



xtensa 7

PRODUCT BRIEF

FEATURES

- 32-bit synthesizable RISC architecture with 5-stage pipeline, 16/24-bit instruction encoding with modeless switching
- Designer-configurable processor options (MMU/MPU, local memory types and sizes, hardware multipliers, etc.)
- Optional Designer-defined application-specific instructions can be added to base architecture
- XPRES™ Compiler automates generation of instruction extensions from C/C++ algorithms
- Automated fine and coarse-grain clock gating for ultra-low power
- Local memories can include parity or ECC

BENEFITS

- Extremely efficient base architecture that is smaller, lower power, and has better code density than other 32-bit embedded processors
- Application-specific instruction extensions provide orders-of-magnitude application performance improvements, eliminating the need for RTL blocks from SOCs
- Pre-verified, correct-by-construction RTL generation lowers verification efforts
- Reduces design risk with post-silicon programmability using processors instead of RTL blocks

Xtensa 7 Configurable Processor Core

Tensilica's Xtensa® 7 processor is a 32-bit RISC-based processor core that can be extended to out-perform any other embedded processor (and in some cases approach hand-optimized RTL) for a specific application utilizing optional custom instructions. The Xtensa 7 processor is extremely versatile – it is well suited for control-plane as well as data-plane SOC applications. With Tensilica's unique automated processor generator, every Tensilica® customer is able to quickly generate a tailored version of the Xtensa processor optimized for their particular application. Designed to be a configurable, extensible architecture, the Xtensa 7 processor is the basic building block for complex SOC designs.

The Xtensa 7 processor is unlike any other conventional embedded processor core – the system designer molds the processor to fit the target application. By selecting and configuring predefined elements of the architecture and by inventing completely new instructions and hardware execution units, the Xtensa 7 processor can deliver performance levels orders of magnitude faster than alternative solutions. The designer defines new instructions utilizing the Tensilica Instruction Extension (TIE) methodology, adding Verilog-like descriptions of datapaths, execution units, and register files that can deliver performance, area, and power characteristics equivalent to custom logic design. Or, the designer can use the Xtensa Processor Extensions Synthesis (XPRES) Compiler to analyze the C/C++ algorithm and automatically suggest configuration options and extensions that will run that algorithm faster. 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, making Xtensa 7 processors an ideal choice for all complex SOC designs.

FEATURE SUMMARY

Tensilica's Xtensa Processor Generator technology creates a complete RTL hardware description of a customized Xtensa processor, plus a comprehensive suite of software development tools, EDA implementation scripts, and system models – all in about 1 hour.

Optional Pre-defined Execution Units

- 32-bit or 16-bit multiplier
- 16-bit MAC
- Floating-point unit

Configurable Interface Options

- Optional processor interface (PIF) to system bus, choice of 32-, 64- or 128-bit width with in-bound DMA [slave mode] option
- Optional high-speed Xtensa Local Memory Interface (XLMI)
- Write Buffer: selectable from 1 to 32 entries

Multiple Processor Design Support

- System modeling capability: optional Xtensa Modeling Protocol (XTMP) simulation environment
- Multiple-processor on-chip debug capable with break-in/out control
- Optional processor ID interface and special register
- Memory synchronization and conditional store instruction option provides support for memory semaphore operations and the release consistency model of memory-access ordering

Complete Hardware Implementation and Verification Flow Support

- Automatic generation of RTL and tailored EDA scripts for leading edge process technologies, including physical synthesis and 3D extraction tools
- Auto-insertion of fine-grained clock gating delivers ultra-low power
- Hardware emulation support including automated FPGA netlist implementation
- Comprehensive diagnostic test bench

High-speed, High-accuracy System Simulation Models Automatically Created for Each Configuration

- Pipeline-modeling, cycle-by-cycle accurate Xtensa instruction set simulator
- High-speed (40X-80X) instruction accurate ISS for software development
- Multiple-processor simulation with XTMP option
- SystemC transaction-level models with XTSC option
- Hardware-software co-verification model for Mentor Seamless

Comprehensive Development Environment and Software Tool Support

- Xtensa Explorer™ development environment accelerates the analysis and development of Xtensa processor cores and SOCs designed with multiple Xtensa processors
- Seventh generation Xtensa Tools development suite automatically configured for each processor, including the advanced Xtensa C/C++ compiler (XCC)
- Operating Systems: Mentor Graphic's Nucleus Plus, Express Logic's ThreadX, Micrium Technologies' uC/OS-II, MontaVista Software's Linux Professional Edition, and Sophia Systems' µITRON

