SOFTWARE MPEG-2 VIDEO DECODER ON A 200-MHz, LOW-POWER MULTIMEDIA MICROPROCESSOR

Kouhei Nadehara, Hanno Lieske[†] and Ichiro Kuroda

C&C Media Res. Labs., NEC Corp. 4-1-1, Miyazaki, Miyamae-ku, Kawasaki 216, Japan {nade,kuroda}@ccm.CL.NEC.co.jp [†]University of Hannover Schneiderberg 32, D-30167 Hannover, Germany lieske@mst.uni-hannover.de

ABSTRACT

This paper presents a low-power, 32-bit RISC microprocessor with a 64-bit "single-instruction multiple-data" multimedia coprocessor, V830R/AV, and its MPEG-2 video decoding performance. This coprocessor basically performs multimedia-oriented four 16-bit operations every clock, such as multiply-accumulate with symmetric rounding and saturation, and accelerates computationally intensive procedures of the video decoding; an 8×8 IDCT is performed in 201 clocks. The processor employs the Concurrent Rambus DRAM interface, and facilities for controlling cache behaviors explicitly by software to speed up enormous memory accesses necessary to motion compensation. The 200-MHz V830R/AV processor with the 600-Mbyte/sec. Concurrent Rambus DRAMs decodes MPEG-2 MP@ML video in realtime (30 frames/sec.).

1. INTRODUCTION

Multimedia signal processing, such as compression and decompression of voice, audio and video, is indispensable in consumer electronic products such as video games, digital video disc players and set-top boxes, as well as personal computers and workstations. Multimedia signal processing is so demanding that it is necessary to incorporate application specific ICs or digital signal processors (DSPs) in addition to a main general-purpose processor.

In progress of processor technology, there is a strong requirement to implement multimedia equipment using software on a general-purpose processor. These systems could be easily compliant to multiple standards by software upgrades. Moreover, it is quite advantageous in system size, power and cost to implement systems' multimedia capability in software without additional signal processing hardware.

Some high-end processors for personal computers and workstations have introduced multimedia-oriented execution units and associated instructions [1, 2]. These processors have already attained sufficient signal processing performance to decode MPEG-2 main profile at main level (MP@ML) bitstreams in real-time [3, 4]. It was not the case, however, in inexpensive consumer electronic products, because low-cost, embedded processors have not introduced multimedia-oriented facilities so aggressively.

For software signal processing in low-cost, low-power systems, a low-power, embedded RISC processor with a 64-bit multimedia coprocessor, named V830R/AV, has been developed. This processor provides high signal processing performance up to 1.6 GOPS by "single-instruction multiple-data (SIMD)-type" parallel operations and a fast 600-Mbyte/sec. Concurrent Rambus DRAM interface.

In this paper, the V830R/AV processor architecture is described first. Next, the processor's signal processing performance is evaluated taking an MPEG-2 video decoding as an example.

2. PROCESSOR ARCHITECTURE

The V830R/AV processor is a new member in the NEC's V800 embedded RISC family [5]. This processor is designed to support real-time signal processing of broadcastquality video. It integrates a 64-bit multimedia coprocessor, the Concurrent Rambus interface, and 16-Kbyte, 4-way, set-associative instruction and data caches, with the V830compatible integer execution pipeline (Figure 1). The integer execution pipeline integrates a 1-clock throughput 32-bit integer/fixed-point multiply-accumulator for highprecision signal processing such as audio encoding and decoding [6].

The processor can issue two instructions simultaneously when the current and the next instructions can be issued to multimedia and integer execution units, respectively. This asymmetric two-way superscalar capability makes maximum use of both execution units, while simplifying an instruction decoder.

The V830R/AV processor is fabricated in a 0.25μ m, 4level metal layer CMOS technology, resulting in 3.9 million



Figure 1: Processor Core Block Diagram.

transistors. The processor core clock frequency is 200 MHz at 2.5V power supply. It dissipates less than 2.0 W.

2.1. Multimedia Coprocessor

The multimedia coprocessor performs SIMD parallel operations on eight 8-bit, four 16-bit and two 32-bit packed data in thirty-two 64-bit multimedia registers. This large multimedia register file eliminates instructions for saving or restoring intermediate results. The coprocessor mainly supports four 16-bit data type, which is sufficient precision for video applications. Table 1 shows the multimedia instruction set supported by the coprocessor, called MIX2 (Multimedia Instruction eXtension 2). The multimedia execution unit is fully pipelined, and has 1-clock throughput and fixed 4-clock latency for simplicity in both the chip design and software programming.

