has a variable number of delay slots and additional conditional branches (e.g., test zero and branch, test notzero and branch), which enable zero-delay branches and zero-overhead loops. All of the functional units in the D30V core are fully pipelined using a four-stage pipeline; they are controlled by a 64-bit VLIW instruction, which contains two short or one long RISC subinstructions. The dual-issue processor has used some advanced techniques to improve the performance. These techniques include predicated execution and speculative execution.

The D30V processor core was designed not only to meet computational requirements and cost but also to provide the flexibility of programmable processors. However, the dual-issue processor is not powerful enough for computation-intensive video applications like MPEG-2 encoding. In an implementation of an MPEG-2 decoder, two D30V cores and several small processing units are required in addition to a dedicated motion-estimation processor.

6.4 Philips TriMedia

The Philips TriMedia [47] is also a general-purpose microprocessor enhanced for multimedia processing. As shown in Figure 12, at the center of the chip is a 400-Mbyte/sec high-speed backbone that connects autonomous modules and provides accesses to internal control registers. The data highway, consisting of a 32-bit data bus and a 32-bit address bus, uses a block transfer protocol, which can transfer 64 bytes at a burst.

TM-1000 incorporates independent DMA-driven peripheral units and coprocessors to streamline data throughput. These on-chip processing units can be masters or slaves on the data highway, and they manage input, output, and formatting of multimedia data streams as well as perform specific functions. While sending an image from SDRAM to the video frame buffer (could be in the SDRAM or the host's PCI-based graphics system), the image coprocessor



Figure 12 Architecture of TriMedia. (From Ref. 48.)

can perform horizontal or vertical image filtering and scaling, YUV to RGB color space conversion, as well as overlay live video on background image. The variable-length decoder (VLD) coprocessor offloads the VLIW CPU by decoding Huffman-encoded video bit streams. Due to the characteristics of the algorithm, this task has little inherent parallelism and, hence, is not suited for VLIW processing. The two coprocessors are microprogrammable. They are independent of the VLIW CPU and are synchronized with it using an interrupt mechanism.

The VLIW processor has a rich instructions set (197 instructions), including many extensions for handling multimedia data types. Parallelism is achieved by incorporating 27 functional units in the VLIW engine and feeding them with five instruction issue slots. The type and number of functional units are listed in Table 12. All of the functional units are pipelined, with a depth ranging from 1 to 17 stages. The five constant units do not perform any calculation except providing ports for accessing immediate values stored in the instruction word. Like many other processors, TM-1000 also provides pack/unpack and group instructions, which can manipulate 4 bytes or two 16-bit words at one time, exploiting subword parallelism. Other special instructions include *me8* for motion estimation, which is similar to the *PDIST* instruction in UltraSparc's visual instruction set. Most instructions accept a guard register for predicated execution. Although the TM-1000 processor has 27 functional units, in each cycle it can issue only up to 5 instructions.

The TM-1000 has a dedicated instruction cache and a data cache on-chip, both of which are eight-way set-associative with LRU (least-recently used) replacement and locking mechanism to improve performance. The 16 KB dual-

Name	Quantity	Latency (cycles)	Recovery (cycles)
Constant	5	1	1
Integer ALU	5	1	1
Memory load/store	2	3	1
Shift	2	1	1
DSP ALU	2	2	1
DSP multiply	2	3	1
Branch	3	3	1
Floating-point ALU	2	3	1
Integer/floating-point multiply	2	3	1
Floating-point compare	1	1	1
Floating-point sqrt/divide	1	17	16

Table 12 TM-1000 Functional Units

Source: Refs. 47 and 48.

ported data cache allows two simultaneous nonblocking accesses. To save bandwidth and storage space, the VLIW instructions are stored and cached using a 2-23-byte compressed format until they are fetched from the instruction cache. The chip also has a glueless interface to support four banks of external SDRAMs.

The TM-1000 development environment includes a VLIW compiler, a C/ C++ software development environment, and the pSOS+ real-time operating system.

The TriMedia CPU64 [49] is a successor to the TM-1000. The CPU64 has a 64-bit word and uses subword parallelism within the VLIW CPU to increase parallelism on small data words.

6.5 Samsung Media Signal Processor

The Samsung media signal processor MSP-1 [50] is a cache-based dual-processor architecture (Fig. 13). The architecture consists of a floating-point vector processor for digital signal processing, an ARM7 RISC CPU for system control and management, a bit-stream processor for parsing the video stream, I/O interfaces, and 10K unused gates for optimal customization. The vector processor, running at 100 MHz, supports various type of integer (from bytes to 32-bit words) and 32-bit IEEE 754 floating-point numbers, with a peak performance of 6.4 billion operations per second (BOPS) 8-bit integers and 1.6 BOPS 32-bit floating points. The ARM7 CPU, running at 50 MHz, is responsible for general functions such as real-time scheduling. Both processor share the same cache subsystem and can operate simultaneously. In this dual-processor architecture, the bulk of the



Figure 13 MSP-1 microarchitecture. (From Ref. 50.)

video processing is performed by the vector processor because the RISC CPU only deals with general system functions.