XPRES Compiler Support

- Automated configuration synthesis from C/C++ source code, no code modification required

PERFORMANCE SUMMARY

Processor Architecture

- High-performance 32-bit RISC with 5-stage pipeline

Instruction Set

Xtensa ISA with compact 16-bit and 24-bit base instruction set

	AREA	CLOCK RATE		POWER DISSIPATION	
	Optimized for area	130nm LV process, speed-optimized netlist. Worst case conditions.	90nm GT process, speed-optimized netlist. Worst case conditions	130 nm LV process, speed-optimized netlist. Typical operating conditions	90nm GT process, speed-optimized netlist. Typical operating conditions
RTOS-ready base configuration 5-stage	28,000	350 MHz	600 MHz	76 µW/MHz	94 µW/MHz
Minimum configuration	20,000	350 MHz	600 MHz	38 µW/MHz	48 µW/MHz

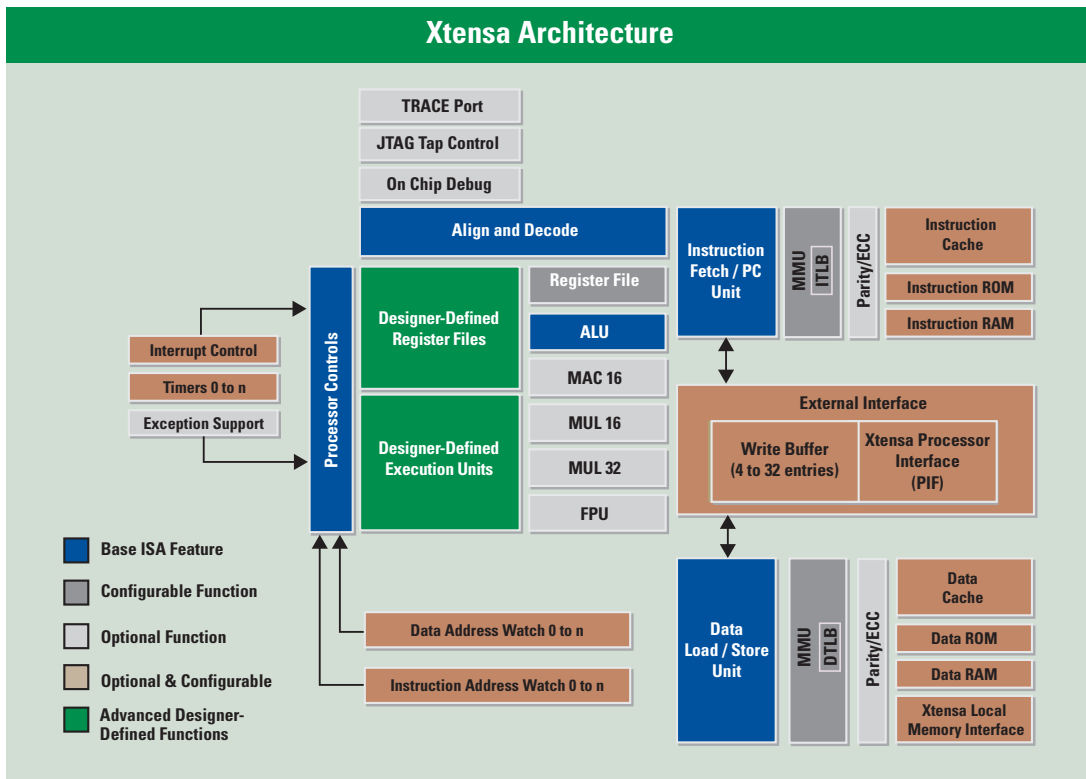


Figure 1: The Xtensa architecture consists of various configurable building blocks. CONFIGURABLE FUNCTION blocks are elements parameterized by the system designer. OPTIONAL FUNCTION blocks indicate elements available to accelerate specific applications. OPTIONAL AND CONFIGURABLE blocks are optional elements scalable to fit applications, including peripherals. ADVANCED DESIGNER-DEFINED FUNCTIONS are hardware execution units and registers added to the processor by the designer to accelerate specific algorithms for a given SOC design. Common in all configurations is the base instruction set architecture.

