An Independent Analysis of the

Tensilica Xtensa LX Processor with Vectra LX

By the staff of

BDTi Berkeley Design Technology, Inc.

Overview

With a BDTIsimMark2000™ score of 6150, the Tensilica Xtensa LX with Vectra LX plus custom instructions is among the fastest processors evaluated by BDTI to date. This report provides an analysis of the capabilities of the Xtensa LX with Vectra LX for digital signal processing applications. It provides insight into the strengths and weaknesses of the Xtensa LX with Vectra LX, comparative benchmark scores, and a discussion of the methodology and the instruction set customizations used in the benchmarking process.

Introduction

The Xtensa LX is a licensable, configurable 32-bit RISC processor core from Tensilica. Announced in May 2004, Xtensa LX is the sixth-generation Xtensa architecture, succeeding the Xtensa V, which was announced in August 2002. The Xtensa LX core targets a wide variety of applications, from low-power consumer electronics to high-performance communications infrastructure equipment.

Since its founding in 1997, Tensilica has emphasized instruction-set configurability as the primary feature that distinguishes Xtensa from other core offerings. At its heart, Xtensa LX is essentially a typical 32-bit RISC core with mixed-width 16- and 24-bit instructions. However, the Xtensa LX instruction set is configurable by licensees using a Verilog-like language called TIE (Tensilica Instruction Extension). Custom instructions added through the TIE language are compiled by Tensilica-provided tools and the associated new hardware is automatically inserted into a synthesizable model of the core. TIE supports the addition of new instructions that support operations ranging in complexity from simple arithmetic to complex, multi-stage computations that can add many thousands of gates to the Xtensa core. The Tensilica processor generator tool also generates a C/C++ compiler and cycle-accurate instruction-level simulator that are aware of the added instructions. Once a licensee is satisfied with the simulated results, the customized core can be implemented using logic synthesis tools and integrated into an ASIC design.

Tensilica has recently announced a new TIE-generating compiler called XPRES. In conjunction with the Tensilica Xtensa C Compiler, XPRES is capable of identifying the performance-critical regions of C/C++ application source code and automatically generating custom instructions that improve performance on this code. The tool generates many different combinations of candidate custom instructions, allowing the licensee to select the best instructions or manually fine-tune them.

A new feature of the Xtensa LX architecture is a VLIW instruction format called FLIX (Flexible-Length Instruction Xtensions). FLIX adds 32- and 64-bit instruction word formats to the base 16- and 24-bit formats found in previous generation Xtensa cores. The FLIX instruction formats support variable-length multi-issue instruction capability for custom instructions written in TIE. Xtensa LX can freely interleave a stream of 16-, 24-, and either 32- or 64-bit instructions without mode changes or stalls.

FLIX and TIE are the enabling technologies of Vectra LX, Tensilica's off-the-shelf DSP-oriented instruction set add-on for the Xtensa LX core. Vectra LX is a packaged group of powerful instructions designed to accelerate DSP applications. Vectra LX adds a quad-MAC unit and a significant number of single-instruction, multiple-data (SIMD) instructions to the base Xtensa instruction set. Vectra LX also adds a bank of sixteen 160-bit vector registers and a second 128-bit load/store unit (for a total of 256-bits/cycle of data memory bandwidth) to the base core.

Configurability for DSP Applications

In typical signal processing applications, processors spend most of their time executing relatively small inner loops of compute-intensive code. By configuring a processor to excel in the key operations found in those inner loops, it is often possible to obtain significant gains in speed and efficiency. However, the flexibility afforded by configurable processors brings some noteworthy trade-offs. One important consideration is the expertise and effort required to select and implement effective custom instructions. In addition, configurable processors tend to have less third-party development support in the form of tools and application software components. And the software development tools provided by configurable processor vendors tend to be less sophisticated than those available for popular fixed-architecture DSPs.

The primary goal of this report is to provide a brief analysis of the capabilities of Xtensa LX and Vectra LX for digital signal processing applications, including insight into its DSP performance based on BDTI Benchmark™ results. The following section provides details about the benchmarking methodology employed by BDTI to help ensure that the benchmark results presented here are representative of what a typical system-on-a-chip developer would achieve when using Xtensa LX with Vectra LX and custom instructions in a signal processing application.