2.2. Cache Management

In embedded processors, which cannot afford large multilevel caches, cache misses impose heavy penalties on their performance. To make matters worse, caching strategy is not very effective on multimedia data, which usually have very large working set and less temporal locality. Therefore, this processor incorporates mechanisms to control cache behaviors explicitly by software according to memory access

Instruction	$8-bit \times 8$	$16-bit \times 4$	$32-bit \times 2$	$64-bit \times 1$
Arithmetic				
Vector Addition	\cap	\cap	\cap	
Vector Subtract	ŏ	ŏ	ŏ	
Vector Multiply	Ŭ	ŏ	0	
Vector Multiply Accumulate		Ō		
Scalar Addition		0		
Scalar Subtract		0		
Scalar Multiply		0		
Scalar Multiply Accumulate		0		
Compare				
Vector Maximum		0		
Vector Minimum		0		
Scalar Maximum		Q		
Scalar Minimum		0		
Format Conversion	~			
Pack	Q	Q	~	
Interleave	0	0	0	
Logical				
And				Q
Or V				Q
Xor				l Q
NOI Shift				0
Janiard Lafe	1	\cap	\cap	\cap
Logical Picht				
Arithmatic Right				
Motion Estimation				
Absolute Difference		\cap		
Shift and Add			0	

characteristics of multimedia applications, in addition to general automatic mechanisms to reduce miss penalties.

For efficient software execution, it is necessary to reduce both cache miss counts and miss penalties per cache miss. First, to reduce cache miss counts, the instruction and data caches have a "freeze" attribute bit in each 64-byte cache line to suppress cache line replacements. When this bit is set, the current data reside in the cache line; the data stored in a "frozen" line is accessible without a cache miss.

Second, to reduce apparent penalties per cache miss, this processor employs a non-blocking data cache which allows one pending miss. The processor deals with a data cache miss in parallel to instruction execution, when there is a enough time between a load instruction which caused the miss and an instruction which refers to the load result. Third, the processor has a "preload" instruction to fill the specified data to the data cache prior to load instructions. This instruction further increases the possibility to handle cache misses in background and prevents the processor pipeline from stalling.

2.3. External Memory

The processor employs a fast memory interface called Concurrent Rambus [7, 8]. The Concurrent Rambus interface



Figure 2: MPEG-2 Video Decoding Process.

has data bandwidth up to 600 Mbyte/sec., because it transfers access commands and data through the 8-bit bus on both edges of fast 300-MHz clock. This interface can provide twice the bandwidth than the synchronous DRAM interface with a 32-bit, 66-MHz bus, with much less pin count. This fast memory interface also reduces cache miss penalties per miss.

3. SOFTWARE MPEG-2 VIDEO DECODING

An MPEG-2 video software decoder for the V830R/AV processor has been developed, based on the sample decoder implementation, *mpeg2decode*, from MPEG Software Simulation Group [9]. The original decoder compiled by a C compiler without MIX2 instructions, takes 891 M instructions/ 1.5 G clocks to decode 30 frames of a 4 Mbps MPEG-2 MP@ML bitstream; it is 7.5 times the processor performance. Therefore the decoder has been optimized by rewriting the macroblock layer and subsidiary functions in the MIX2 assembler.

The MPEG-2 video decoding process mainly comprises of four procedures as shown in Figure 2; variable length decoding (VLD), inverse quantization (IQ), inverse discrete cosine transform (IDCT), and motion compensation (MC).

3.1. VLD and IQ

A variable-length codeword is extracted from the current bit position of an MPEG-2 video bitstream in VLD. Theoretically, variable-length codes are decoded one by one, because it is impossible to determine each code length in advance.

A *doubleword shift* instruction in the integer execution unit can efficiently extract a 32-bit word containing a current variable-length cordword even when the codeword is stored across two bitstream buffer entries (Figure 3). IQ is performed every time the codeword is decoded.



Figure 3: VLD Using Doubleword Shift Instruction.

The software decoder "freezes" data cache lines correspond to variable-length code tables to avoid cache misses during code table lookups. For the effective use of the 16-Kbyte data cache, small multi-level code tables are employed instead of a large single-level code table. Each element is packed into bit fields for further table compaction.

3.2. IDCT

Since IDCT is a multiplication intensive procedure with high data parallelism, this procedure is very appropriate to the SIMD coprocessor. SIMD multiply-accumulate instructions are very suitable for fast IDCT. An 8×8 2D IDCT is first decomposed into two consecutive 1D 8-point IDCT, and four 1D 8-point IDCTs are performed in parallel by SIMD multiply-accumulate instructions with symmetric rounding. This instruction contributes to simple and fast implementation of an IDCT compliant to the IEEE1180 standard [10].