The MSP is a fully programmable media processor with a rich instruction set, including standard ARM RISC instructions for scalar processing, high-performance SIMD instructions for vector processing, I/O instructions for block load/store, and special instructions for filtering and MPEG applications. The programming model also has macro library instructions such as *DCT*, *CONV*, and *MULM*. The software development tools include MSP-oriented assembler, compiler, linker, debugger, and simulator.

6.6 Texas Instruments' TMS320C8x Multimedia Video Processor

Texas Instruments' TMS320C8x (MVP) [51] family has four members and two architectures. The TMS320C80 is a highly integrated multiprocessor and was an early single-chip MPEG-1 encoder. As shown in Figure 14, the C80 includes four advanced integer DSPs (ADSPs) and a floating-point RISC master processor (MP), which are integrated with a transfer controller (TC), a video controller (VC), and five memory banks. The TMS320C80 allows 5 instruction fetches and 10 parallel data accesses in each cycle, allowing a transfer rate as high as 1.8 Gbyte/sec for instruction and 2.4 Gbyte/sec for data. The younger member, TMS320C82, is a scaled down version of the TMS320C80. It provides better cost/performance ratio for some cost-sensitive applications by removing two integer DSPs, the video controller, and some on-chip memory.

The master processor of C80 is a general-purpose RISC processor with an IEEE 754 compatible three-stage floating-point pipeline. The RISC processor is



Figure 14 Architecture of Texas Instruments' TMS320C80 (MVP). (From Ref. 51.)

powered by 32-bit instructions and can issue, in each cycle, a parallel multiply, an add, and a 64-bit load/store, yielding 100 MFLOPS at 50 MHz. The floating-point unit can perform both single- and double-precision arithmetic.

Each of the four ADSPs is a 32-bit integer DSP optimized for bit- and pixel-oriented imaging and graphics applications. Each parallel processor can issue, in each cycle, a multiply, an ALU operation, and two memory accesses within a single 64-bit instruction. The parallelism comes from two independent datapaths. The multiplier datapath includes a three-stage 16×16 multiplier, a half-word swapper, and rounding hardware. The ALU datapath includes a 32bit three-input ALU, a barrel rotator, a mask generator, a 1-bit to *n*-bit expander, a left/rightmost and left/rightmost bit-change logic, and several multipliers. The 32-bit three-input ALU can perform all of the 256 three-input Boolean combinations as well as many other mixed logical and arithmetic operations. Both the multiplier and the ALU are splittable. Although the 16×16 multiplier can be split into two 8×8 multipliers, the 32-bit ALU can be divided into two 16-bit ALUs or four 8-bit ALUs. The big register file contains 8 data registers, 10 address registers, 6 index registers, and 20 other user-visible registers. Three hardware loop controllers enable zero-overhead looping/branching and multiple-loop end points. The ADSPs provide conditional operation (also referred to as predicated execution).

The video controller handles both video input and output and can simultaneously support two independent capture or display systems. The transfer controller combines a memory interface and a DMA engine, handling data movement within the MVP system as requested by the master processor, parallel processors, video controller, and external devices. In addition to handling internal block transfers, it also supports direct interface to SRAM, DRAM, SDRAM, and VRAM, with a peak external bandwidth of 400 Mbyte/sec.

Processing elements in the MVP architecture communicate by a global crossbar (actually an incomplete crossbar because it is missing some interconnect paths), which provides internal fast data communication on a cycle-by-cycle basis. The crossbar automatically sets up the connection between a module and a memory bank every cycle without special configuration instructions. It allows 5 instruction fetches and 10 parallel data accesses at a time, totaling 1.8 Gbyte/ sec instruction transfer and 2.4 Gbyte/sec data transfer.

The C80 comes with an open programming environment which includes assembler, C compiler, linker, simulator, debugger, and emulator.

6.7 Texas Instruments' VelociTI

Texas Instruments' VelociTI [52] is an advanced VLIW architecture for digital signal processing. Although it is not designed for video applications and contains no functional units dedicated to video operation, the potential performance can greatly accelerate some multimedia applications. Therefore, we highlight its features here.

Announced in February 1997, TMS320C6201 is the first member of the VelociTI architecture and it offers a breakthrough for digital signal processing by enabling a sustained throughput of up to eight 32-bit instructions every cycle, achieving 1600 MIPS at 200 MHz. As shown in Figure 15, the C6201 has two independent 32-bit datapaths, each having four functional units (.L, .S,. M, and .D; see descriptions in Table 13) and a register file containing 16, 32-bit registers. A data bus joins the two register files and provides fast data exchanges between them, so logically there is only one unified register file. The instruction set is in favor of digital signal processing: It supports saturation, bit-field set/clear and extract, bit counting, normalization, and various data types from a 1-bit to a 32bit word. All instructions are conditional (predicated) to provide more parallelism. The 4-Gbyte space is byte-addressable with dual-endian support and a variety of addressing modes, including circular addressing with a 5-15-bit offset. The C6201 incorporates large amount of memory on-chip. The 1-Mbit RAM is split between data and instruction memory. The external memory interface (EMIF), capable of operating at 200 MHz, supports both SRAM and synchronous DRAM.

