



XPRES

PRODUCT BRIEF

FEATURES

- ANSI standard C/C++ input
- Automatic generation of Tensilica Instruction Extension (TIE) specifications for a complete processor
- GUI-driven, interactive control and feedback visualization
- Comprehensive analysis of millions of possible processor instruction set extensions exploring SIMD, fusion and FLIX™ architectural options, in a matter of minutes

BENEFITS

- Automatic specification of high-performance embedded processors without modification of the original C/C++ source code
- Further optimization (optional) through manual tuning of the processor or the C code
- Run other software on the tailored processor and take advantage of the optimizations without modification
- Graphical interface helps designers select the right application acceleration versus hardware cost optimization point for the target system design

XPRES Compiler

The Tensilica® XPRES™ (Xtensa® PProcessor Extension Synthesis) Compiler is a powerful synthesis tool that creates tailored processor descriptions for the Xtensa 6 and Xtensa LX processors from native C/C++ code. Algorithm and systems designers can use the XPRES Compiler, coupled with Tensilica's proven Xtensa Processor Generator technology, to synthesize highly optimized processor hardware RTL directly from C/C++ reference code or algorithmic specifications.

This tool enables the rapid development of optimized SOC hardware blocks and associated software tools without requiring designers to hand code their hardware using design languages like VHDL and Verilog, which take months of design and verification effort. Instead, designers input the original algorithm that they're trying to optimize, written in standard ANSI C/C++, and the XPRES compiler automatically determines which functions should be accelerated in hardware. Various area/performance/power trade-offs can be quickly evaluated using the graphical output from the XPRES Compiler.

Optimized Hardware Directly from Source C Code

The C programming language is the natural language of choice for algorithm developers and systems architects to capture and model system behavior. For more than a decade, the embedded design industry has struggled to find a path that takes those C specifications of algorithms and automatically or systematically transforms them into efficient hardware designs. A variety of approaches – with buzzwords like behavioral synthesis, C language hardware synthesis, and ESL – have fallen short because they all have tried to conquer a nearly intractable problem – transforming a language designed to be executed in a sequential manner on a microprocessor into a highly parallel system of interoperating, non-programmable hardware elements.

XPRES Compiler Takes Advantage of Xtensa LX FLIX Capabilities

The Xtensa LX processor incorporates Tensilica's FLIX (Flexible Length Instruction Xtensions) architecture. FLIX allows designer-defined instructions to consist of multiple, independent operations bundled into a compact 32-bit or 64-bit instruction word that coexists with the native 16-bit and 24-bit Xtensa ISA. The FLIX architecture allows the implementation of highly parallel processors with a range from 2 to 15 parallel execution units. Thus Xtensa LX processors can deliver the ultra-high performance characteristic of specialty ultra-wide instruction word processors without the code bloat typically incurred by such VLIW or ULIW processors.

The XPRES Compiler enables designers to rapidly explore the benefits of FLIX by automating the analysis of the cost-benefit tradeoffs of the parallelism of FLIX.

FLIX

Instruction extensions for the Xtensa LX processor that exploit the FLIX architecture allow the combination of multiple independent operations scheduled and bundled at compile time by the XCC Compiler. To achieve higher performance, FLIX supports multiple independent execution pipelines and adding additional ports to Xtensa LX register files. The XPRES Compiler comprehensively evaluates the performance benefit of creating FLIX implementations versus the hardware cost factor when creating optimal processor configurations.

Specialized Operations

Use of the specialized operations technique reduces the size of an instruction so more independent operations can be packed together into a single FLIX instruction bundle. The XPRES Compiler can create these specialized instructions from existing operations, including base Xtensa LX ISA operations. An example: the base Xtensa LX ISA includes the 16-bit ADDI.N instruction that includes a 4-bit immediate value and two register specifiers into the base Xtensa LX register file. The XPRES Compiler can create specialized versions of ADDI that include a narrow set of register specifiers and/or a narrower range of immediate values. The XPRES Compiler can also create highly specialized versions of ADDI that include a limited number of fixed constants for the immediate value if the application frequently performs such computations. These optimizations can result in a specialized ADD instruction that requires as few as 9 bits to encode in a FLIX bundle.

