

Xtensa Processor Developer's Toolkit

Rapidly Customize Standard Processors for Differentiation

Product Brief

FEATURES

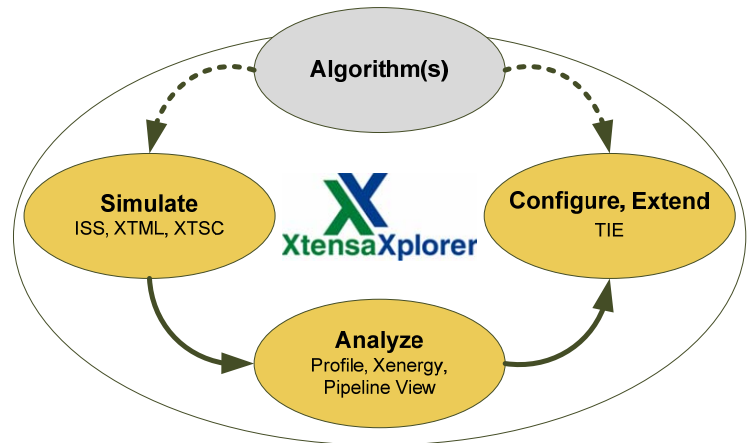
- Comprehensive, integrated Xtensa dataplane processor design and optimization environment
- Xtensa® Xplorer™ Integrated Development Environment (IDE) with full graphical user interface (GUI)
- Click-box and drop-down configuration options
- Simple Verilog-like Tensilica Instruction Extension (TIE) language for processor customization
- Multiple processor design support
- Fully integrated with Xtensa Software Developers Toolkit (see separate product brief) for software compilation, profiling, simulation, debugging, etc.
- TIE Port/Queue Wizard makes creating interfaces to other system RTL simple

BENEFITS

- Easy-to-use Xplorer IDE based on familiar Eclipse platform
- Quickly optimize the processor to efficiently implement your algorithms
- Differentiate your solution with unique processor optimizations
- Many processor configurations can be saved, profiled and compared to narrow down the best solution
- Easily create multiple-core subsystems in minutes for evaluation and load profiling in simulation
- Automatically generate customized processor RTL with full synthesis scripts in about an hour
- RTL is pre-verified and guaranteed to be correct by construction
- All software development tools and simulation models are generated automatically with the processor

The Most Powerful and Complete Processor Design Environment

Tensilica delivers patented, proven tools that automate the process of generating a custom processor along with matching software tools. This is our ninth generation of these tools, which have been proven in hundreds of designs. Whether your design is for a simple controller or a complex multi-core DSP design, Tensilica has the tools you need to create successful products.



Tensilica's Eclipse-based Xtensa Xplorer IDE serves as the cockpit for custom processor development

Automated Processor Development

If you've looked at Tensilica's web site or processor product briefs, you know that you can extend Tensilica's Xtensa dataplane processors (DPUs) – adding instruction sets, execution units, processor I/O interfaces – to match your specific application needs.

By customizing the DPU for a particular application you can often get significantly lower energy consumption and 10-100x performance increases. This level of performance and efficiency is often essential in the SOC dataplane.

By customizing the DPU, you also create a core that's uniquely yours. This gives you extra protection in today's highly competitive marketplace.

The Xtensa Processor Developer's Toolkit is the integrated design environment that delivers powerful tools to your desktop that guide you through the processor customization process. You'll find that Tensilica has created the most advanced, powerful and easy-to-use tools for processor customization.

The Processor Developer's Toolkit is required for any design team that is using Tensilica's TIE (Tensilica Instruction Extension) instructions to modify the processor. If you are using an Xtensa processor with no modification or only changes to configuration options, you do not need the Processor Developer's Toolkit – you'll only need the Software Developer's Toolkit (see separate product brief).



Dataplane. DPU. Differentiate.

A Comprehensive System

Now in its ninth generation, Tensilica's tools are highly refined and provide developers with a complete, comprehensive solution for both system design and software development. Tensilica's Processor Developer's Toolkit contains all the tools necessary to create highly customized Xtensa processors.

Everything You Need to Build High-Performance Xtensa DPUs

Tensilica's Processor Developer's Toolkit contains all the tools necessary to create and analyze extremely high-performance application-specific Xtensa DPUs. Tensilica's Xtensa Xplorer IDE serves as the cockpit for the entire design experience. From

Xtensa Xplorer, you can profile your application code, identify "hot spots" that can benefit from acceleration, and add the TIE instructions necessary to speed up that code.