Thirty-two 64-bit registers in the coprocessor are large enough to hold all the 8×8 elements during an IDCT. They eliminate instructions for saving and restoring intermidiate results. As a result, an 8×8 2D IDCT is performed in 272 instructions/201 clocks at the V830R/AV processor, which is 6.8 times faster than the original.

3.3. MC

A macroblock is reconstructed by adding difference signals, which are the IDCT outputs, and reference block(s) pointed by motion vector(s). There are heavy cache miss penalties in MC, because of extensive read and write accesses to external frame buffers in main memory.

Figure 4 shows how the preload instruction is used to mitigate cache miss penalties in reading reference macroblocks. There is not sufficient time to preload a current reference macroblock shown in solid lines between a macroblock's motion vector decode and the reference macroblock access, since motion vectors and difference signals are successive in a bitstream.

Therefore, preload instructions are issued for the next MC. Assuming that motion vectors are similar in adjacent macroblocks, the center of the next reference macroblock



Figure 4: Preloading Reference Macroblocks.

area shown in dashed lines is considered to be a prospective candidate for preloading. Therefore, preload instructions are issued to the hatched area.

The preload instruction is also effective to allocate cache lines corresponding to a current block in a reconstructed frame to reduce write cache miss penalties. These explicit cache control greatly reduces a clock count necessary in MC to 14% compared to the original version.

3.4. Implementation Results

Figure 5 shows clock and instruction counts necessary to decode a 30 frame, 4 Mbps, MPEG-2 MP@ML video sequence for the original and optimized decoders. After optimizing software decoder using MIX2 instruction set and explicit cache control. clock and instruction counts are reduced to 196.7 M and 168.2 M, respectively. This result shows the 200-MHz V830R/AV has enough performance to decode MPEG-2 MP@ML video (720x480 pixels, 30 frames/sec.) in real-time.

4. CONCLUSION

The software MPEG-2 video decoder implemented on the low-power, low-cost, RISC microprocessor with a 64-bit multimedia coprocessor, V830R/AV, is presented. This processor's multimedia instructions, which perform parallel operations on multimedia registers, reduce the dynamic instruction count for decoding 30 frames of MPEG-2 video from 891 million to 168.2 million. In addition, cache line freezing and cache preloading mechanisms reduce the clock count to 196.7 million. As a result, the 200-MHz V830R/AV processor with the Concurrent Rambus DRAMs achieves software MPEG-2 MP@ML video decoding in real-time. It shows that this processor can provide sufficient signal processing performance to deal with broadcast-quality video in low-power, low-cost, consumer products.



Figure 5: Software Decoder Performance.

5. ACKNOWLEDGMENTS

The authors would like to thank Dr. Kazunori Ozawa, Dr. Takao Nishitani, and Mr. Takashi Miyazaki of the NEC R&D group for their continous support.

6. REFERENCES

- Marc Tremblay et al., "VIS Speeds New Media Processing," IEEE Micro, pp. 10–20, Aug. 1996.
- [2] Linley Gwennap, "Intel's MMX Speeds Multimedia," Microprocessor Report, Vol. 10, No. 3, pp. 1, 6–10, Micro Design Resources, Mar. 5, 1996.
- [3] ISO/IEC 13818-2, "Information Technology Generic Coding of Moving Pictures and Associated Audio," 1995.
- [4] Masao Ikekawa et al., "A Real-Time Software MPEG-2 Decoder for Multimedia PCs," Proc. of Int'l Conf. on Consumer Electronics, pp. 2–3, Jun. 1997.
- [5] Tomohisa Arai et al., "Embedded Multimedia Superscalar RISC Processor with Rambus Interface," IEEE Hot Chips Symposium IX, Aug. 1997.
- [6] Kouhei Nadehara et al., "Low-Power Multimedia RISC," IEEE Micro, Vol. 15, No. 6, pp.20–29, Dec. 1995.
- [7] Masaki Kumanoya et al., "Advances in DRAM interfaces," IEEE Micro, Vol. 15, No. 6, pp.30–36, Dec. 1995.
- [8] Allen Roberts, "Direct Rambus: The New Memory Standard," Conference Materials of Microprocessor Forum, Oct. 1997, also seen at Rambus WWW page: (URL:http://www.rambus.com/ html/mp_forum.html).
- [9] MPEG Software Simulation Group WWW page: (URL:http://www.mpeg.org/MSSG/).
- [10] "IEEE Standard Specifications for the Implementations of 8×8 Inverse Discreate Cosine Transform," Std 1180-1190, Dec. 1990.