This technique can be quite useful in highly-optimized application realms. Example: color space conversion algorithms in consumer applications often implement a series of multiply-by-constant operations, with a limited set of constant values, which are highly amenable to this technique.

Tensilica's XPRES Compiler takes a different approach. The XPRES Compiler does not transform the C code. It automatically tailors the Xtensa microprocessor into an efficient machine that executes the original C code much faster. The tailoring process takes only minutes. Using the XPRES Compiler, designers often find that the automatically generated Xtensa processor is fast enough to meet their system's specifications. In other cases, engineers can further extend the processor with manually generated custom instructions to get even higher performance.

XPRES Compiler for Xtensa Processors

Xtensa 6 and Xtensa LX processors are the only processor cores for system-on-chip (SOC) designs that provide I/O bandwidth, compute parallelism, and low-power optimization equivalent to hand-optimized, RTL-designed, non-programmable hardware blocks.

The configurable, extensible and synthesizable Xtensa 6 and Xtensa LX processor cores are unlike any other conventional embedded processor or DSP core. The SOC designer can mold one or more Xtensa processors to exactly fit the target task. The designer selects and configures predefined processor attributes and adds descriptions of execution datapaths and registers that can deliver performance, area, and power characteristics equivalent to custom logic design.

The XPRES Compiler automates and accelerates the design and optimization of Xtensa 6 and Xtensa LX processors. Compared to traditional hardware design, Xtensa processors deliver similar quality of results with the added benefits of accelerated design time and post-silicon software programmability. This makes Xtensa processors an ideal choice for all blocks in complex SOC designs.

Processor Extensions - Accelerating Processor Performance to Custom Logic Design Levels

The Tensilica Instruction Extension (TIE) language is used to describe new instructions, new registers and execution units, and new I/O ports that are then automatically added to Xtensa processors by the Xtensa Processor Generator. TIE is a Verilog-like language used to describe new registers, register files, and desired instruction mnemonics, operands, encoding and execution semantics. TIE files are inputs to the Xtensa Processor Generator. The Generator automatically builds a version of the Xtensa processor and the complete tool chain that incorporates the new TIE registers, execution units, and instructions.

The XPRES Compiler automates the analysis and creation of TIE descriptions, dramatically shortening the design time needed to transform an algorithmic concept into an optimized, pre-verified, software-programmable hardware element.

