Programmable VLIW and SIMD Architectures for DSP and Multimedia Applications

Deepu Talla
Laboratory for Computer Architecture
Department of Electrical and Computer Engineering
The University of Texas at Austin
deepu@ece.utexas.edu

Abstract – Digital Signal Processing (DSP) and multimedia workloads are expected to be the dominant workloads on future computer systems. This is true in both low cost embedded applications that use specialized microprocessors like DSPs and in the general-purpose processor market. Very Long Instruction Word (VLIW) architectures have multiple functional units to take advantage of vastly available Instruction Level Parallelism (ILP) in such applications. Single Instruction Multiple Data (SIMD) techniques operate on multiple data in a single instruction (exploiting data parallelism). This paper proposes to evaluate the benefits of using the above two techniques for DSP and multimedia applications. Using a modern commodity processor from each category – Texas Instruments Inc.'s TMS320C6x (VLIW) and Intel's Pentium II with MMX (SIMD), several DSP and multimedia benchmarks will be evaluated.

1. Introduction

Digital Signal Processing (DSP) and multimedia applications, where text becomes the exception rather than the rule, are now starting to become exceedingly important for computer systems as a dominant computing workload [5][6]. Dynamic multimedia component technologies such as video conferencing, video authoring, visualization, 3D graphics, animation, realistic simulation, speech processing and recognition, and broadband communications hold a great promise. In contrast to traditional applications, multimedia and DSP-rich applications will involve significant demands on the processor. With an ever-increasing proportion of CPU cycles being used to run such applications, it is pertinent to design machines that speed up programs that constitute a large portion of computation time.

Current solutions for these compute-centric applications are based principally on VLSI implementations except for certain control functions that may be implemented on a programmable micro-controller. To make the implementation flexible and cost effective over a variety of products and product generations, however, there is now a great deal of interest in migrating functionality from application specific hardware into software running on a programmable CPU or DSP.

The importance of multimedia technology, services and applications is being widely recognized by microprocessor designers. A number of manufacturers are offering multimedia processors that are claimed to be able to decode coded video streams in real-time in software. Most of such processors like the Trimedia processor from Philips and the Multimedia signal processor from Samsung usually have hardware assists for one or more of the multimedia decoding functions. The market for these special purpose multimedia processors will be in low cost embedded applications such as set-top boxes,

wireless terminals, digital TVs, and stand-alone entertainment devices such as DVD players. A number of general-purpose CPU manufacturers are offering multimedia enhanced versions of their CPUs for accelerating audio and video processing. The UltraSPARC processor enhanced with the Visual Instruction Set (VIS) from Sun, and the multimedia-enhanced MMX Pentium processors from Intel are examples. Such CPUs are likely to take over multimedia and DSP functions like audio-video decoding/encoding, modem, telephony functions, and network functions on a PC/workstation platform, along with the general purpose computing they currently perform.

2. Objectives and Motivation

DSP and multimedia applications possess several distinguishing characteristics than the normal workloads on desktop computing systems. Diefendorff and Dubey [5] specified the following characteristics of the media-centric applications – real-time response, processing of continuous-media types, significant fine and coarse grained parallelism, high instruction-reference locality, and high network and memory bandwidth. There is significant data and instruction level parallelism (ILP) that can be exploited in these workloads.

Very Long Instruction Word (VLIW) architectures incorporate multiple functional units in the data path to exploit the ILP in applications. A single instruction specifies more than one concurrent operation (for example, two loads, two adds, two multiplies and two shifts all in a single instruction). The instruction width is quite large (sometimes up to 8 times than normal architectures) and takes many bits to encode multiple operations. VLIW processors rely on software to pack the collection of operations (compaction) and in workloads with limited ILP, instruction bandwidth is wasted with no-operations placed

in the instruction. Examples of modern VLIW processors are from major DSP vendors – TI's TMS320C6x series, Analog Devices Inc.'s TigerSHARC and the joint venture of Motorola and Lucent known as StarCore. For multimedia and DSP applications VLIW processors seem to be an intuitive performance win over traditional single instruction per cycle architectures. Figure (1) shows the CPU core of the C6x processor having eight functional units in the data path.