Using the same VelociTI architecture, the C67x extends the C6201 by adding floating-point capability to six of the eight functional units. The new instruction set is a superset of the integer DSP, and old binary programs can be run without any modification. The new processor can execute two floating-point arith-



Figure 15 Architecture of Texas Instruments' TMS320C6201. (From Ref. 52.)

Functional unit	Description		
.L (.L1 and .L2)	32/40-Bit arithmetic and compare operations		
	Finds leftmost 1 or 0 bit for 32-bit register		
	Normalization count for 32 and 40 bits		
	32-Bit logical operations		
.S (.S1 and .S2)	32-Bit arithmetic operations		
. ,	32/40-Bit shifts and 32-bit bit-field operations		
	32-Bit logical operations		
	Branching		
	Constant generation		
	Register transfers to/from the control register file		
.M (.M1 and .M2)	16×16 -Bit multiplies		
.D (.D1 and .D2)	32-Bit add, subtract, linear, and circular address calculation		

Table 13 Functional Units and Descriptions

Source: Ref. 52.

metic operations, two floating-point reciprocal/absolute value/square root operations, and two floating-point multiply operations per cycle, resulting in 1 GFLOPS at 167 MHz.

This architecture also supports an open programming environment. The C compiler performs a variety of optimizations, including software pipelining, which can effectively improve code performance on VLIW machines.

6.8 Commentary

Programmable VSPs represent a new trend in multimedia systems. They tend to be more versatile in dealing with multimedia applications, including video, audio, and graphics. VSPs have to be very powerful because the amount of computation required by video compression is enormous. To meet the performance demands, all of the VSPs employ parallel processing techniques to some degree: VLIW (SIMD), multiprocessor-based MIMD, or the concept from vector processing (SIGD). However, none of these programmable VSPs are able to compete with dedicated state-of-the-art VSPs—none of them could support real-time MPEG-2 encoding yet.

It is not surprising to see that many programmable VSPs adopt VLIW architecture. There are basically two reasons for doing this. First, there is much parallelism in video applications [53]. Second, in VLIW machines, a high degree of parallelism and high clock rates are made possible by shifting part of the hardware workload to software. This kind of shift once happened in the microprocessor evolution from CISC to RISC. By relieving the hardware burden, RISC achieved a new level that CISC was unable to compete with and the revolution has been a milestone in microprocessor history. Analogously, we would expect VLIW to outperform other architectures. Unlike their superscalar counterparts, VLIW processors rely on the compilers entirely to exploit the parallelism; static scheduling is performed by sophisticated optimizing compilers. All of this raises challenges for next-generation compilers. More discussions on the VLIW architecture as well as its associated compiler and coding techniques can be found in Fisher et al.'s review [54]. Although offering architectural advantages for general-purpose computing (where unpredictability and irregularity are high), multithreading architectures are not as optimal for video processing where regularity and predictability are much higher.

7 RECONFIGURABLE SYSTEMS

Reconfigurable computing is yet another approach to balancing performance and flexibility. In contrast with VSPs where the programmability relies in the instruction set architecture (ISA), the flexibility of reconfigurable systems comes from

Solution	Performance	Flexibility	Power	Cost	Density
Multimedia instruction extension	Low	High	Medium	High	High
Application-specific codec	High	Low	Low	Low	Medium
Programmable VSP Reconfigurable computing	Medium Medium	High Medium	Medium High	High Medium	Medium Low

 Table 14
 Comparison of Different Solutions for Video Signal Processing

a much lower level—logic gate arrays. Table 14 compares different solutions for video signal processing from several perspectives.

Reconfigurable computing has evolved from the original field programmable gate array (FPGA), which was invented in the early 1980s and has been undergoing vast improvements ever since. Traditionally, FPGAs were only used as a replacement of glue-logic and fast prototyping, but their applications have been widened in the past decade. The introduction of SRAM-based FPGAs by Xilinx Corporation [55] in 1986 opened a new era. SRAM-based FPGAs use SRAM cells to store logic functionality and wiring information and thus can be programmed an infinite number of times. Almost all of the modern FGPAs choose look-up-table (LUT)-based design, where the each logic cell consists of one or two LUT units, each driven by a limited number of inputs. The LUT units can be configured to implement any multiple-input (usually less than five) singleoutput Boolean function, providing fine-grained parallelism. With technology advances, the density (reported in equivalent gate counts) of state-of-the-art FPGAs is approaching 1 million gates. Further discussion on FPGA technologies is beyond the scope of this survey, so we refer interested readers to other literature [56]. In the following subsections, we will focus on using reconfigurable computing for video signal processing.

