MAJC-5200: A High Performance Microprocessor for Multimedia Computing

Subramania Sudharsanan

Sun Microsystems, Inc., Palo Alto, CA 94303, USA

Abstract. The newly introduced Microprocessor Architecture for Java Computing (MAJC) supports parallelism in a hierarchy of levels: multiprocessors on chip,vertical micro threading, instruction level parallelism via a very long instruction word architecture (VLIW) and SIMD. The first implementation, MAJC-5200, includes some key features of MAJC to realize a high performance multimedia processor. Two CPUs running at 500 MHz are integrated into the chip to provide 6.16 GFLOPS and 12.33 GOPS with high speed interfaces providing a peak input-output (I/O) data rate of more than 4.8 G Bytes/second. The chip is suitable for a number of applications including graphics/multimedia processing for high-end set-top boxes, digital voice processing for telecommunications, and advanced imaging.

1 Introduction

MAJC-5200 is a high performance general purpose microprocessor (based on MAJC [1]) exceptionally suitable for multimedia computing. The processor targets networked and communication devices, client platforms, and application servers delivering digital content and Java applications. It is a multiprocessor system on a chip integrating two CPUs, a memory controller, a PCI controller, two high bandwidth I/O controllers, a data transfer engine, and a crossbar interfacing all the blocks. This article describes the MAJC-5200 microprocessor and points out its capabilities in multimedia computing. The descriptions start with an overview of the MAJC architecture in Section 2. Section 3 details the MAJC-5200 microprocessor. The CPU, its pipeline, various functional blocks, memory subsystem, and external interfaces are described. Instruction set architecture as implemented in MAJC-5200 is given in Section 4. In Section 5, we list benchmark numbers for some multimedia and signal processing applications. Concluding remarks are in Section 6.

2 Architecture

The design of MAJC has been primarily towards exploiting the advances in both hardware and software technologies to address the new computational challenges. These challenges come from the increase in communication bandwidth and the techniques to communicate and process the data that is increasingly of auditory and visual type. The advances in the semiconductor technology boost the number of transistors in a chip and the frequency they can be switched at. Software technology improvements in compilers, Java, and multi-threaded applications make it possible for high level language development of multimedia and telecommunication infrastructure applications to match the *internet-speed* time-to-market requirements. MAJC is scalable, exploits parallelism at a hierarchy of levels and modular for ease of implementation. At the highest level of parallelism, MAJC provides inherent support for multiple processors on a chip. The next level is the ability of *vertical micro-threading* which is attained through hardware support for rapid, low overhead context switching. The context switches can be triggered through either due to a long latency memory fetch or other events. The next hierarchy of parallelism comes from an improved very long instruction word (VLIW) architecture. The instruction packets can vary in length, up to 128 bits, with a maximum of four instructions each of 32 bits per packet. The lowest level of parallelism comes from single instruction multiple data (SIMD) or sub-word parallelism.

3 MAJC-5200 Microprocessor

3.1 Building Blocks

The first implementation of MAJC, MAJC-5200, is a 500 MHz, dual-CPU multimedia processor with a high I/O bandwidth. It implements several key parallelism features of MAJC. The two CPUs share a coherent four-way set-associative 16-KB data cache and common external interfaces. Each of these CPU is a four-issue MAJC VLIW engine. Each CPU contains its own two-way set-associative instruction cache of 16 KB. A high throughput bandwidth requirement is addressed by a multitude of interfaces with built-in controllers. The main memory is a direct Rambus DRAM (DRDRAM) with an interface supporting a peak transfer rate of 1.6 GB/s. A direct interface to 32-bit/66 MHz PCI provides DMA and programmed I/O (PIO) capabilities to transfer up to 264 MB/s. There are two other interfaces that support up to 4.0 GB/s that could be used for high speed parallel (64-bits at 250 MHz) interfaces: North and South UPA. (Universal Port Architecture or UPA has been an interface for graphics and multi-processor configurations of UltraSparc- based systems [3].) The NUPA block contains a 4 KB input FIFO buffer that can also be accessed by both CPUs. The other specialized block in the chip is a graphics preprocessor (GPP). The GPP has built-in support for real-time 3D geometry decompressing, data parsing, and load balancing between the two processors. An on-chip Data Transfer Engine (DTE) provides DMA capabilities amongst these various memory and i/o devices, with the bus interface unit acting as a central crossbar. The chip block diagram is depicted in Figure 1.