SOC Design Methodology

SOC design provides higher performance, lower cost, smaller form factor and longer battery life through lower power. But SOC designers often experience design bottlenecks in block-level integration and verification, hardware/software co-development and process technology portability. Until recently, embedded SOC designers have had to develop solutions based around rigid processor cores that were designed for workstation architectures, and augment those cores with large amounts of custom logic. The Xtensa 7 processor solution can handle not only conventional processor tasks, but is easily augmented to deliver the performance and power of custom logic. The Xtensa processor is quickly integrated with other system blocks and is easily adapted to the needs of today's high volume, high performance embedded applications.

Post-RISC Processor Architecture

The Xtensa 32-bit architecture features a compact instruction set optimized for embedded designs. The base architecture has a 32-bit ALU, up to 64 general-purpose physical registers, 6 special purpose registers and 80 base instructions, including improved 16- and 24-bit (rather than 32-bit) RISC instruction encoding (with modeless switching). The Xtensa CPU implements the proven Xtensa instruction set architecture (ISA),

which enables designers to achieve significant code size reductions compared to conventional RISC cores. Reducing code size results in higher performance and better power dissipation – key to saving cost in highly integrated SOC designs. The Xtensa ISA's 16-and 24-bit encoding also provides powerful branch instructions and zero-overhead loops, and bit manipulations including funnel shifts and field-extract operations.

Processor Extensions - Accelerating Processor Performance to Custom Logic Levels

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

Adding TIE instructions to a Tensilica processor core never compromises the underlying base Xtensa instruction set, thereby ensuring availability of a robust ecosystem of third

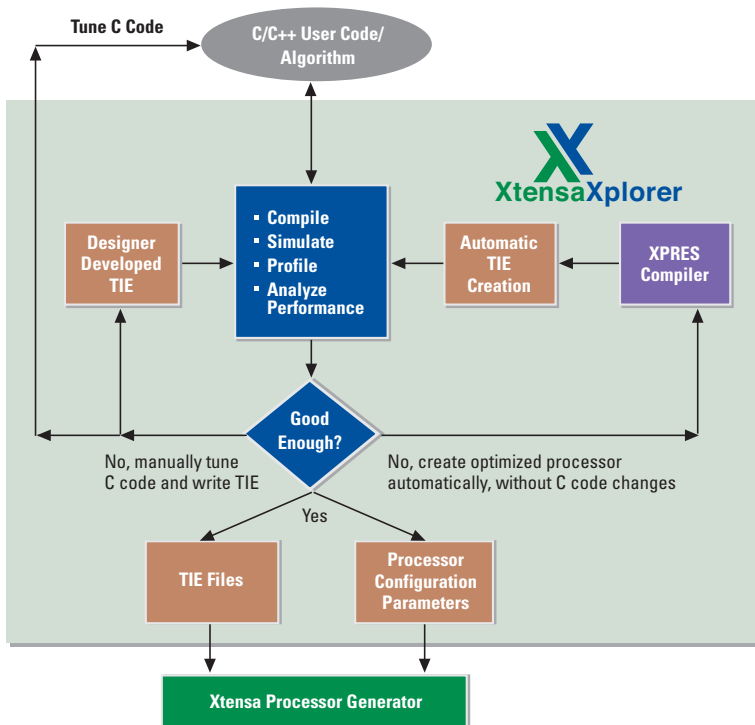


Figure 2: Tensilica's Xtensa Xplorer design environment enables engineers to either automatically generate custom TIE instructions to optimize the processor (using the XPRES Compiler) or manually develop TIE instructions. Output from Xtensa Xplorer is input to the Xtensa Processor Generator [see Figure 3].