7.1 Implementation Choices

Unlike the other three approaches we have discussed previously, we have not yet seen single-chip or even chipset solutions for reconfigurable video signal processing. However, there are systems existing for this application. Reconfigurable systems typically consist of a general-purpose microprocessor and some reconfigurable logic such as FPGAs. Although the computational cores are mapped to the reconfigurable logic, the microprocessor takes care of everything else that cannot be implemented efficiently with reconfigurable hardware, including branches, loops, and other expensive operations. A natural question then arises: Where does one draw the boundary between the CPU and the FPGA?

Location	Role	Bus	Bandwidth
Inside CPU	Functional unit	Internal data bus	5-10 Gbyte/sec
CPU-L1 cache] L1-L2 cache]	Closely coupled coprocessor	CPU bus	2–5 Gbyte/sec 1–2 Gbyte/sec
L2-main memory	Loosely coupled coprocessor	Memory bus	500-1000 Mbyte/sec
Beyond main memory	Standalone processor	I/O bus	66-200 Mbyte/sec

Table 15 Implementation Choices for Combining CPU and Reconfigurable Logic

There are many implementation choices for how to combine generalpurpose microprocessor with reconfigurable logic in the CPU-memory hierarchy. As can be seen in Table 15, the closer the configurable logic sits to the CPU, the higher the bandwidth will be. It is difficult to compare in general which one is better, because different applications have different needs and they yield different results on different systems.

Reconfigurable logic supports implementing different logic functions in hardware. This has two implications. First, it means that reconfigurable computing has the potential to offer massive parallelism. Numerous studies have shown that video applications bear a huge amount of parallelism, so, theoretically, reconfigurable computing is a sound solution for video signal processing. Second, LUT-based FPGAs exploit parallelism at a very fine granularity. When dealing with fine-grained parallelism, it is desirable for the reconfigurable logic to sit closer to the CPU. This is because fine-grained parallelism will yield many intermediate results, which requires a high bandwidth to exchange. In this approach, the reconfigurable logic can be viewed as a functional unit, providing functions that can be altered every once a while, depending on the need for reconfiguration. This flexibility can even be built into the instruction set architecture, generating an application-specific instruction set on a general-purpose microprocessor, which can speed up many different applications potentially. Although this idea is very attractive, it comes with a significant cost. In order to be flexible, reconfigurable hardware has a large overhead in the wiring structure as well as inside the LUT-based logic cell. As the silicon resource becomes more and more precious inside microprocessors, it is probably not worth using it for reconfigurable logic; putting some memory or fixed functional units in the same area is likely to yield better performance.

By moving the reconfigurable logic outside a CPU, we may achieve a better utilization of the microprocessor real estate, but we will have to sacrifice some bandwidth. In this approach, reconfigurable resources are used to speed up certain operations which cannot be done efficiently on the microprocessor (e.g., bit-serial operations, deeply pipelined procedures, application-specific computations, etc.). Depending on the application, if we can map a coarser level of parallelism to the FPGA coprocessor, we may still benefit considerably from this kind of reconfigurable computing.

For coarse-grained parallelism or complex functions, we may need to seek heterogeneous system solutions. In this approach, commercial FPGAs are used to form a stand-alone processing unit which performs some complicated and reconfigurable functions. Because the FPGAs are loosely coupled with the microprocessor, this kind of system is often implemented as an add-on module to an existing platform. Like other alternatives, this one has both advantages and disadvantages. Although it can implement very complicated functions and be CPUagnostic, it introduces a large communication delay between the reconfigurable processing unit and the main CPU. If not carefully designed, the I/O bus in between can become a bottleneck, severely hampering the throughput.

In addition to the whereabouts of reconfigurable logic in a hybrid system, there are many other hardware issues, such as how to interconnect multiple FPGAs, how to reconfigure quickly, how to change part of a configuration, and so forth. Due to limited space, we refer users to some good surveys on FPGA [56,57].

7.2 Implementation Examples

The fine granularity of parallelism and pipelined nature of reconfigurable computing make it a particularly good match for many video processing algorithms. Among several implementation options for reconfigurable computing system, it is not clear which one is the winner. In the following paragraphs, we will enumerate a few reconfigurable computing systems, with emphasis on the hardware architecture instead of the application software. Note that our interests are in multimedia applications, so we will not address every important reconfigurable system.

Splash II [58], a systolic array processor based on multiple FPGAs, is one of a few influential projects in the history of reconfigurable computing. The 16 Xilinx FPGAs on each board in Splash II are connected in a mesh structure. In addition, a global crossbar is provided to facilitate multihop data transfer and broadcast. To synchronize communication at the system level, a high-level inter-FPGA scheduler as well as a compiler are developed to coordinate the FPGAs and the associated SRAMs. Among many DSP applications that have been mapped to the Splash II architecture, there are various image filtering [59], 2D DCT [60], target recognition [61], and so forth.