Using a check-box menu within the GUI, you can configure processors to include features you need and remove features you don't – options for processor interface, memories, operating system support, EDA scripts, debug and trace, and much more are supported.

Tensilica's TIE language is similar to Verilog and simplifies your ability to add optimizing custom instructions to Xtensa processors. By modifying the Xtensa processor, Tensilica's customers often get 10 to 100 times better performance and lower energy consumption when compared to alternative processor architectures. This allows Xtensa processors to be used in critical dataplane SOC functions where standard microprocessors or DSPs

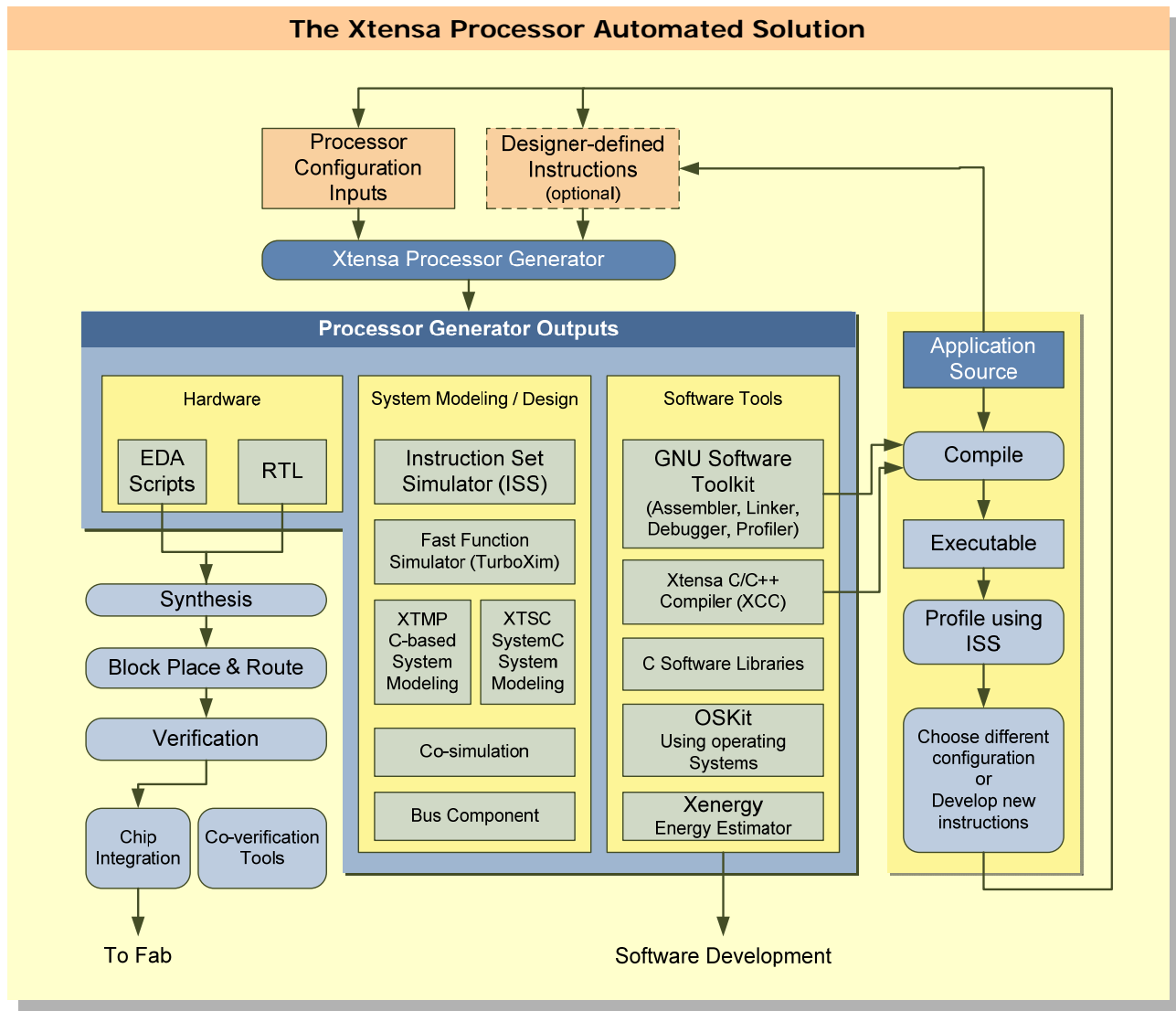


Figure 1. Tensilica's proven methodology automates the creation of customized processors and matching software tools



can not meet the needed performance, throughput or energy budget. Or, until now, the only alternative was to use hand-coded RTL hardware blocks.