The BDTI Benchmarks™

The BDTI Benchmarks are a set of twelve digital signal processing functions that BDTI has independently designed to provide an objective basis for comparing processor performance characteristics—such as speed and memory use—for signal processing applications. Implementations of the BDTI Benchmark functions are carefully optimized for a given processor to allow a realistic assessment of signal processing performance. BDTI Benchmark scores are available for a wide range of licensable DSP cores, packaged DSP processors, and general-purpose processors. Table 1 lists the twelve BDTI Benchmark functions.

Real Block FIR	Two-Biquad IIR	Viterbi Decoder
Single-Sample FIR	Vector Dot Product	Control
Complex Block FIR	Vector Add	256-Point FFT
LMS Adaptive FIR	Vector Maximum	Bit Unpack

Table 1. The BDTI Benchmarks™

Benchmarking Methodology

Benchmarking a processor with a configurable instruction set presents some significant challenges. Typically, the BDTI Benchmarks are implemented by an experienced engineer who writes optimized assembly code for the target processor. The process is more complicated for Xtensa LX because the benchmark implementor must not only write optimized assembly code, but must also choose, implement, and verify custom instructions that will increase performance on the benchmarks. The addition of a new instruction may necessitate a complete rewrite of existing optimized assembly code. It should be noted, however, that Xtensa LX licensees are not required to write custom instructions to use this processor.

The implementor is confronted with a variety of trade-offs when selecting and implementing custom instructions. For example, new instructions can significantly increase chip area and reduce achievable clock frequency. In short, the number of factors and trade-offs that must be considered is significantly increased when benchmarking a configurable processor, compared to benchmarking a fixed-architecture processor.

In general, the BDTI Benchmarks can be implemented by the processor vendor, by experienced engineers at BDTI, or a combination of the two. In all cases, adherence to the BDTI Benchmark specifications are certified by BDTI before results can be published.

The BDTI Benchmarks for Xtensa LX with Vectra LX were implemented and optimized jointly by Tensilica and BDTI. Tensilica was responsible for creating and verifying custom instructions, with some guidance on instruction selection from BDTI.

It should be noted that an experienced engineer from Tensilica was responsible for developing the custom instructions based on input from BDTI. In this respect, the benchmark implementation process and results shown here may differ from what licensees might experience when implementing and optimizing custom instructions for their own applications.

Tensilica's TIE-generating XPRES tool was not used in this benchmarking effort. Although the XPRES tool might have decreased the amount of time required for the benchmarking effort, it was decided that manual selection of custom instructions would result in the fastest and best-optimized benchmark implementations.

Custom Instruction Selection

Even without the addition of custom instructions, Xtensa LX with Vectra LX is a powerful architecture for typical signal processing tasks. Nonetheless, initial estimates suggested that all of the twelve BDTI Benchmarks would benefit from custom instructions.

The process of selecting custom instructions for use on the BDTI Benchmarks was constrained primarily by making conservative assumptions about the trade-offs that a typical Tensilica licensee would be likely to make. BDTI and Tensilica followed an instruction selection process that attempted to optimize the following characteristics, in rough order of decreasing priority: cycle counts on the BDTI Benchmarks, clock rate, program memory use, silicon area, and power consumption. It is important to realize that these characteristics often interact with each other. A custom instruction that quadruples the core area and cost but only offers a 10% performance improvement is probably not an attractive one. Whenever such conflicts arose during the benchmarking process, conservative trade-offs were made.

The BDTI Benchmarks, like typical DSP applications, spend most of their time in the inner loops of compute-intensive code. Thus, a guiding principle for selecting the custom instructions was the desire to combine several inner-loop operations into a single instruction in order to increase performance. For example, many common DSP functions exhibit a similar pattern: data is loaded into registers, one or more computations are performed on the data, and the results are scaled, possibly rounded, and stored. In many cases it is possible to execute some or all of these operations in parallel, saving a significant number of cycles during each software loop iteration. A typical high-performance VLIW DSP can execute several instructions in parallel, but in many cases the architecture imposes constraints that prevent the optimal use of available hardware on a particular DSP algorithm. In contrast, a configurable architecture such as Xtensa LX can provide instructions that execute a specific set of operations matched to the needs of a particular DSP algorithm loop.