3.2 MAJC CPU

The core of each CPU as depicted in Fig. 2 has four functional units, FU0 through FU3. The VLIW instruction packet can have one, two, three, or four

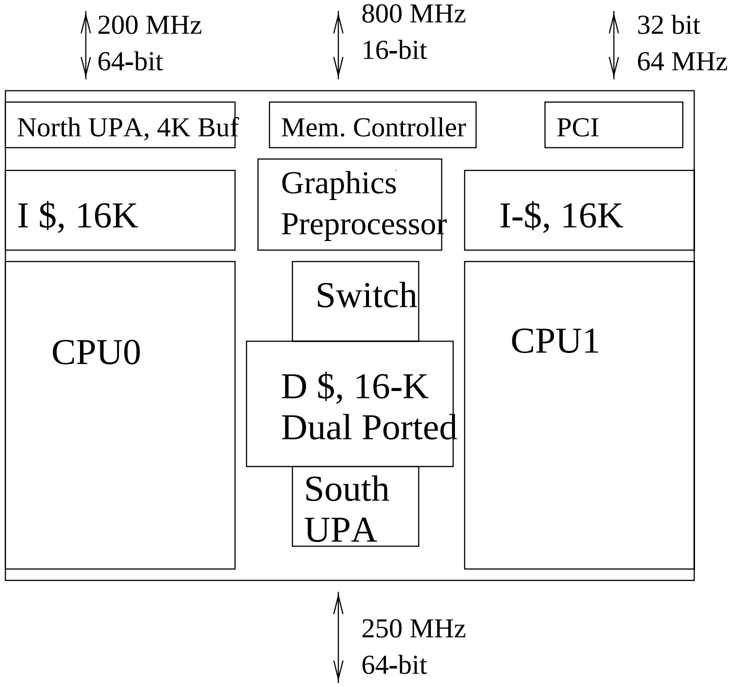

Fig. 1. MAJC-5200 Block Diagram

instructions in it. A two-bit header indicates the issue width, reducing unnecessary `nops` in the instruction stream. The first instruction in a packet must be an FU0 instruction which can be a memory operation (loads and stores), control flow operation (branch, jump and link, etc.) or an ALU (shift, add, etc.) type. FU0 interfaces with the LSU to perform the memory operations. Some special compute instructions are also executed in the FU0. The instructions for FU1-3 are of compute types.

Processor Pipeline During the fetch stage, a 32-byte aligned data is brought from the instruction cache unit. The next stage aligns an instruction packet of 16 bytes based on the header bits. The aligned instructions are then placed in an instruction buffer to feed the decode stage. The branch prediction information bits are looked at this stage to prepare for both static and dynamic predictions. The instruction decode and operand read from the register file occur in the next stage. Instructions then enter the execution pipelines in the functional units. Since the instruction scheduling is a compiler driven task in a VLIW machine, only the non-deterministic loads and long latency instructions are interlocked through a score-boarding mechanism. All the other instructions have a deterministic delay. MAJC-5200 provides precise exception handling capabilities for most instructions.