Another important milestone during the evolution of reconfigurable computing is the Programmable Active Memory (PAM) [62] developed by DEC (now Compaq). PAM also consists of an array of FPGAs arranged in a two-dimensional mesh. With the interface FPGA, PAM looks like a memory module except that the data written in may be read out differently after being massaged by the reconfigurable logic. PAM has also been used in numerous image processing and video applications, including image filtering, 1D and 2D transforms, stereo vision, and so forth.

The Dynamic Instruction Set Computer (DISC) [63] is an example of integrating FPGA resources into a CPU. Due to limited silicon real estate, DISC treats instructions as swappable modules and pages them in and out continuously during program execution through partial reconfiguration. In some sense, the onchip FPGA resources function like an instruction cache in microprocessors. Each instruction in DISC is implemented as an independent module, and different modules are paged in and out based on the program needs. The advantage with this approach is that limited resources can be fully utilized, but the downside is that the context switching can cause delay, conflict, and complexity. DISC adopts a linear, one-dimensional hardware structure to simplify routing and reallocation of the modules. Although the logic cells are organized in an array, only adjacent rows can be used for one instruction. The width of each instruction module is fixed, but the height (number of rows) is allowed to vary (Fig. 16). In an experiment of mean image filtering, the authors reported a speedup of 23.5 over a general-purpose microprocessor setup.

The Garp architecture combines reconfigurable hardware with a standard MIPS processor on the same die to achieve high performance [64]. The top-level block diagram of the integration is shown in Figure 17. The internal architecture



Figure 16 Linear reconfigurable instruction modules.



Figure 17 Block diagram of Garp.

of the reconfigurable array is very similar to DISC (Fig. 16). Each row of the array contains 23 logic blocks, each capable of handling 2 bits. In addition, there is a distributed cache built in the reconfigurable array. Similar to an instruction cache, the configuration cache holds the most recent configuration so as to expedite dynamic reconfiguration. Simulation results show speedups ranging from 2 to 24 against a 167-MHz Sun UltraSPARC 1/170.

The REMARC reconfigurable array processor [65] also couples some reconfigurable hardware with a MIPS microprocessor. It consists of a global control unit and 8×8 programmable logic blocks called nanoprocessors (Fig. 18). Each nanoprocessor is a small 16-bit processor: It has a 32-entry instruction RAM, a 16-entry data RAM 1 ALU, 1 instruction register, 8, 16-bit data registers, 4 data input registers, and 1 data output register. The nanoprocessor are interconnected in a mesh structure. In addition, there are eight horizontal buses and eight vertical busses for global communication. All of the 64 nanoprocessors are controlled by the same program counter, so the array processor is very much like a VLIW processor. REMARC is not based on FPGA technology, but the authors compared it with an FPGA-based reconfigurable coprocessor (which is about 10 times larger than REMARC) and found that both have similar performance, which is 2.3– 7.3 times as fast as the MIPS R3000 microprocessor. Simulation results also show that both reconfigurable systems outperform Intel MMX instruction set extensions.

Used as an attached coprocessor, PipeRench [66] explores parallelism at a coarser granularity. It employs a pipelined, linear reconfiguration to solve the problems of compilability, configuration time, and forward compatibility. Targeting at stream-based functions such as finite impulse response (FIR) filtering, PipeRench consists of a sequence of stripes, which are equivalent to pipeline stages. However, one physical stripe can function as several pipeline stages in a



Figure 18 Block diagram of REMARC with a microprocessor.

time-sharing fashion. For example, a five-stage pipeline can be implemented on three stripes with slight reconfiguration overhead. As shown in Figure 19, each stripe has an interconnect network and some processing elements (PEs), which are composed of ALUs and registers. The ALUs are implemented using look-up tables and some extra logic for carry chains, zero detection, and so forth. In addition to the local interconnect network, there are also four global buses for forwarding data between stripes that are not next to each other. Evaluation of certain multimedia computing kernels shows a speedup factor of 11–190 over a 330-MHz UltraSPARC-II.

The Cheops imaging system is a stand-alone unit for acquisition, processing, and display of digital video sequences and model-based representations of moving scenes [67]. Instead of using a number of general-purpose microprocessors and DSPs to achieve the computation power for video applications, Cheops abstracts out a set of basic, computationally intensive stream operations required for real-time performance of a variety of applications and embodies them in a compact, modular platform. The Cheops system uses stream processors to handle video data like a data flow machine. It can support up to four processor modules. The block diagram of the overall architecture is depicted in Figure 20.



Figure 19 Stripe architecture.



Figure 20 Block diagram of the Cheops system.