One of the custom instructions added to Xtensa LX for use on the BDTI Benchmarks is called the BDTI MAC instruction. To understand the role of this instruction, the built-in MAC capabilities of Vectra LX must first be introduced. The Vectra LX extensions include a quad-MAC unit capable of computing four 16 × 16 multiplies on independent data per cycle. However, the processor's load/store bandwidth is adequate to sustain eight 16 × 16 multiplies per cycle. Because several of the BDTI Benchmarks make heavy use of MAC operations, this presented an excellent opportunity for a custom instruction. The BDTI MAC instruction is a custom instruction that creates four additional 16 × 16 MAC units and computes eight 16 × 16 multiply-accumulates per cycle. A further change was made to how results are accumulated. The Vectra LX quad-MAC instructions implement a "vertical" multiply-accumulate, where the results from four multiplies are accumulated into four separate accumulators in a SIMD result register. This vertical accumulation was not a good match for several of the BDTI Benchmarks. Thus, the BDTI MAC was defined so that it that accumulates "horizontally," i.e., it adds all eight results into a single accumulator.

There are three variants of the BDTI MAC instruction benefiting five of the benchmarks; the variants differ slightly in how the results are scaled before being stored to a result register.

Tensilica also added several new custom instructions for use on the Bit Unpack and Viterbi Decoder benchmarks. Tensilica had previously developed custom instructions that accelerated these functions, and adapted these custom instructions for use on the BDTI Benchmarks.

Final Core Configuration

In total, eleven custom instructions were added to the Xtensa LX with Vectra LX for use on the BDTI Benchmarks. In addition to the three BDTI MAC instructions, these include three instructions for the Bit Unpack benchmark and five for the Viterbi Decoder benchmark. The same customized core was used for all twelve BDTI Benchmarks.

According to detailed synthesis and physical compiler results developed by Tensilica, these eleven instructions resulted in a core area increase of about 16% over the base Xtensa LX with Vectra LX, for a total core area of roughly 4.42 square millimeters in a 0.13-micron TSMC process. The projected worst-case clock speed for the customized core in a 0.13-micron TSMC process is 369 MHz at 1.08 volts and 125 degrees Celsius. Estimated power consumption at 369 MHz is 200 mW, excluding power for memory.

Benchmark Results

In this section we compare BDTI Benchmark™ composite scores for the customized Xtensa LX with Vectra LX against scores for two other licensable DSP cores: the CEVA CEVA-X1640 and the StarCore SC1400. For additional insight into Xtensa's performance, we also compare this core against a high-end packaged DSP processor, the Texas Instruments TMS320C64x.

Speed: BDTImark2000™ and BDTIsimMark2000™

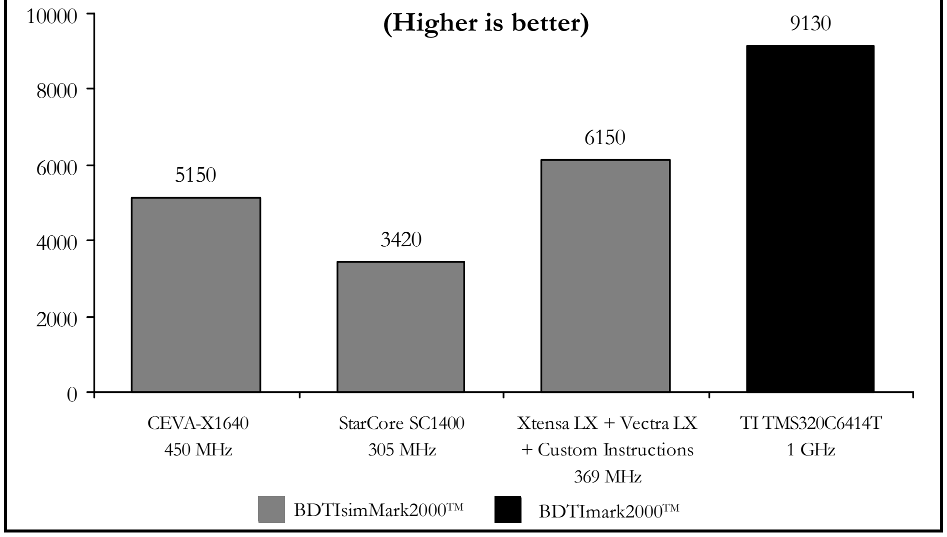
The BDTImark2000 and BDTIsimMark2000 are composite performance metrics that are based on a processor's speed on the full set of BDTI Benchmarks. BDTImark2000 scores are provided only when a processor's performance has been verified on hardware, whereas BDTIsimMark2000 scores are provided for processors for which only simulated results are available. For further information on the BDTI Benchmarks, the BDTImark2000, and the BDTIsimMark2000, see http://www.BDTI.com/benchmarks.html.