For any TIE changes, updates to the entire Xtensa toolchain including the ISS and SystemC models, can be made in minutes on your desktop.

You can profile, compare and save many different processor configurations, so you can narrow down the right one for your application. Also, you can model and simulate multiple-processor subsystems in this environment using Tensilica's Xtensa Modeling Protocol (XTMP) or SystemC (XTSC) – see Xtensa Software Developer's Toolkit product brief.

For integration into the dataplane with custom logic, Tensilica's Xtensa SystemC (XTSC) can be used for simulation of DPUs with SystemC Transaction Level Modeling (TLM) hardware. Co-simulation with RTL level blocks and the Xtensa ISS is possible using Pin Level XTSC. This offers pin-level, cycle-accurate modeling of Xtensa DPU interfaces for use in Verilog simulations.

Xtensa Xplorer serves as the gateway to the Xtensa Processor Generator. Once a processor configuration is finalized, the Xtensa Processor Generator creates the automatically verified Xtensa processor to match all of the configuration options and extensions you have defined, in about an hour. The full software tool chain is also created, matching all processor modifications you have made.

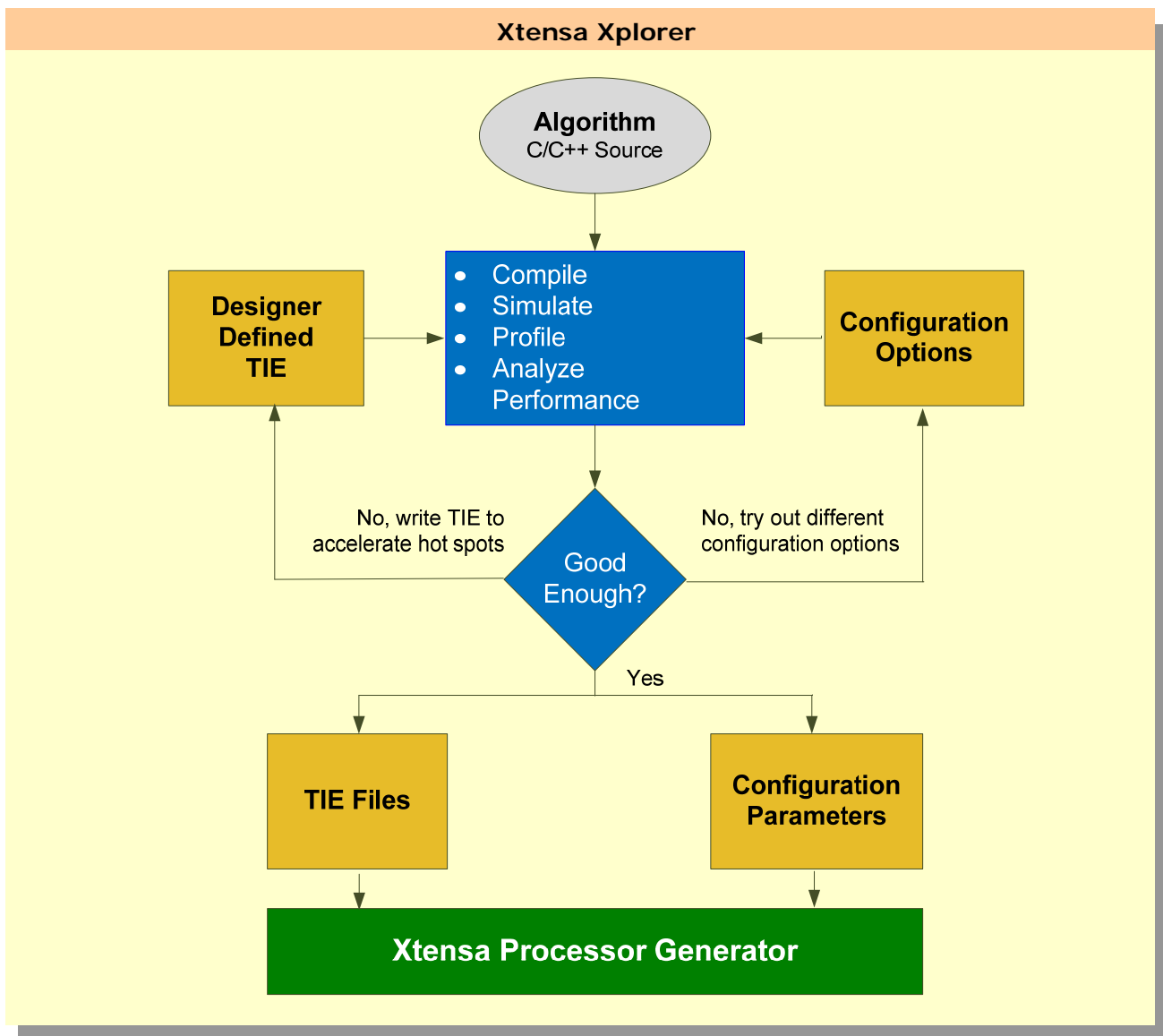


Figure 2. Xtensa Xplorer serves as the cockpit for custom processor development

Pipeline Viewer

Xtensa Xplorer includes a sophisticated Pipeline Viewer that helps you visualize the performance impact of designer-defined instructions, without the need to become a processor pipeline expert. In one quick step, Xtensa Xplorer's Pipeline Viewer will illustrate the impact of an instruction on the execution pipeline, providing instant feedback on the efficiency of a proposed new instruction and helping you tune the source TIE for optimal implementation. See Figure 3.

Analyze and Iterate

Xtensa Xplorer not only provides the tools to quickly develop optimized processors and simulate systems of processors, but also provides the tools that enable you to visualize and analyze simulation results, tune system or processor configurations, and rapidly compare alternative implementations. See Figure 4.