Each processor module consists of eight memory units and eight stream processors which are connected together through a cross-point switch. The DMA controllers on each VRAM bank are capable of handling one- and two-dimensional arrays and they can pad or decimate depending on the direction of data flow. The Cheops is a rather complete system which even includes a multitasking operating system with a dynamic scheduler.

7.3 Commentary

Reconfigurable computing systems to some degree combine the speed of ASIC and the flexibility of software together. They emerge as a unique approach to high-performance video signal processing. Quite a few systems have been built to speed up video applications and they have proven to be more efficient than systems based on general-purpose microprocessors.

This approach also opens a new research area and raises many challenges for both hardware and software development. On the hardware side, how to couple reconfigurable components with microprocessors still remains open, and the granularity, speed, and portion of reconfiguration as well as routing structures are also subjects of active research. On the software side, CAD tools need great improvement to automate or accelerate the process of mapping applications to reconfigurable systems.

Although reconfigurable systems have shown the ability to speed up video signal processing as well as many other types of applications, they have not met the requirement of the marketplace; most of their applications are limited to individual research groups and institutions.

8 CONCLUSIONS

In this chapter, we have discussed four major approaches to digital video signal processing architectures: Instruction set extensions try to improve the performance of modern microprocessors; dedicated codecs seem to offer the most cost-effective solutions for some specific video applications such as MPEG-2 decoding; programmable VSPs tend to support various video applications efficiently; and reconfigurable computing compromises flexibility and performance at the system level. Because the four approaches are targeted at different markets, each having both advantages and disadvantages, they will continue to coexist in the future. However, as standards become more complex, programmability will be important for even highly application-specific architectures. The past several years have seen limited programmability become a commonplace in the design of application-specific video processors. As just one example, all the major MPEG-2 encoders incorporate at least a RISC core on-chip.

Efficient transferring of data among many processing elements is another key point for VSPs, whether dedicated or programmable. To qualify for realtime video processing, VSPs must be able to accept a high-bandwidth incoming bit stream, process the huge amount of data, and produce an output stream. Parallel processing also requires communication between different modules. Therefore, all of the VSPs we have discussed use either a very wide bus (e.g., in Chromatic Research Mpact2, the internal data bus is 792-bit wide) or a crossbar (e.g., Texas Instruments' TMS320C8x), or some other extremely fast interconnect mechanisms to facilitate high-speed data transfer. External bandwidth is usually achieved by using Rambus RDRAM or synchronous DRAM.

A few VSPs (e.g., Sony CXD1930Q, MicroUnity Cronus, Philips Tri-Media) are equipped with a real-time operating system or multitasking kernel to support different tasks in video applications. This brings video signal processing to an even more advanced stage. Usually in a multimedia system, many devices are involved. For example, in an MPEG-2 decoder, video and audio signals are separately handled by different processing units, and the coordination of different modules is very important. Multitasking will become an increasingly important capability as video processors are asked to handle a wider variety of tasks at multiple rates. However, the large amount of state in a video computation, whether it be in registers or main memory, creates a challenge for real-time operating systems. Ways must be found to efficiently switch contexts between tasks that use a large amount of data.

It is natural to ask which architecture will win in the long run: multimedia instruction set extensions, application-specific processors, programmable VSPs, or reconfigurable? It is safe to say that multimedia instruction set extensions for general-purpose CPUs are here to stay. These extensions cost very little silicon area to support, and now that they have been designed into architectures, they are unlikely to disappear. These extensions can significantly speed up video algorithms on general-purpose processors, but, so far, they do not provide the horsepower required to support the highest-end video applications; for example, although a workstation may be able to run MPEG-1 at this point in time, the same fabrication technology requires specialized processors for MPEG-2. We believe that the greatest impediment to video performance in general-purpose processors is the memory system. Innovation will be required to design a hierarchical memory system, which competes with VSPs yet is cost-effective and does not impede performance for traditional applications.

Application-specific processors are unlikely to disappear. There will continue to be high-volume applications in which it is worth the effort to design a specialized processor. However, as we have already mentioned, even many application-specific processors will be programmable to some extent because standards continue to become more complex.

Reconfigurable logic technology is rapidly improving, resulting in both higher clock rates and increased logic density. Reconfigurable logic should con-

tinue to improve for quite some time because it can be applied to many different applications, providing a large user base. As it improves, we can expect to see it used more frequently in video applications.

The wild card is the programmable VSP. It provides a higher performance than multimedia extensions for CPUs but is more flexible than applicationspecific processors. However, it is not clear what the "killer application" will be that drives VSPs into the marketplace. Given the cost of the VSP itself and of integrating it into a complex system like a PC, VSPs may not make it into wide use until some new application arrives which demands the performance and flexibility of VSPs. Home video editing, for example, might be one such application, if it catches on in a form that is sufficiently complex that the PC's main CPU cannot handle the workload. The next several years will see an interesting and, most likely intense battle between video architectures for their place in the market.