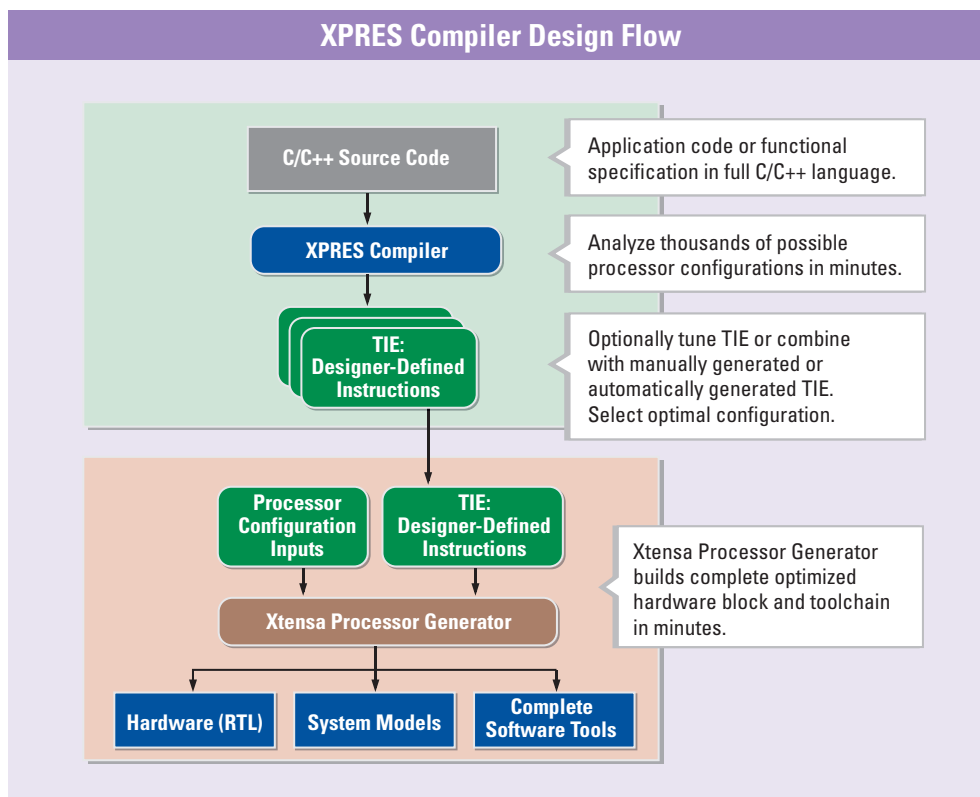


Figure 1: The XPRES Compiler automates the creation of extensions for Tensilica's Xtensa 6 and Xtensa LX processors. It works with Tensilica's proven Xtensa Processor Generator technology to rapidly create optimized hardware blocks for advanced SOC designs.

Automated Processor Optimization without C Code Modifications

The XPRES Compiler works with the Xtensa C/C++ Compiler (XCC) to analyze the performance-critical regions of your application software or algorithm specification. The XPRES Compiler then uses this analysis to generate one or more alternative TIE files designed to increase the performance of your code. The generated TIE files represent a range of customized Xtensa processors [Figure 2] that trade-off increased application performance versus increased hardware (area) cost.

Comprehensive and Automated Software Tool Support

Every Xtensa processor is automatically generated with a complete, matching set of software development and modeling tools tailored to the exact processor configuration. The Xtensa Tools suite includes the high-performance vectorizing Xtensa C/C++ Compiler (XCC); a GNU-based assembler, linker, debugger, and profiler; the pipeline-modeled, cycle-accurate Xtensa Instruction Set Simulator; and the Xtensa Modeling Protocol (XTMP) system simulation API. And all of the Xtensa development tools are tightly integrated within the Xtensa Explorer development environment, which integrates software development, processor optimization and multiple-processor SOC architecture tools into one common design environment.

It's easy to utilize the new TIE instructions with your original program or any other program. Just recompile the application program and XCC will automatically take advantage of new register files, new instructions and FLIX combinations of independent operations. No source code modifications are required. Because the XPRES Compiler creates TIE source files, the system designer is free to modify the TIE for further optimizations.

Performance Acceleration versus Hardware Cost Trade-off Analysis

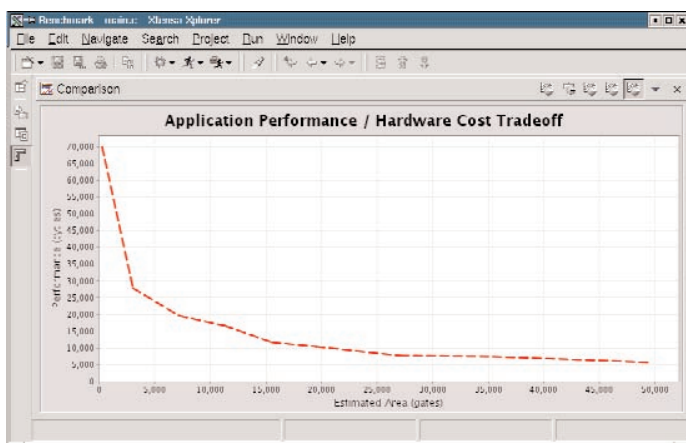


Figure 2: XPRES Compiler rapidly explores millions of possible processor configurations.

Wide Range of Optimization Techniques

The XPRES Compiler explores a wide range of possible optimizations using four main acceleration techniques: operator fusion, SIMD/vectorized operations, parallelism of independent operations using FLIX (LX only), and specialized operations.

Fusion

Operator fusion is a technique that creates instructions (operations) that consist of several simpler operations. A simple example: combining a basic ADD and SHIFT operation to form an ADD_SHIFT instruction that executes in one cycle. This ADD_SHIFT instruction could replace two sequentially issued instructions, thus saving a clock cycle and saving code-size. Fusion can be used to combine existing base Xtensa ISA instructions, or other operations previously created using TIE. The XCC Compiler, when compiling C code into a binary executable, utilizes sophisticated graph-matching algorithms to automatically infer the best use of the fused operation to replace individual, simple operations. XPRES Compiler offers visualization and control mechanisms [Figure 3] to allow the designer to optionally explore and control the number and types of fusions created.