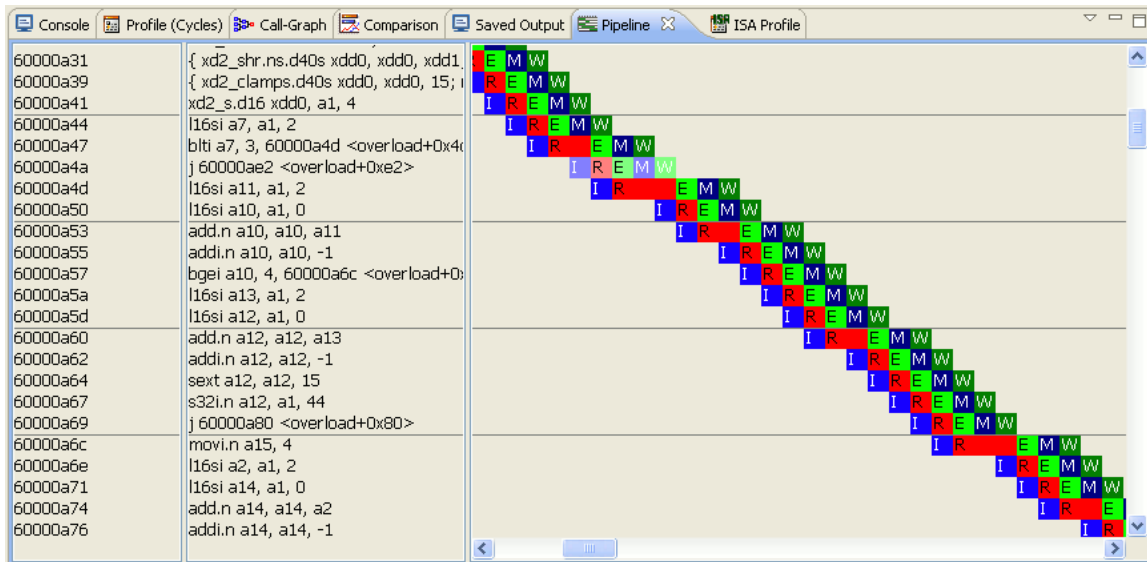


Figure 3. Pipeline Viewer shows instruction flow of disassembled code

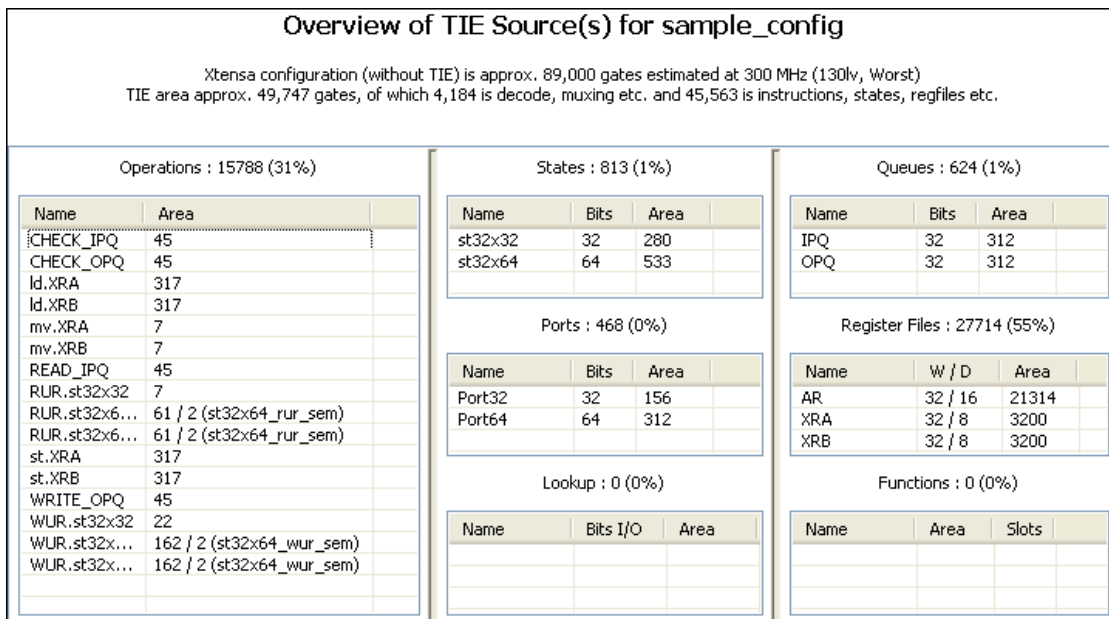


Figure 4. The TIE analyzer estimates gate count and silicon area for new instructions



The screenshot displays the Xtensa LX4 configuration interface, organized into several sections:

- Memory Management Options:** Includes 'XEA2: Region protection' (dropdown), 'Memory management selection' (dropdown), 'I-entries per way' (dropdown, value 4), and 'D-entries per way' (dropdown, value 4).
- Arithmetic Instruction Options:** Includes 'MUL32 implementation selection' (dropdown, value None), 'MUL16' (checkbox), 'MAC16 DSP instruction family' (checkbox), 'CLAMPS' (checkbox), '32 bit integer divider' (checkbox), and 'Double Precision FP Accelerator' (checkbox).
- ISA Instruction Options:** Includes 'NSA/NSAU' (checkbox, checked), 'Sign Extend to 32 bits' (checkbox, checked), 'Enable Boolean Registers' (checkbox), 'TIE arbitrary byte enables' (checkbox), 'Synchronize instruction' (checkbox), 'Number of Coprocessors (NCP)' (dropdown, value 0), 'Thread Pointer' (checkbox), 'MIN/MAX and MINU/MAXU' (checkbox, checked), 'Enable density instructions' (checkbox, checked), 'Enable Processor ID' (checkbox), 'Zero overhead loop instructions' (checkbox, checked), 'Conditional store synchronize instruction' (checkbox), and 'Miscellaneous Special Register count' (dropdown, value 0).
- ISA Configuration Options:** Includes 'Number of AR registers for call windows' (dropdown, value 32), 'Byte order (endianness)' (dropdown, value Little endian), 'Action when handling an unaligned load / store' (dropdown, value Take exception), 'Max instruction width in bytes. 4 byte and greater instructions are part of the FLIX option' (dropdown, value 3), and 'Pipeline length' (dropdown, value 5).
- DSP Coprocessors:** Includes 'Vectra LX DSP coprocessor instruction family' (checkbox), 'VectraVMB: Extra DSP Instructions' (checkbox), 'Vectra adapts to match the number of configured Load/Store units' (checkbox), and 'ConnX D2 DSP' (checkbox).
- HIFI Audio Coprocessor:** Includes 'HIFI2 Audio Engine DSP coprocessor instruction family' (checkbox) and 'HIFI EP Audio Engine DSP extensions' (checkbox).
- Fixed Core Extensions:** Includes 'FLIX3: 3-way FLIX' (checkbox).
- ConnX BBE16 Coprocessor Family:** Includes 'BBE16 Baseband Engine' (checkbox), '8-way vector divide' (checkbox), '4-way reciprocal square root' (checkbox), and '16-way Despreader' (checkbox).