REFERENCES

- 1. JL Mitchell, WB Pennebaker, CE Fogg, DJ LeGall. MPEG Video Compression Standard. New York: Chapman & Hall, 1997.
- 2. Texas Instruments, TMS34010 graphics system processor data sheet, http://www-s.ti.com/sc/psheets/spvs002c/spvs002c.pdf
- Philips Semiconductors. Data sheet—SAA9051 digital multi-standard color decoder.
- Philips Semiconductors. Data sheet—SAA7151B digital multi-standard color decoder with SCART interface, http://www-us.semiconductors.philips.com/acrobat/ 2301.pdf
- K Aono, M Toyokura, T Araki. A 30ns (600 MOPS) image processor with a reconfigurable pipeline architecture. Proceedings, IEEE 1989 Custom Integrated Circuits Conference, IEEE, 1989, pp 24.4.1–24.4.
- T Fujii, T Sawabe, N Ohta, S Ono. Super high definition image processing on a parallel signal processing system. Visual Communications and Image Processing '91: Visual Communication, SPIE, 1991, pp 339–350.
- KA Vissers, G Essink, P van Gerwen. Programming and tools for a general-purpose video signal processor. Proceedings, International Workshop on High-Level Synthesis, 1992.
- T Inoue, J Goto, M Yamashina, K Suzuki, M Nomura, Y Koseki, T Kimura, T Atsumo, M Motomura, BS Shih, T Horiuchi, N Hamatake, K Kumagi, T Enomoto, H Yamada, M Takada. A 300 MHz 16b BiCMOS video signal processor. Proceedings, 1993 IEEE Int'l Solid State Circuits Conference, 1993, pp 36–37.
- 9. Intel Corp. i860 64-Bit Microprocessor, Data Sheet. Santa Clara, CA: Intel Corporation, 1989.
- 10. Superscalar techniques: superSparc vs. 88110, Microprocessor Rep 5(22), 1991.
- 11. R Lee, J Huck. 64-Bit and multimedia extensions in the PA-RISC 2.0 architecture. Proc. IEEE Compcon 25–28, February 1996.

- 12. R Lee. Subword parallelism with MAX2. IEEE Micro 16(4):51-59, 1996.
- R Lee, L McMahan. Mapping of application software to the multimedia instructions of general-purpose microprocessors. Proc. SPIE Multimedia Hardware Architect 122–133, February 1997.
- 14. L Gwennap. Intel's MMX speeds multimedia. Microprocessor Rep 10(3), 1996.
- L Gwennap. UltraSparc adds multimedia instructions. Microprocessor Rep 8(16): 16–18, 1994.
- 16. Sun Microsystems, Inc. The visual instruction set, Technology white paper 95-022, http://www.sun.com/microelectronics/whitepapers/wp95-022/index.html
- 17. P Rubinfeld, R Rose, M McCallig. Motion Video Instruction Extensions for Alpha, White Paper. Hudson, MA: Digital Equipment Corporation, 1996.
- 18. MIPS Technologies, Inc. MIPS extension for digital media with 3D, at http://www.mips.com/Documentation/isa5_tech_brf.pdf, 1997.
- T Komarek, P Pirsch. Array architectures for block-matching algorithms. IEEE Trans. Circuits Syst 36(10):1301–1308, 1989.
- M Yamashina et al. A microprogrammable real-time video signal processor (VSP) for motion compensation. IEEE J Solid-State Circuits 23(4):907–914, 1988.
- H Fujiwara et al. An all-ASIC implementation of a low bit-rate video codec. IEEE Trans. Circuits Sys Video Technol 2(2):123–133, 1992.
- 22. http://www.8x8.com/docs/chips/lvp.html
- 23. http://products.analog.com/products/info.asp?product=ADV601
- 24. http://www.c-cube.com/products/products.html
- 25. T Sikora. MPEG Digital Video Coding Standards. In: R Jurgens. Digital Electronics Consumer Handbook. New York: McGraw-Hill, 1997.
- ESS Technology, Inc. ES3308 MPEG2 audio/video decoder product brief, http:// www.esstech.com/product/Video/pb3308b.pdf
- 27. http://www.chips.ibm.com/products/mpeg/briefs.html
- W. Bruls, et al. A single-chip MPEG2 encoder for consumer video storage applications. Proc. IEEE Int. Conf. on Consumer Electronics, 1997, pp 262–263.
- Philips Semiconductors. Data sheet—SAA7201 Integrated MPEG2 AVG decoder, http://www-us.semiconductors.philips.com/acrobat/2019.pdf
- 30. http://www-us.semiconductors.philips.com/news/archive.stm
- P. Lippens, et al. Phideo: A silicon compiler for high speed algorithms. European Design Automation Conference, 1991.
- 32. Sony Semiconductor Company of America. CXD1922Q MPEG-2 technology white paper, http://www.sel.sony.com/semi/CXD1922Qwp.html
- Sony Semiconductor Company of America. Press releases—virtuoso IC family, http://www.sel.sony.com/semi/nrVirtuoso.html
- 34. http://www.dvimpact.com/products/single-chipn.html
- 35. http://www.lsilogic.com/products/ff0013.html
- 36. http://eweb.mei.co.jp/product/mvd-lsi/me-e.html
- T Araki et al. Video DSP architecture for MPEG2 codec. Proc. IEEE ICASSP 2: 417–420, April 1994.
- 38. http://www.mitsubishi.com/ghp_japan/TechShowcase/Text/tsText08.html
- 39. http://www.visiontech-dml.com/product/index.htm
- 40. http://www.mpact.com/