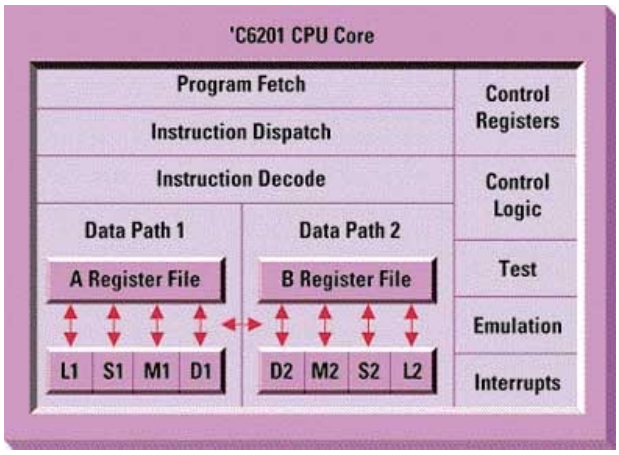

Figure. 1. CPU core of the C6x processor (VLIW)

Single Instruction Multiple Data (SIMD) techniques traditionally have been instruction set architecture extensions to general-purpose superscalar processors. Such architectures exploit data parallelism as opposed to ILP – each instruction operates on multiple data in a single instruction (for example, four loads or four additions, etc. but not a combination of different operations). Many of the DSP and multimedia applications can use vectors of packed 8-, 16- and 32-bit integers and floating-point numbers that allows potential benefits of SIMD architectures like the MMX for the Pentium family of processors and the Visual Instruction Set (VIS) extensions for the UltraSPARC processors. Figure (2) shows the "multiply and accumulate" instruction operating on multiple data in the case of MMX technology.

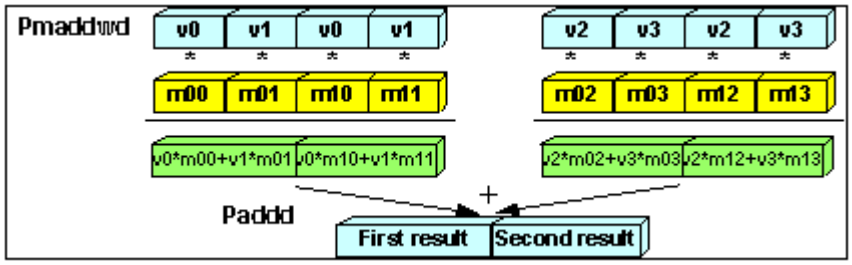

Figure. 2. Multiply and accumulate instruction operating on several data values

The objective of this project is to evaluate the effectiveness of VLIW and SIMD architectures for DSP and multimedia applications. Choosing one modern representative commodity processor from each category – TI's C6x DSP processor as a VLIW representative and Intel's Pentium II with MMX as a SIMD representative, we propose to assess DSP and multimedia workload performance on each processor.

3. Previous Work

Bhargava et al. evaluated the effectiveness of MMX instructions for DSP and multimedia applications on a Pentium processor [1]. The authors evaluated four kernels and four applications on a Pentium processor with MMX using VTune version 2.5. The kernels consisted of a Finite Impulse Response (FIR) Filter, Infinite Impulse Response (IIR) Filter, Fast Fourier Transform (FFT), and Matrix and Vector Arithmetic (matvec). The suite of applications were JPEG image compression, image manipulation, G.722 speech encoding, and Doppler radar processing. Measured statistics include the execution cycles, static and dynamic instruction count, amount of memory references and the number of MMX instructions executed (including a breakup of MMX instruction categories). The methodology adopted for this work was to compare efficient C code with

respect to MMX assembly code (obtained from Intel's optimized libraries for signal processing, image processing, etc.).

The amount of MMX instructions present in the benchmarks varied from 5% to about 90% of the total dynamic instruction count. As a fair comparison, the authors measured speedup of MMX code over optimized floating-point assembly code. Observed speedups ranged from 1.25 to 6.6 in the case of kernels and 0.49 to 5.5 for the applications. Two applications exhibited slowdowns, due to cases of excessive function calls, switching from MMX to floating-point code and vice versa, etc. though the kernel portions of the applications were sped up by MMX technology. It was observed that MMX technology decreased the dynamic instructions and execution time, but increased static code. Furthermore, code development using MMX meant resorting to library functions (which in turn were robust with error checking code, but increasing execution time) and the effects of precision create an additional burden on application developers. Compiler technology is yet to catch up in generating efficient SIMD code from C code (CodeWarrior claims to do it efficiently).