Figure 5. Xtensa LX4 configuration screen

Configuration Options

You don't need to be locked into a set of features that were predetermined years or decades earlier by a processor designer. You can configure your Xtensa processor to exactly match your application.

Not sure exactly what options you need? Our Xtensa Explorer IDE will help you evaluate tradeoffs. We offer tools that can help you make intelligent decisions about options such as memory types and sizes, interface options and other execution unit and ISA options.

Configurability of a Tensilica processor core never compromises the underlying base Xtensa instruction set, thereby ensuring availability of a robust ecosystem of third 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.

Click-Box Choices

It's as easy as checking a box to pick the options for your application. You get more choices with Xtensa LX4, but even the choices in Xtensa 9 can make big differences in your design. See Figure 5 for an Xtensa LX4 configuration screen shot.



The TIE Language

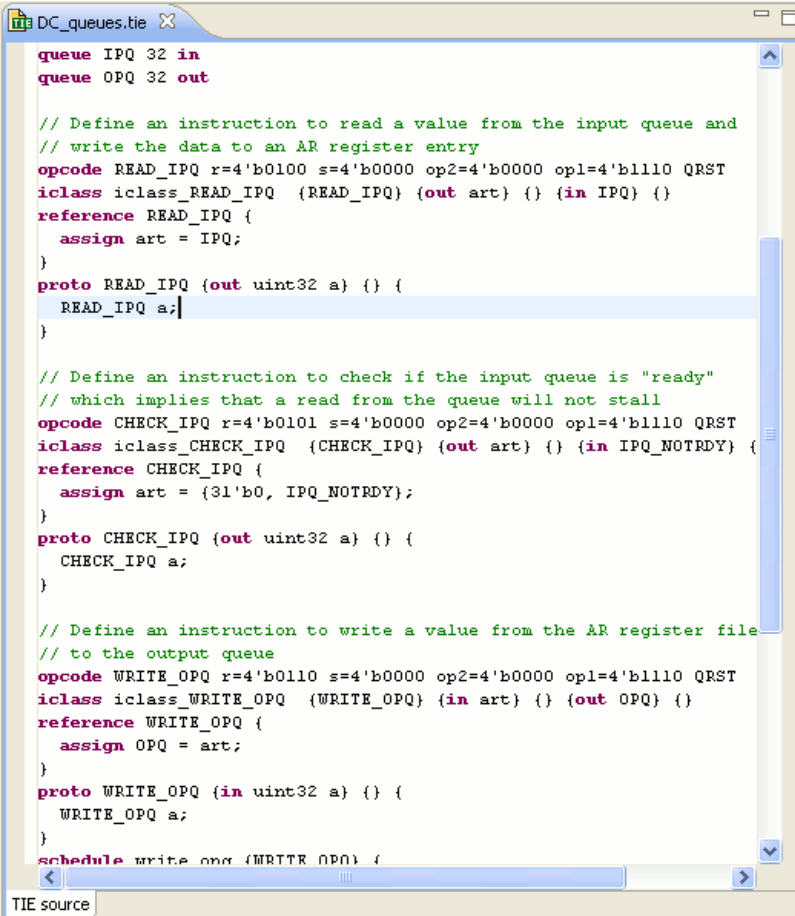
The TIE language offers a wide range of flexibility in adding multi-cycle, pipelined execution units, register files, state registers, SIMD arithmetic and logic units, creating wide (up to 512-bit) Load/Store instructions, and adding designer-defined I/O Ports, Queues, and Lookups.

The TIE Compiler

While you can select from a large number of check-box configuration options, the real power of Tensilica's Xtensa design environment comes from the use of TIE. TIE "bridges the gap" between the software and hardware design realms, as it is a hybrid of C and Verilog and very easy to learn. TIE lets you add new processor functionality in the form of instructions, execution units, wide load-store instructions, designer-defined I/O interfaces, and designer-defined register files and state registers – all without the need to modify (and then verify) the processor RTL.