Vector / SIMD

Vector operations increase performance by performing the same logical operation simultaneously on more than one data element. Example: a 2-wide vector addition operation can perform two simultaneous 32-bit additions from one 64-bit register location. XPRES Compiler automatically explores 2-, 4- and even 8-wide implementations of SIMD operations and explores vectorized versions of both base Xtensa ISA operations as well as SIMD versions of manually generated TIE operations. When compiling C code into a binary executable, the XCC Compiler utilizes advanced vectorization techniques to “unroll” inner loops of performance-intensive applications to take advantage of the SIMD versions of such operations without the need to modify the C code to explicitly use the SIMD functions. XPRES Compiler weighs both the added hardware cost of parallel execution units needed for SIMD operations and the added register-file cost of wider operands when evaluating SIMD techniques for acceleration.

Using the XPRES Compiler

The XPRES Compiler gives the design team complete usage flexibility. It can be used in full automation mode, where a C/C++ program is input and optimized TIE instructions are output, or it can be used under full designer control, so the designer can guide the tool, select instructions, and even tune the original application to take better advantage of the added hardware instructions.

A simple 5-step process explains how the XPRES Compiler is used.

Step 1. Compile the original C/C++ application code. No recoding is required. Tensilica's C/C++ compiler generates information about the application, performing such functions as ranking code regions by frequency, determining which loops can be vectorized, generating dataflow graphs for important regions, and performing operation counts for each type of opcode for every region.

Step 2. Run the XPRES Compiler to determine the best processor configuration and extensions for that code. The XPRES Compiler evaluates all generated configurations across all

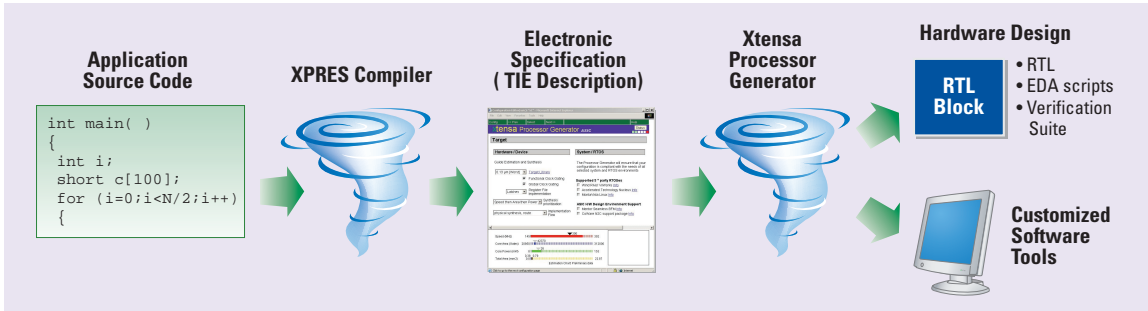
regions and determines the best set of merged configurations given a particular gate-count budget. The XPRES Compiler is able to conduct abstract evaluations and search through millions of configuration possibilities, usually in less than an hour.

Step 3. (Optional) Manually tune the automatically generated custom configuration. Power users will want to refine or optimize the code and add additional instructions for algorithms or functions that other programs might need.

Step 4. Generate the optimized processor. Use Tensilica's proven Xtensa Processor Generator technology and, in less than one hour, generate the complete processor RTL with EDA support, complete software-development tool chain, simulations and modeling environment, and RTOS support.

Step 5. Compile and use the original, unmodified C/C++ application using the newly optimized processor configuration. No need to modify the C code to make it Tensilica-specific. No need to use time-consuming assembly level optimizations. No need to design custom hardware accelerators using traditional RTL design methods.

From an ANSI C / C++ application the XPRES Compiler generates an optimized set of processor extensions ...



... that is reusable over a range of similar application software code.



Tensilica, Inc.

3255-6 Scott Boulevard, Santa Clara, CA 95054-3013, USA

Tel. 408-986-8000 • Fax. 408-986-8919 • Website: www.tensilica.com

© 2005, Tensilica, Sea of Processors and Xtensa are registered trademarks of Tensilica, Inc. XPRES, FLIX, Vectra and Xplorer are trademarks of Tensilica, Inc. All other trademarks are the property of their respective owners.