Lee et al. have proposed a suite of benchmarks for media processing and evaluated performance characteristics on an experimental IBM 40x PowerPC core in [2]. The goals of their work was to accurately represent the workload of emerging multimedia and communications systems by focussing on portable applications written in high-level languages and to develop a tool that is effective for system evaluation as well as system synthesis. Their suite MediaBench is composed of complete applications coded in high-level languages. They gathered a suite of 19 applications culled from available image processing, communications and DSP applications. Some of their benchmarks are JPEG, MPEG, GSM 06.10, G.721 Voice compression, PGP encryption, Mesa – a 3D graphics library, EPIC – an image compression utility, ADPCM, etc. However, they do not

evaluate any SIMD benefit or available parallelism in those benchmarks. They evaluated the instruction cache variation, bus utilization and the instructions per cycle and compared with general-purpose benchmarks from the SPEC suite. Performance/cost variation was studied with cache sizes and the main observation in their work was that the stress on both instruction and data caches was reduced in MediaBench workloads compared to SPEC benchmarks. The main benefit of this reference is the collection of benchmarks that can be evaluated on SIMD and VLIW architectures.

Ranganathan et al. have evaluated the performance of image and video processing with general-purpose processor and media ISA extensions [3]. They used detailed simulation of 12 benchmarks to study the effectiveness of current architectural features and identify future challenges for those workloads. They presented 6 kernels and 6 applications. Kernels were – addition, blending, convolution, dot-product, scaling and thresholding. Applications were – JPEG encoding (progressive and non-progressive), JPEG decoding (progressive and non-progressive) and MPEG (coding and decoding). They studied the VIS media extensions for in-order and out-of-order processor models (RSIM – their simulator has this ability to model different configurations and also include support for VIS). Their overall results were that both multiple issue and out-of-order issue processors provide substantial reductions in execution time for most of their benchmarks. Compared to a single-issue in-order processor, on the average, multiple issue improves performs by a factor of 1.2X, while the combination of multiple issue and out-of-order issue improves performance by a factor of 3.1X. The VIS media ISA extensions provided significant performance improvements for all the benchmarks (1.1X to 7X). Furthermore VIS was found to reduce dynamic instruction count, branch count, and number of memory instructions. Studies were also performed on the impact of caches and software prefetching. The major conclusion of their work is – on a single-issue in-order processor,

all of their benchmarks are primarily compute-bound. By using conventional ILP features and the VIS instructions together, 5 of their benchmarks became memory-bound. Increasing cache sizes showed no impact on several of their benchmarks. If software prefetching was added to the code, all of their benchmarks reverted to being compute bound.

Ongoing research in the Laboratory for Computer Architecture has had more of evaluating characteristics of multimedia applications on a Pentium II processor. Major conclusions of our studies indicate that branch frequency in multimedia applications is less than half of normal SPEC workloads, media extensions (MMX) provide significant improvement in overall performance and more functional units are desired for better processing of available ILP. Preliminary studies were also performed on several DSP kernels implemented on C6x and Pentium II processors.

In this project we propose to extend our work done on kernels onto several multimedia and DSP applications such as speech compression, image processing, 3D graphics, video processing, data encryption, etc. Tools to be used include – C6x stand-alone simulator, full-simulator and debugger for the C6x and VTune 3.0 and Performance counters for the Pentium II processor. Optimized assembly is easily obtainable for kernels, but difficult to code for complete applications. CodeWarrior compiler is to be evaluated for successful generation of MMX code from user written C code. Statistics for the execution time, number of dynamic instructions, %MMX instructions (if applicable), etc. will be gathered.

References:

[1] R. Bhargava, L. John, B. Evans and R. Radhakrishnan, "Evaluating MMX Technology Using DSP and Multimedia Applications", Proceedings of *IEEE Micro-31*, pp. 37-46, Dec 1998.

[2] C. Lee, M. Potkonjak and W.H. Smith, "MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems", *Proceedings of IEEE Micro-30*, pp. 330-335, Dec 1997.

[3] P. Ranganathan, S. Adve and N. Jouppi, "Performance of Image and Video Processing with General-purpose Processors and Media ISA Extensions", To appear in the *Proceedings of the 26th International Symposium on computer Architecture*, May 1999.

[4] Laboratory for Computer Architecture, "*Ongoing Research*", Department of Electrical and Computer Engineering, The University of Texas at Austin.

[5] K. Diefendorff and P.K. Dubey, "How Multimedia Workloads Will Change Processor Design", *IEEE Computer Magazine*, pp. 43-45, Sep 1997.

[6] C.E. Kozyrakis and D.A. Patterson, "A New Direction for Computer Architecture Research", *IEEE Computer Magazine*, pp. 24-32, Nov 1998.