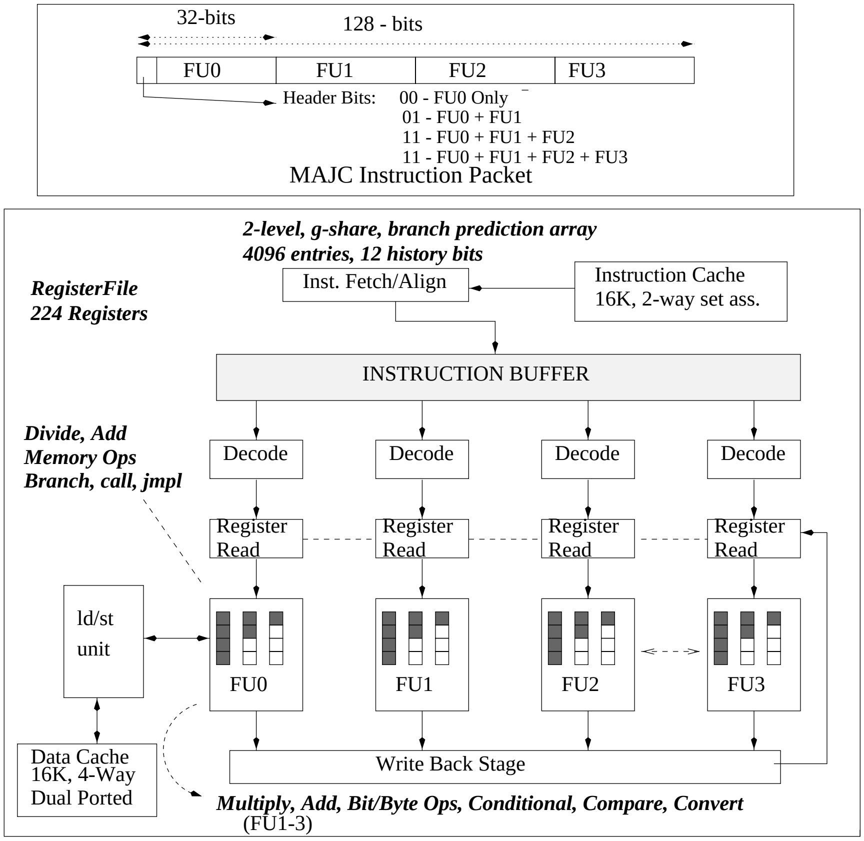

Fig. 2. MAJC CPU

The results are committed to the register file via the write back (WB) stage. In general, computational results become visible to other functional units from Trap/WB stage onwards, with some optimizations for results of FU0 and FU1. Majority of the instructions take a single cycle, with some taking two and four cycles. Within a functional unit, the results are bypassed to younger instructions as soon as available. The results of FU0 are forwarded to FU2 and FU3 with one cycle delay and are visible to FU1 in the next cycle. Similarly, the results of FU1 are forwarded to FU0 without any delay. This complete bypass between FU0 and FU1 enables a simple two-scalar performance for programs that exhibit lower levels of ILP.

Each CPU has a total of 224 logical registers partitioned into 96 global, accessible by all units, and 128 local, 32 for each FU, local registers. The Load/Store unit (LSU) on each CPU handles all memory operations to and from the registers. The LSU aggressively implements a non-blocking memory subsystem to allow multiple outstanding loads and stores. It provides buffering for up to five loads and eight stores. It allows a maximum of four cache misses without blocking the execution and handles out- of-order data returns. Non-faulting prefetch instructions that prefetch a block of data (32-byte) from the main memory to the data cache are also queued in LSU. Support for memory barrier and *atomic* instructions for synchronization of threads is also part of the LSU unit. The data cache access is brisk, allowing a two cycle load-to-use performance. Coupled with the synchronization instructions, this shared data cache provides a powerful, very low overhead communication between the two CPUs thus allowing a wide variety of compiler and run-time optimizations.

4 Instruction Set

MAJC-5200 provides support for 64-, 32-, 16-, and 8-bit integers in addition to 32- and 16-bit fixed point numbers. Both single (32-bit) and double (64-bit) IEEE 754-1985 floating point numbers are supported as well. A variety of load/store operations supporting byte, short, word, long, and group (1,2,4,8, and 32 bytes) can be issued as cached, non-cached, or non-allocating ones. The prefetch instruction is useful in programs with predictable data access patterns common in multimedia and image processing/graphics and other signal processing applications. MAJC-5200 implements branch (on conditions), call, and jump-and-link as FU0 instructions for program flow control. Predication instructions that conditionally pick (FU1-3), move (all FUs), and store (FU0) facilitate compiler to generate codes with fewer conditional branches.