The BDTIsimMark2000 and BDTImark2000 are designed to provide a convenient shorthand for processors' signal processing speeds, and are far more accurate than simplified metrics such as MIPS or MFLOPS for this purpose. BDTImark2000 and BDTIsimMark2000 scores for the processors considered in this report are shown in Figure 1.

It is important to be cautious when comparing scores for chips to scores for cores. For chips, vendors guarantee that the processor will achieve a certain clock speed. For cores, the clock speed depends on the fabrication process, synthesis targets, and other factors. Hence, the clock speed of a core may vary dramatically from one chip design to the next.

For consistency, BDTI calculates scores for licensable cores using projected worst-case clock speeds in a 0.13 µm process. In this context, "worst-case clock speed" means the clock speed projected for a core assuming worst-case process, voltage, and temperature variations. For packaged processors, scores are computed using the fastest available family member.

Figure 1. Overall Speed: BDTIsimMark2000™ and BDTImark2000™ (Higher is better)



Overview of Speed Results

The score that a processor achieves on the BDTIsimMark2000 and BDTImark2000 metrics tends to be proportional to both its clock speed and its architectural parallelism. The ability to perform several multiply-accumulate (MAC) operations in parallel is particularly important. Like the applications they represent, many of the BDTI Benchmarks make heavy use of MAC operations. For these benchmarks, cycle counts are often closely related to the number of MAC operations a processor can perform in parallel.

It is important to note, though, that MAC throughput is not by itself a reliable predictor of performance. Many factors other than MAC throughput affect performance. This is particularly true for the single-sample benchmarks such as the Single Sample FIR, which tend to spend a minority of their cycles performing MAC operations.

All four processors shown in Figure 1 are VLIW DSPs capable of executing multiple instructions per cycle. Of the four processors, three are capable of executing four 16-bit MACs per cycle: the CEVA-X1640, StarCore SC1400, and Texas Instruments TMS320C64x. The customized Xtensa LX can achieve eight 16-bit MACs per cycle.

Although the CEVA-X1640 is technically an eight-issue VLIW DSP processor, it is capable of executing up to eleven operations per cycle through a feature called instruction duplication, which is used to implement SIMD operations across its data paths. This processor also includes an unusually diverse instruction set that is tailored for a wide variety of DSP applications. These features help the CEVA-X1640 achieve good cycle efficiency on the BDTI Benchmarks. Combined with its relatively fast 450 MHz clock rate, the CEVA-X1640 low cycle counts lead to a BDTIsimMark2000 that is more than 50% higher than that of the six-issue quad-MAC StarCore SC1400.

The fastest processor shown here is the quad-MAC eight-issue TMS320C64x. Although this packaged DSP does not exhibit the same level of cycle efficiency as the other processors shown here (especially the customized Xtensa LX), this processor combines moderate levels of architectural parallelism with a clock rate that is three times higher than that of the Xtensa core. This allows the 1 GHz TMS320C6414T to achieve a BDTImark2000 score that is nearly 50% higher than the BDTIsimMark2000 score of the Xtensa LX with Vectra LX.

BDTIsimMark2000 results for the customized Xtensa LX are discussed below.

Analysis of Xtensa LX Speed Results

The Xtensa LX with Vectra LX is technically only a three-issue VLIW processor. However, the addition of custom instructions allows this processor to execute many more benchmark-specific operations per cycle than the other processors considered here. In addition, even without custom instructions, Xtensa LX with Vectra LX is quite capable of achieving good performance on typical DSP applications. For example, it can perform eight-way SIMD operations with guard bits using eight 16-bit values in each of its SIMD registers. In addition, its two 128-bit load/store units are capable of supplying very high data bandwidth to its execution units. Overall, the DSP functionality provided by Vectra LX compares favorably to the other processors considered here.