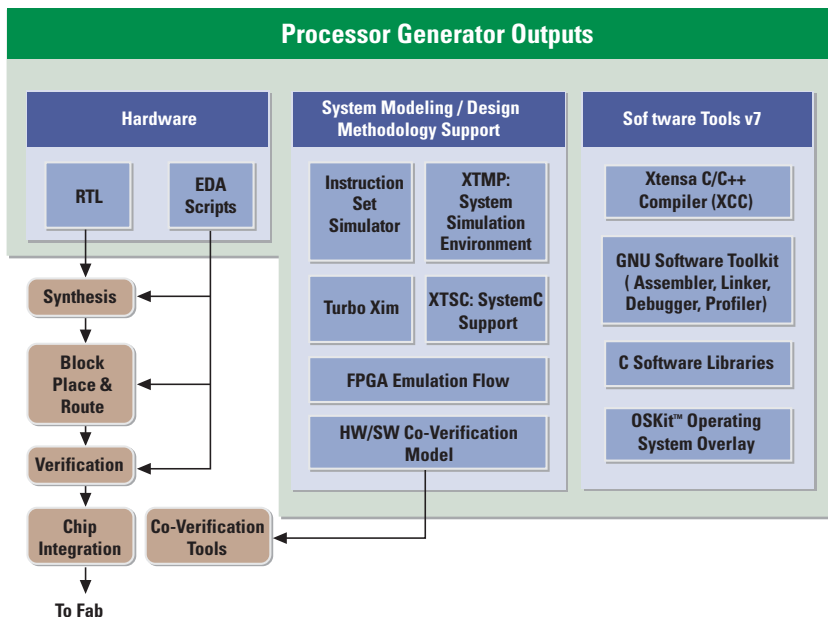


Figure 3: The Xtensa processor generator automatically creates matching hardware, software and system modeling deliverables.

Highly Configurable Base Processor Feature Set

Via the Xtensa Xplorer design environment, the Xtensa Processor Generator allows quick configuration of many Xtensa core options, including:

Execution Unit and ISA Options

- Multipliers: 32- or 16-bit
- Single 16-bit MAC
- Floating point unit
- Multiple-processor synchronization instruction set option

Interface Options

- Processor interface (PIF)
 - Width: 32/64/128-bit
 - Optional “no PIF” configuration
 - Inbound DMA option
- XLMI high-speed local interface
- Big-Endian/Little-Endian byte ordering
- On-chip debug port
- Trace port signals
- Up to 32 interrupts

Memory Subsystem Options

- Memory Management Options
 - Region Protection
 - Region Protection with Translation
 - Memory Management Unit (MMU) with Translation Look Aside Buffers (TLBs), includes no-execute bit security support
- Local data and instruction caches
 - Up to 4-way set associative
 - Up to 32 KB
 - Write-back and write-through cache write policy
- Separate local RAM, ROM areas for data, instructions up to 256Kbytes each
- Optional parity or ECC for all local memories

Design Support

- Pipeline-modeled, cycle-accurate instruction set simulator
- High-speed (up to 40X-80X) instruction-accurate simulator for software development
- Xtensa Modeling Protocol (XTMP) for multi-CPU simulation
- Xtensa SystemC (XTSC) transaction-level modeling (TLM) support
- Hardware/software co-verification model for Mentor Graphics' Seamless
- Mentor Graphic's Nucleus Plus, Express Logic's ThreadX, Micrium Technologies' μ C/OS-II, MontaVista Software's Linux Professional Edition, and Sophia Systems' μ ITRON operating systems

party application software and development tools. All configurable, extensible Xtensa processors are always compatible with major operating systems, debug probes and ICE solutions; and always come with an automatically generated, complete software development toolchain including an advanced integrated development environment based on the ECLIPSE framework, a world-class compiler, a cycle-accurate SystemC-compatible instruction set simulator, and the full industry-standard GNU toolchain.

Program optimization using TIE can be accomplished in two ways - automatically with Tensilica's XPRES Compiler or manually by the designer. The XPRES Compiler 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 are trying to optimize, written in standard ANSI C/C++, and the XPRES compiler automatically generates custom instructions in the TIE language that accelerate the performance critical portions of the application. The designer can also quickly evaluate various area/performance/power trade-offs using the visualization and analysis tools available in the Xtensa Explorer environment. The XPRES C-to-hardware tool significantly reduces the time required to generate application-specific processors based on custom application instructions.

If the design team decides to develop their own TIE instructions to accelerate the Xtensa 7 processor, manual TIE development process starts with the identification of "hot spots" - performance-sensitive regions of application software. By utilizing the execution profiler, the designer is able to analyze the efficiency of an application program and evaluate where TIE can be used to accelerate the performance of the software. The designer can quickly weigh the benefits of adding custom TIE instructions and functional units by iteratively profiling the application with the new instructions. Aggressive use of parallelism and other techniques can often deliver 10X, 100X or even greater performance increases using the new TIE instructions [see Figure 4]. Designers with existing application software can profile the application, identify hot spots, add new instructions and execution units to optimize performance and regenerate a new processor - all within a matter of hours.

Hardware designers with a reference specification can quickly design execution unit semantics with the desired datapath characteristics, add new instructions to the processor, and write simple C programs to test the new processor - all from within the integrated Xtensa Explorer design environment. The programmer uses the added instructions in C-code as intrinsic function calls, and the Xtensa compiler automatically does all optimizations and scheduling of the assembly code.