- 41. S Purcell. Mpact2 media processor, balanced 2X performance. Proc. SPIE Multimedia Hardware Architectures, 1997, pp 102–108.
- 42. C Hansen. MicroUnity's media processor architecture. IEEE Micro 16(4):34-41, 1996.
- 43. http://www.microunity.com/www/mediaprc.htm
- R Hayes, et al. MicroUnity Software Development Environment. Proc. IEEE Compcon 341–348, February 1996.
- 45. E Holmann, et al. A media processor for multimedia signal processing applications. IEEE Workshop on Signal Processing Systems, 1997, pp 86–96.
- 46. T Yoshida, et al. A 2V 250MHz multimedia processor. Proc. ISSCC, 266–267:471, February 1997.
- 47. K Suzuki, T Arai, K Nadehara, I Kuroda. V830R/AV: Embedded multimedia superscalar RISC processor. IEEE Micro 18(2):36–47, 1998.
- 48. http://www.trimedia.philips.com/
- JTJ van Eijndhoven, FW Sijstermans, KA Vissters, EJD Pol, MJA Tromp, P Struik, RHJ Bloks, P van der Wolf, AD Pimentel, HPE Vranken. TriMedia CPU64 architecture. In: Proceedings, ICCD '99. Los Alamitos, CA: IEEE Computer Society Press, 1999, pp 586–592.
- 50. L Nguyen, et al. Establish MSP as the standard for media processing. Proc. Hot Chips 8: A Symposium on High Performance Chips, 1996.
- 51. http://www.ti.com/sc/docs/dsps/products/c8x/index.htm
- 52. http://www.ti.com/sc/docs/dsps/products/c6x/index.htm
- Z Wu, W Wolf. Parallelism analysis of memory system in single-chip VLIW video signal processors. Proc. SPIE Multimedia Hardware Architectures, 1998, pp 58– 66.
- 54. P Faraboschi, G Desoli, JA Fisher. The latest word in digital and media processing. IEEE Signal Process Mag 15(2):59–85, 1998.
- 55. Xilinx Corporation, http://www.xilinx.com/
- 56. S Hauck. The roles of FPGAs in reprogrammable systems. Proc IEEE 615–638, April 1998.
- 57. K Compton, S Hauck. Configurable computing: A survey of systems and software. Technical Report. Northwestern University, 1999.
- 58. J Arnold, D Buell, E Davis. Splash II. Proc. 4th ACM Symposium of Parallel Algorithms and Architectures, 1992, pp 316–322.
- PM Athanas, AL Abbott. Real-time image processing on a custom computing platform. IEEE Computer 28(2), 1995.
- 60. N Ratha, A Jain, D Rover. Convolution on Splash 2. Proc. IEEE Symposium on FPGAs for Custom Computing Machines, 1995, pp 204–213.
- 61. M Rencher, BL Hutchings. Automated target recognition on Splash II. Proc. 5th IEEE Symposium on FPGAs for Custom Computing Machines, 1997, pp 192–200.
- J Vuillemin, P Bertin, D Roncin, M Shand, H Touati, P Boucard. Programmable active memories: Reconfigurable systems come of age. IEEE Trans. VLSI Syst 4(1): 56–69, 1996.
- 63. MJ Wirthlin, BL Hutchings. A dynamic instruction set computer. IEEE Workshop on FPGAs for Custom Computing Machines, 1995, pp 99–107.
- 64. JR Hauser, J Wawrzynek. Garp: A MIPS processor with a reconfigurable coproces-

sor. IEEE Workshop on FPGAs for Custom Computing Machines, 1997, pp 24-33.

- T Miyamori, K Olukotun. A quantitative analysis of reconfigurable coprocessors for multimedia applications. Proc. IEEE International Symposium on FPGAs for Custom Computing Machines, 1998, pp 2–11.
- SC Goldstein, H Schmit, M Moe, M Budiu, S Cadambi. PipeRench: A coprocessor for streaming multimedia acceleration. Proc. International Symposium on Computer Architecture, 1999, pp 28–39.
- 67. VM Bove Jr, JA Watlington. Cheops: A reconfigurable data-flow system for video processing. IEEE Trans. Circuits Syst Video Technol 5:140–149, April 1995.