The Vectra LX extensions do have some weaknesses for DSP applications, however. For example, as described earlier, the SIMD "vertical" accumulation of the Vectra LX

MAC instructions was not a good match for the MAC-intensive BDTI Benchmarks. Another minor weakness is the lack of non-SIMD variants of certain instructions. For example, loading a single 16- or 32-bit value automatically replicates the value across the fields in a 160-bit vector register, and the availability of non-SIMD shifts is limited.

Weaknesses in the Vectra LX instructions can generally be overcome with custom instructions. However, custom instructions are bounded by real hardware constraints, and realistically cannot reduce every relevant set of signal processing operations into just one instruction with single-cycle latency. For example, a complex operation implemented in TIE may significantly reduce the operating frequency of the core unless the operation is broken up into pipelined stages. Once implemented, such an instruction may be difficult to use effectively due to its longer latency. Also, certain resources are limited and must be monitored when developing custom instructions, such as the silicon area consumed by the added instructions, the number of state bits that can be used for temporary storage, and the instruction-slot opcode space when using the 32-bit or 64-bit FLIX instruction formats. Finally, because they are already so complex, some built-in features are also hardware-intensive and cannot be easily extended to support new features. For example, according to Tensilica it is difficult to add new access ports or extend the functionality of the Vectra LX 160-bit register file without significantly increasing area or reducing clock speed.

The BDTI MAC instructions developed for use on five of the benchmarks allow the customized Xtensa LX with Vectra LX to execute up to eight MACs per cycle, along with the accumulation and scaling required by the benchmark specifications. The customized Xtensa LX exhibits very high cycle efficiency on the BDTI Benchmarks, and its BDTIsimMark2000 score of 6150 is roughly 20% higher than that of the next fastest licensable core in Figure 1, the CEVA-X1640. However, as explained above, this high cycle efficiency is not enough to overcome the much higher clock speed of the TMS320C64x.

It should be stressed that the BDTIsimMark2000 score shown here for the Xtensa LX with Vectra LX applies only to the core configuration that was developed by Tensilica and BDTI. Although BDTI believes this score is indicative of the performance typical licensees will achieve when configuring the Xtensa LX with Vectra LX for signal processing applications, the actual performance achieved by licensees may differ significantly from the results presented here. This is because licensees may choose to make very different instruction choices or make different design trade-offs that could significantly increase or decrease performance for a given application.

Energy Efficiency

Overall energy efficiency is measured by the BDTImark2000/Watt metric. This is a composite performance metric based on a processor's typical energy use on the full set of BDTI Benchmarks. If a processor's BDTImark2000 score is known, its BDTImark2000/Watt score can be computed by dividing the BDTImark2000 score by the processor's power consumption. The BDTIsimMark2000/Watt metric is used for processors for which a BDTIsimMark2000 score is available.

For licensable cores, the BDTIsimMark2000/Watt score is computed based on estimated power consumption in an energy-efficient 0.13 μm fabrication process. This fabrication process differs from the high-speed 0.13 μm process used to calculate worst-case speeds in the BDTIsimMark2000 speed score shown in Figure 1.

As of this writing, a suitable power consumption estimate for the customized Xtensa LX in an energy-efficient 0.13 μm fabrication process was not available from Tensilica. Thus, the BDTIsimMark2000/Watt score for the customized Xtensa LX is not currently available.

However, power consumption estimates for both the customized Xtensa LX and the StarCore SC1400 are available for a high-speed 0.13 μm process. Thus, some general energy-efficiency comparisons can be made between these two cores. The customized Xtensa LX is projected to consume 200 mW at 1.2 volts and 369 MHz in a high-speed 0.13 μm fabrication process, whereas the SC1400 is projected to consume 201 mW at 1.2 volts and 305 MHz in the same process. These estimates do not include power for memory. The customized Xtensa LX has a BDTIsimMark2000 score that is nearly 80% higher than that of the SC1400 at these clock speeds, suggesting that the customized Xtensa LX will exhibit much higher energy efficiency than the SC1400 in typical DSP applications.

