



xtensa LX2

PRODUCT BRIEF

FEATURES

- Highly efficient, small, low-power base 32-bit modern architecture
 - Configurable over a wide range of options - get just what you need
 - Configure processor as a multi-issue VLIW using FLIX™ architecture
- Extend processor with application-specific instructions, execution units, and register files
- Unlimited I/O bandwidth with designer-defined FIFO, GPIO, and Lookup interfaces
- Automated fine-grained clock gating for ultra-low power
- Designer-selectable 5- or 7-stage pipeline depth
- Automatic synthesis of processor configurations and extensions with XPRES™ Compiler
- Local memories configurable up to 4MB with the option for parity or ECC
- Modelessly intermix 24-bit base ISA instructions with 16-bit narrow instructions and VLIW instructions

BENEFITS

- Implement hardware architecture equivalent to RTL-based hardware design, with dramatically faster design time and much lower verification effort
- High bandwidth data flow through processor with flexible I/O interfaces that bypass the system bus
- Quickly and easily scale hardware architecture by simply using more task-customized processors
- Lower verification effort with pre-verified, correct-by-construction RTL generation
- Post-silicon programmability
- Higher code density due to 24-bit ISA leads to savings in memory area

Xtensa LX2 Configurable Processor Core

Tensilica's Xtensa® LX2 processor core has two essential unique features: **configurability** and **extensibility**. These features enable Xtensa processors to be used as an application-specific processor by software developers and as an RTL alternative by hardware designers. Ideal for handling traditional SOC embedded processor control tasks as well as compute-intensive datapath hardware tasks, the Xtensa LX2 processor is the new basic building block for complex SOC design.

Configurability means that the designer is offered a menu of checkbox and drop-down menu options. For example, the designer can choose to have execution units such as a 16x16 multiplier, a floating-point unit, a barrel shifter, et cetera. Additionally, the designer can choose the width and number of local instruction and data interfaces and choose from optional data paths such as the HiFi 2 Audio Engine or the Vectra™ LX general-purpose DSP engine.

Extensibility enables the designer to add multi-cycle execution units, registers, register files, FIFO interfaces, general purpose IO (GPIO) pins, single-instruction multiple-data functional

units, or even make the basic RISC pipeline into a multi-issue VLIW (very long instruction word) processor. All this is obtained using the TIE methodology in which the designer only has to specify and verify the functional behavior of the new data path and the RTL is automatically generated. To a software developer, this means that the processor now has many more instructions and register files that the compiler can automatically use, the debugger can debug, and the instruction-set simulator can simulate. To hardware designers, this means that they can implement the data path elements of the RTL blocks in the processor pipeline and the control FSM (finite state machine) as software running on the processor.