Ideal for Applications Where Low Power is Critical

Power is quickly becoming the key issue in SOC design, as consumers are demanding longer battery life in portable devices and systems designers struggle with heat dissipation requirements. Tensilica employs many techniques to reduce power consumption, both built in to the base hardware and configuration options allowing more control over system and memory interfaces. Tensilica processors consistently consume less power than other licensable embedded CPUs at equivalent gate counts.

Clock gating is a very effective power reduction technique that shuts down clock supply to parts of the logic that are not in use on a particular clock cycle. Tensilica has automated the insertion of fine-grained clock gating for every functional element of the Xtensa processor including functions conceived of and created by the designer. This automation gives the Xtensa a significant advantage over RTL design, where manual, error-prone post-layout tuning of clock circuits is often required. Coarse-grain clock gating is also implemented, where large areas of the processor are put into idle mode, when certain long latency operations are executed, such as a cache line refill.

Accessing local memories is one of the highest power-consuming activities an embedded processor must perform. Tensilica has designed the Xtensa processor to eliminate any unnecessary local memory interface activation if it that memory is not directly addressed by the processor, the implementation of this power-saving technique is automatically inserted by the Xtensa Processor Generator. In addition, the designer can configure the external data bus width and internal local memory data widths independently. This allows system-level power optimizations depending on whether the processor is constrained by external or internal instruction and data accesses.

The Xtensa processor's architecture dramatically lowers power consumption in large configurations with many designer defined functions. But even without the inclusion of designer-defined functionality, the Xtensa processor is designed to use power very efficiently. The minimum configuration of the Xtensa processor dissipates less than 30 $\mu\text{W}/\text{MHz}$ in a representative 130 nm process technology.

Soft Memory Error Detection and Correction

As process geometries continue to shrink, soft memory errors caused by alpha particle collisions with embedded memory cells increase due to lower cell capacitances and lower supply

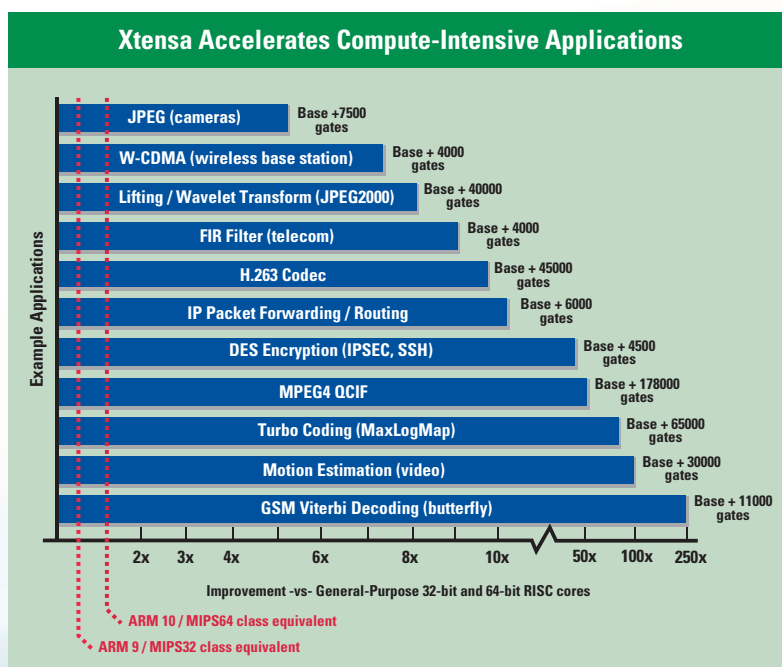


Figure 4: Designer-defined extensions can have a huge impact on system performance

voltages. Xtensa processors can be configured to detect or correct memory errors using either parity or ECC (Error-Correcting Code). Parity will generate an exception when a single-bit soft error is detected in the cache data array, cache tag array, or local memory (instruction and/or data memories). ECC is also available as an option in Xtensa processors; it will correct single-bit errors and detect double-bit errors. Error correction is extremely important in storage and networking applications where a single-bit error can be catastrophic.

Comprehensive and Automated Software Tool Support

Every Xtensa processor is automatically generated with a complete set of software development and modeling tools tailored to the exact Xtensa configuration. All of the Xtensa development tools are tightly integrated within the Xtensa Xplorer development environment [see Figure 5]. This environment integrates software development, processor optimization and multiple-processor SOC architecture tools into one common design environment.

The Xtensa software development tools are generated from the same database as the processor hardware description. This assures correctness and consistency by construction. Designers get a compiler, linker, assembler, and debugger tuned exactly for their hardware. The software tool chain is automatically updated and optimized to make use of the designer-defined instructions added during the hardware-generation process.