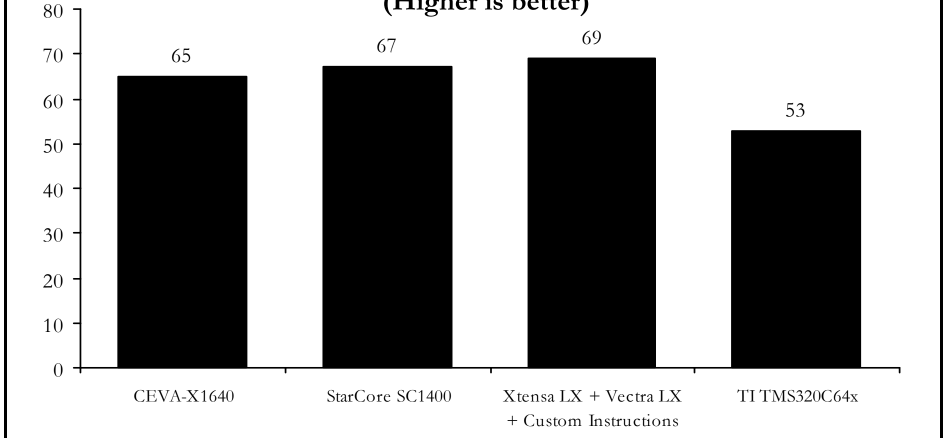
According to Tensilica, a number of power-saving features have been incorporated into Xtensa LX in order to achieve good energy efficiency. For example, the Xtensa LX includes extensive clock-gating capabilities. Tensilica's processor generation tools automatically partition the processor into separate clock subnets, and place each custom instruction into its own subnet. This results in hundreds of separate clock subnets in a typical Xtensa LX configuration.

Memory Efficiency: BDTImemMark2000™

The memory requirements of an application can have a significant impact on overall system cost. In addition, processors may experience significant performance degradation when application code and data do not fit in on-chip memory. Because of these and other factors, memory use efficiency is an important metric in processor selection.

This section focuses on processor memory use as measured by the BDTImemMark2000. This is a composite performance metric based on a processor's memory use on the full set of BDTI Benchmarks. The BDTImemMark2000 is designed to provide a convenient shorthand for processors' memory efficiency on typical signal processing applications. Most signal-processing applications are composed of instructions, constant data such as filter coefficients, and non-constant data such as input and output buffers. Therefore, the BDTImemMark2000 includes the program, constant, and non-constant data memory use for all twelve benchmarks. In most applications, the control-oriented code is much larger than the signal-processing code. Therefore, the BDTImemMark2000 assigns a much greater weight to Control benchmark results than it does to results for other benchmarks.

Figure 2. Overall Memory Efficiency: BDTImemMark2000™ (Higher is better)



Memory use is affected by a number of factors, including data word widths, instruction word widths, architectural parallelism, and instruction latency. Higher levels of architectural parallelism and longer instruction latencies tend to result in higher program memory use and lower BDTImemMark2000 scores. Conversely, processors with lower levels of architectural parallelism and shorter instruction latencies tend to use less program memory and have higher BDTImemMark2000 scores.

BDTImemMark2000 scores are shown in Figure 2. The CEVA-X1640, StarCore SC1400, and customized Xtensa LX with Vectra LX all exhibit relatively good BDTImemMark2000 scores. Although these processors are all multi-issue processors, they achieve good memory use efficiency through efficient instruction encodings. They are each capable of using relatively short 16-bit instruction word widths for the majority of control-oriented code, resorting to longer word widths for increased performance on DSP algorithm code.

Notably, the use of custom instructions can drastically reduce Xtensa LX program memory requirements for DSP algorithm code. This is because one custom instruction typically performs multiple operations, and in extreme cases can represent many hundreds of operations that normally would require hundreds of instructions to implement. Entire program loops can sometimes be compressed to just a few custom instructions, obviating the need for significant loop unrolling or other performance-enhancing techniques often employed on VLIW architectures. On the Viterbi Decoder benchmark, for example, the Xtensa LX with Vectra LX consumes roughly 80% less program memory than the CEVA-X1640 and StarCore SC1400 due to its heavy use of custom instructions.

The TMS320C64x has the lowest BDTImemMark2000 score, which indicates relatively poor memory use efficiency. This can be attributed to three primary factors: it issues up to eight instructions per cycle, it has large fixed-length instructions (each instruction is 32 bits), and has long instruction latencies (which promote loop unrolling and software pipelining).

Programming Effort and Tools

This section provides brief comments about BDTI's experience of the programming effort required to develop optimized DSP software for Xtensa LX, and the available software development tools.