Standard logical operations, AND, OR, XOR; logical and arithmetic shifts; and arithmetic operations, ADD and SUB, on 32-bit integers can be executed on all the functional units. All the units are capable of setting arbitrary constants. FU1-3 additionally provide saturated addition and subtraction of 32-bit integers. The FU1-3s further provide two cycle 32-bit integer multiplication and fused multiply-add (or multiply-sub) instructions that can be fully pipelined. Instructions that produce the high 32-bit part of a 64-bit product of two 32-bit integers facilitate obtaining 64-bit multiplies. Integer (32-bit) divide instruction is non-pipelined and available as an FU0 instruction.

SIMD versions of the arithmetic and logical operations available on FU1-3 take just one cycle. These operate on 16-bit short integer pairs or S.15 or S2.13 fixed point formats (Sign.integer.fraction). Furthermore, four different saturation modes can be enabled to automatically saturate the results. Fully pipelined SIMD counterpart of the multiply-add fused operation again has both regular and saturated versions operating on the same data types. In addition a dot product instruction that preserves the full 32-bits of precision multiplies and accumulates two pairs of 16-bit values. Saturated S.31 product of two S.15 quantities can also be obtained. Another set of unique FU0 SIMD instructions are six-cycle parallel divide and reciprocal square-root for S2.13 fixed point format data. These instructions form a basis for very powerful DSP, media, and graphics capabilities: multiply-add fused and dot product instructions aiding the filtering and transforming operations; and the FU0 SIMD helping the graphics lighting routines. At 500 MHz clock frequency, the peak performance becomes more than 12.33 GOPs for the 16-bit quantities.

The single precision floating point instructions are equally powerful. Pipelined addition, subtraction, multiply and fused multiply-add are available in the FU1-3s with just four cycle latency. Divide and reciprocal square-root instructions implemented in the FU0 have just six cycle latency. These make the processor capable of 6.16 GFLOPS at 500 MHz. The result is a very powerful single precision floating point FFT and graphics transform routines. Functional units FU1-3 provide double precision floating point addition, subtraction, and multiply operations. These instructions are partially pipelined for optimal performance and simpler scheduling by the compiler. Other floating point (single and double precision) instructions include minimum and maximum of two numbers; and a negate operation.

The bit manipulation instructions in the FU1-3s (bit field extract and leading zero detect) are quite useful in parsing compressed bit streams and in handling data communications. The bit extract is also a general purpose alignment instruction since the field extracted can span two registers. The byte-shuffle instruction in the FU1-3s provides a very versatile permutation function that can be used for alignment, table look-up, and zeroing byte-fields in a register. The pixel distance instruction computes the L_1-norm distance between two vectors of four packed bytes and accumulates them to a register. This instruction, combined with byte-shuffle provides excellent motion estimation performance for video coding applications.

5 Performance in Multimedia Applications

MAJC-5200 is slated to become the primary microprocessor in a variety of products that include high-end graphics systems, telecommunication infrastructure and document processing. For 3D graphics processing, MAJC-5200 has two features that significantly enhance its performance: the graphics pre-processor (GPP) and the two CPUs. The GPP decompresses compressed polygon information and distributes the uncompressed information to the CPUs using a load balancing mechanism. The geometry transformation and lighting are then performed using the CPUs. This pipelined architecture delivers a performance of between 60 and 90 million triangles per second. With such performance, MAJC-5200 can be used in a variety of graphics system architecture including multi-chip configurations to realize high-end graphic systems [4]. All the performance numbers provided are estimated using instruction accurate and cycle accurate simulators.