The TIE Compiler is a tool that you use during the development of custom processor hardware extensions. The TIE Compiler also updates the compiler tool chain (XCC compiler, assembler, debugger, profiler) and the simulation models (instruction set simulator and the XTMP, XTSC and pin-level XTSC system modeling environment) so they understand and fully utilize these new functions.

You use the TIE Compiler on your local workstation/PC to determine the best instructions for the performance you need combined with any area and power considerations. Iterating in minutes, once you've determined the optimal TIE instructions for your application, that TIE file becomes input for the Xtensa Processor Generator. The Processor Generator automatically produces the entire processor RTL, including the base processor with all configuration options and all TIE extensions plus a complete matching software tool chain.



```
DC_queues.tie X
queue IPQ 32 in
queue OPQ 32 out

// Define an instruction to read a value from the input queue and
// write the data to an AR register entry
opcode READ_IPQ r=4'b0100 s=4'b0000 op2=4'b0000 op1=4'b1110 QRST
iclass iclass_READ_IPQ {READ_IPQ} {out art} {} {in IPQ} {}
reference READ_IPQ {
    assign art = IPQ;
}
proto READ_IPQ {out uint32 a} {} {
    READ_IPQ a;
}

// Define an instruction to check if the input queue is "ready"
// which implies that a read from the queue will not stall
opcode CHECK_IPQ r=4'b0101 s=4'b0000 op2=4'b0000 op1=4'b1110 QRST
iclass iclass_CHECK_IPQ {CHECK_IPQ} {out art} {} {in IPQ_NOTRDY} {}
reference CHECK_IPQ {
    assign art = {31'b0, IPQ_NOTRDY};
}
proto CHECK_IPQ {out uint32 a} {} {
    CHECK_IPQ a;
}

// Define an instruction to write a value from the AR register file
// to the output queue
opcode WRITE_OPQ r=4'b0110 s=4'b0000 op2=4'b0000 op1=4'b1110 QRST
iclass iclass_WRITE_OPQ {WRITE_OPQ} {in art} {} {out OPQ} {}
reference WRITE_OPQ {
    assign OPQ = art;
}
proto WRITE_OPQ {in uint32 a} {} {
    WRITE_OPQ a;
}
schedule write opq {WRITE_OPQ} {
```

Figure 6. The TIE editor simplifies new instruction development



Energy Xplorer (Xenergy)

During development, it's not always clear whether one way of implementing an extension is lower power than another way without running simulations in RTL. That can take a long time to do.

Xenergy is an Xplorer tool that allows you to compare the energy usage and power dissipation of multiple implementations without the need to run RTL simulations. It's designed to highlight the relative energy usage of your application for various:

- Implementation choices
- Operating conditions
- Memory technologies
- Cache configurations
- Clock speeds, and
- Memory latencies.

View and compare the results of different scenarios side by side while your design is refined.

TIE Port/Queue Wizard

When creating point-to-point connections between the processor and other parts of your system, Tensilica offers Port and Queue interfaces along with all of the instructions needed to read and write to them.

These are created by writing TIE, but you can use the TIE Port/Queue Wizard to create them if you'd like to see how the TIE should be written. The Wizard creates a TIE file based upon your input along with a testbench showing read/write operations.

FLIX and Specialized Operations

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-, 64-, or 128-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 of 2 to 30 parallel execution units. Thus Xtensa LX processors can deliver the high performance characteristic of specialty long instruction word processors without the code bloat typically incurred by such VLIW or ULIW architectures.

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.

Summary

Tensilica has invested heavily in its Xtensa Xplorer design environment to make your task—customizing a processor for your exact application—as quick and friendly as possible. Some designers find that the base configuration is sufficient for their design. Others add pre-built options using the simple check boxes. And some designers write their own TIE for maximum performance.

Whatever way you want to work, Tensilica provides proven tools to make you more productive.

Tensilica, Inc.

3255-6 Scott Blvd., Santa Clara, CA 95054-3013, USA
Tel: 1-408-986-8000 Fax: 408-986-8919 Website: www.tensilica.com

©March 2011 Tensilica and Xtensa are registered trademarks of Tensilica, Inc.
The Tensilica logo, Xenergy, Xplorer and TurboXim are trademarks of Tensilica, Inc. All other trademarks are the property of their respective owners.