If custom instructions are not considered, the programming effort required to write hand-optimized assembly code for Xtensa LX with Vectra LX extensions is similar to that of other multi-issue VLIW processors. The three instruction issue slots do have some restrictions on the types of instructions they can contain, but the restrictions do not significantly increase programming effort.

The BDTI Benchmarks were implemented and verified using Tensilica-provided command-line GNU tools. These tools include a compiler, assembler, and instruction-set simulator, each of which is aware of custom instructions added by the user. This suite provides an adequate platform with which to develop DSP functions using assembly code, but is somewhat basic compared to feature-rich development platforms available for some packaged DSPs.

According to Tensilica, a graphical integrated development environment is now available for its development tools.

The TIE language is very similar to Verilog, so those familiar with Verilog will require less time to learn and use TIE. The TIE development environment includes a library of macros for common hardware functions such as multipliers and adders. One TIE feature that was used on the BDTI Benchmarks is a hardware sharing facility that allows multiple custom instructions to share common hardware resources. This facility was used in the BDTI MAC instructions, which use eight total MAC units: four dedicated MAC units and four MAC units that are shared with the Vectra LX instructions.

Tensilica's tool suite does not include logic synthesis or other EDA tools. Instead, the tools generate a synthesizable model of the processor along with configuration files for the instruction-level simulator. Synthesis and post-layout simulation of the processor is performed using third-party EDA tools.

Conclusion

Tensilica has been targeting DSP applications for some time, but Vectra LX is clearly its most direct effort to bring the Xtensa architecture into the domain of licensable DSP cores and packaged DSP chips. Enhancing a general-purpose processor with DSP-oriented features is not new among processor vendors; companies like ARM and Renesas have been busy adding such features in the past few years. But with its eight-way SIMD operations, wide SIMD registers, and huge memory bandwidth, Vectra LX is one of the most aggressive DSP add-ons developed to date for an embedded general-purpose processor core. Tensilica's combination of a powerful off-the-shelf DSP add-on with support for custom instructions makes the Xtensa LX with Vectra LX a candidate for a wide range of signal processing applications that require a custom chip.

One of Tensilica's strongest selling points is performance. Because virtually any set of DSP operations can be encapsulated into custom instructions, customized Xtensa LX cores should be capable of outperforming most DSPs and general-purpose processors on most DSP applications. In addition, for battery powered applications, Xtensa's high performance can lead to power savings if the core can operate at a reduced clock frequency and voltage while still meeting throughput needs. And because custom instructions target a specific application, an Xtensa LX may be more area-efficient than a processor core that attempts to perform well on a wide range of applications but is only used for one specific application.

However, using a customized Tensilica processor in a system-on-a-chip design presents a variety of development challenges that do not arise when using a fixed-architecture core. For example, because of the tendency for custom instructions to be application-specific, a particular customized Tensilica core is less likely to be used across a range of applications; Tensilica licensees are likely to create a number of customized processor variants. These variants will not be fully compatible, limiting the re-use of optimized software from one customized core to another.

Relatedly, because of instruction set differences from one customized Xtensa LX core to another, it is more challenging for third-party software component and tool providers to support Tensilica cores compared to fixed-architecture cores. As a result, the software development infrastructure surrounding Xtensa LX and Vectra LX is less extensive than that associated with established fixed-architecture cores. Recognizing the need for off-the-shelf solutions, Tensilica has begun to offer application solutions which include a preconfigured core along with application software, but currently only a few of these solutions are offered.

An additional trade-off relates to the physical implementation of the processor core. Typically, a licensee of a fixed-architecture processor core will "harden" the core—that is, create an optimized physical implementation—once, and then re-use the hardened core in multiple designs. With a customizable core, in addition to the effort required to customize the core, the hardening process must be repeated for each new customized version created.

Despite the increased development effort and limited range of off-the-shelf software components, Tensilica processors are an attractive option for many signal processing applications that require a custom chip. For example, custom instructions can be used to accelerate new or unique applications for which there are no effective off-the-shelf solutions.

For ASIC designers targeting typical signal-processing applications, the significant performance and efficiency benefits of a customized processor may very well outweigh the added costs and effort associated with "rolling your own" processor. For such designers, the Xtensa LX with Vectra LX merits consideration.