The performance in video and image processing applications is mainly attained using the instruction set that is suitable for several key kernels. Table 1 lists a few video signal processing benchmarks that indicate the performance of MAJC using a single CPU. Of course, in several of the applications it is possible to obtain thread level parallelism to effectively use both the CPUs. The versatile bit and byte manipulation operations help the variable length decoding common for image and video decompression. Combining this with static ILP and software pipelining, one can decode a variable length symbol and perform inverse zip-zag transform and inverse quantization [5] within 18 cycles. Motion estimation for a video encoder is significantly sped up via the byte permutation and pixel distance operations. Using a logarithmic search mechanism [5], a motion vector with a ±16 range can be found within about 3000 cycles. Large register file aids in convolution operations since the filter coefficients, image data, and the intermediate values can be easily stored in registers avoiding memory operations even when software pipelining techniques that typically require many registers are employed.

Table 1. Video/Image Processing Benchmarks (From Simulators)

Benchmark Kernel	Per Single MAJC CPU
8x8 IDCT	304 cycles
8x8 DCT + Quantization	200 cycles
MPEG-2 VLD+IZZ+IQ	27 MSymbols/sec
Motion Est./ ±16 MV range	3000 cycles
5x5 Convolution (512x512)	1.65 Mcycles
512x512 Color Conversion	0.9 MCycles

MAJC performance for one dimensional signal processing benchmarks also stands out as given in Table 2. The performance primarily stems from fused multiply-add instructions and corresponding memory bandwidth to actively keep the functional units computing. The benchmark kernels are commonly used ones in comparing digital signal processors and are heavily used in communication and speech processing applications. One interesting point to note is that unlike traditional DSPs that have smaller register files, MAJC-5200 is capable of using the compute efficient Radix-4 FFT algorithms. Bit reversal for FFT is however required to be performed using table look-up since no bit-reversed addressing is available. All the given benchmarks are for floating point numbers.

Using similar kernels, several applications have been developed. A small set is given here in Table 3. Again, these performances are estimated using simulators for a single CPU of the MAJC-5200 microprocessor running at 500 MHz.

6 Conclusion

MAJC-5200, the first implementation of the MAJC architecture, has been briefly described. Covered are the major architectural features and the the multimedia capabilities of the microprocessor. At 500 MHz, the processor delivers more than

Table 2. Signal Processing Benchmarks (From Simulators)

Benchmark Kernel	Per Single MAJC CPU
Cascade of eight 2nd order Biquads	63 cycles
64-sample, 64-tap FIR	2757 cycles
64-sample, 16th order IIR	2021 cycles
64-sample, 64-tap Complex FIR	8643 cycles
Single Sample, 16th order LMS	64 cycles
Max Search, maximum value in array of 40	126 cycles
Radix-2, 1024-point complex FFT	25196 cycles
Radix-4, 1024-point complex FFT	16996 cycles
Bit reversal, 1024-point	2484 cycles

Table 3. Application Performance (From Simulators)

Application	Single MAJC-5200 CPU Utilization
G.723.1 (encode) - float	1.6 % (1% without memory effects)
G.729.A (encode) - float	2.0 % (1% without memory effects)
MPEG-2 Video Decode (5Mbps, MP@ML)	75% (43 % without memory effects)
AC-3, MP2 Audio Decode	3-5 %
JPEG Baseline Encode	40 MB/s
Proprietary Lossless Coding	40 MB/s
H.263 Codec (128 kbps, 15 fps, CIF)	50 %

6 GFLOPS and 12 GOPS of raw performance. The performance of MAJC-5200 in graphics, image/video processing, and signal processing has been shown to show the suitability of it in a number of high performance applications.

References

1. M.Tremblay “A microprocessor architecture for the new millennium,” Hot Chips 11, Palo Alto, CA, August 1999.
2. L.Kohn, *et al* “Visual Instruction Set (VIS) in UltraSparc,” *Proceedings of Compcon*, IEEE CS Press, pp. 462-469,1995.
3. Sun Microsystems, Inc. http://www.sun.com/microelectronics/UltraSPARC/
4. M. Tremblay, “A VLIW convergent multiprocessor system on a chip,” Microprocessor Forum, San Jose, CA, Oct. 1999.
5. V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architecture*, Second Edition, Kluwer, Boston, 1997.

This article was processed using the LaTeX macro package with LLNCS style