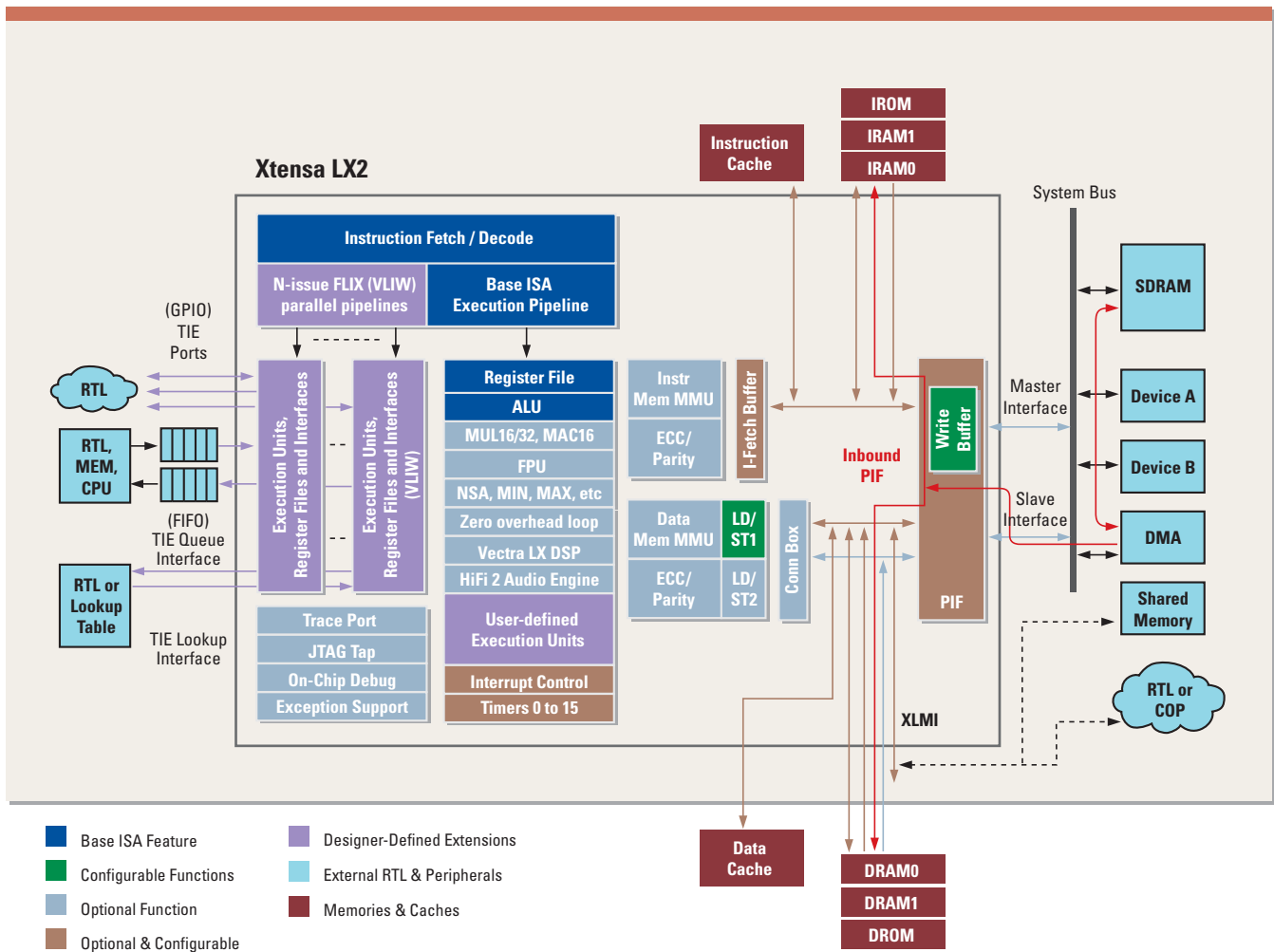


Figure 1: The Xtensa LX2 architecture starts with base ISA features common to all Xtensa LX2 cores. SOC developers can configure the Xtensa LX2 processor from menus of predefined options and add functional units of their own design to optimize algorithm performance, achieving designs equivalent to hand-coded custom logic blocks in a fraction of the time.