The Xtensa C/C++ Compiler (XCC) is an optimizing and parallelizing compiler for the Xtensa instruction set architecture and uses proprietary techniques to enable various instruction packages based on the particular hardware configuration. As is the case with the rest of the standard Xtensa GNU-based software development tool suite, XCC is generated from the same database as the processor hardware description to assure correctness by construction.

The Xtensa Instruction Set Simulator (ISS) is a multiple-processor-capable instruction set simulator that is callable in the designer's system API. This means multiple ISSs can be instantiated in a C/C++ environment along with other system components for accurate subsystem-level simulation and analysis. The Xtensa Modeling Protocol (XTMP) is an environment for Xtensa processor subsystem modeling and simulation. It allows rapid assembly of system-level simulations of one or more Xtensa processors and various memories and building blocks. With the Xtensa ISS and XTMP, designers can rapidly build and simulate complete SOC subsystems comprised of multiple, heterogeneous Xtensa processors.

The Xtensa OSKit™ consists of configured overlays for Monta Vista Linux. The OSKit overlays ensure that all configured features of the architecture, including designer-defined instructions are fully supported by the standard RTOS runtime environment. Other operating systems do not require the OSKit overlays. In addition, support for Mentor Graphics Seamless Co-verification Environment is provided for the Xtensa architecture.

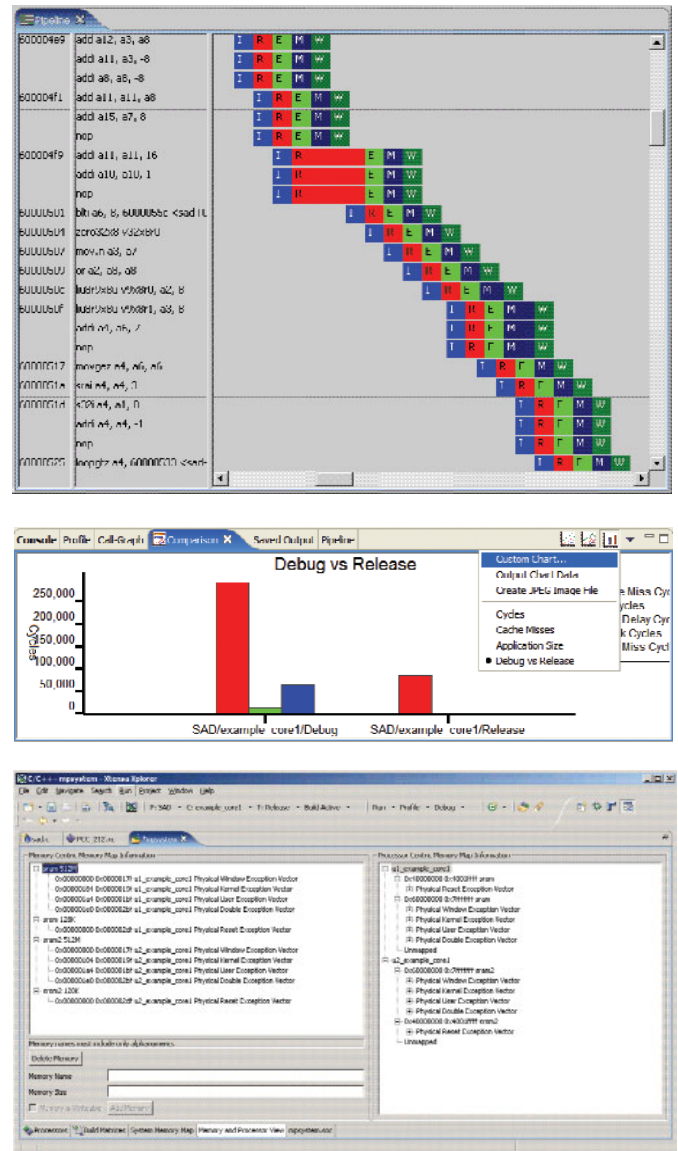


Figure 5: The Xtensa Xplorer development environment provides powerful visualization tools to assist in the configuration of an Xtensa processor



Tensilica, Inc.
 3255-6 Scott Boulevard, Santa Clara, CA 95054-3013, USA
 Tel. 408-986-8000 • Fax. 408-986-8919 • Website: www.tensilica.com

© 2006, Tensilica, Sea of Processors and Xtensa are registered trademarks of Tensilica, Inc. Tensilica logo and Xplorer are trademarks of Tensilica, Inc. All other trademarks are the property of their respective owners.