Xtensa LX2 as		
	Conventional CPU or DSP	RTL Alternative
Configurability	Configure your processor to fit your application. Get the options you want and not the ones you don't want	Choose from a menu of common, pre-optimized data path elements like multipliers and shifters
Extensibility	Add application-specific instructions to accelerate the hot spots in your application	Add multi-cycle execution units, registers, register files, and SIMD units to create the same data path you would in RTL
Designer-defined I/O interfaces	Use TIE Ports (GPIOs) and Queues (FIFO interfaces) to avoid the bottlenecks of the system bus	Interface to other RTL blocks and processors using direct wires and FIFOs, as you would if you were using RTL
Lower power	Use application-specific extensions to create a higher performance processor without increasing frequency and power	Fine grained clock gating is automatically generated by the Xtensa Processor Generator. This leads to higher power savings than with EDA-generated clock gating of manually written RTL because clock nets are automatically gated off cycle-by-cycle under program flow execution. No risk of introducing bugs while adding clock gating
Lower verification effort	Automatic pre-verified RTL generation, including control logic, bypass logic, and data path elements	Only have to verify functional specification of custom instructions and execution units. Significantly lower verification effort than RTL
Flexibility	Extending processor gives headroom to map more tasks as requirements and standards change, unlike fixed processors that rely on increasing frequency (MHz) to increase capability	Programmability of processor means that multiple applications can be mapped to the same SOC, software can be updated as algorithms change, and bugs can be fixed post-silicon
Faster time to market	Spend less time optimizing software or, on the backend, trying to increase frequency and, instead, just accelerate the application using designer-defined instructions	Lower verification effort and easy scalability by adding more task-optimized processors.
Smaller core area and memory area	Base processor configuration is less than 20K gates. Also, 24-bit ISA with 16-bit narrow encodings means higher code density than conventional RISC and DSP cores and, thus, smaller memory area.	Create optimized task engines with little or no area overhead for the processor.

Post-RISC Processor Core

The Xtensa LX2 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.

The Xtensa LX2 architecture features distinct features and advantages over traditional fixed processors, some of which are:

- **Base instruction set compatibility:** 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.

- **Smaller code size:** The Xtensa LX2 can modelessly issue 24-bit and 16-bit instructions, leading to 25-50% better code density and, therefore, smaller memories than mixed 32- and 16-bit architectures. Since memories typically dominate SOC area, this code density advantage translates into significant SOC area savings.
- **Powerful base ISA:** The Xtensa ISA also provides (a) powerful compare-and-branch instructions and zero-overhead loops, which are very useful for the compiler to generate tight, optimized loops, and (b) bit manipulations including funnel shifts and field-extract operations that are very useful for applications such as networking that process the fields in packet headers and perform rule-based checks.
- **Extendable ISA:** One of the fundamental technology innovations in the Xtensa processor is the ability to easily and seamlessly add new instructions and the associated C data types, along with the software tool chain support and the hardware data path to the processor. The scope of the instructions that can be added is general enough to enable the base RISC Xtensa LX2 processor to become an 8-way SIMD (single instruction, multiple data) general purpose DSP engine, a 3-instruction issue high-performance processor, or a small, low power cache-less controller. For example, a designer can add multi-cycle execution units, registers, register files, general purpose IO pins (Ports), and FIFO interfaces (Queues). The specification of this new data path and associated instructions and C data types is done in the Tensilica Instruction Extension (TIE) language, which is explained in more detail in a later section.
- **Multi-issue VLIW technology:** The Xtensa LX2 processor core features Tensilica's powerful FLIX technology, which allows the designer to configure the processor as a multi-issue VLIW processor. The Xtensa C/C++ Compiler (XCC) automatically extracts parallelism from C/C++ code and bundles multiple operations into FLIX (VLIW) instructions. In this way, a 3-issue Xtensa processor configuration running at 300Mhz can deliver performance up to the equivalent of a 900Mhz processor. Additionally, the compiler can bundle the branch and load/store instructions in parallel with compute instructions into VLIW instructions to gain a performance boost over straight-line code. This feature can be used selectively when needed, and the FLIX instructions are modelessly intermixed with the standard 16- and 24-bit instructions to avoid code bloat.
- **Configurable local and system interfaces:** The designer has flexibility to select the number and width of the local and system interfaces on the Xtensa LX2 processor. An Xtensa LX2 processor can have up to two local instruction and data RAMs and ROMs, instruction and data caches, and a single-cycle access general-purpose interface called XLMI. The widths of these local interfaces can be set to 32 bits, 64 bits, or 128 bits, independent of the PIF system interface that can also be set to any of these widths. This allows the designer to design a flexible system and memory architecture around the Xtensa LX2 processor.
- **Flexible designer-defined IO interfaces:** The designer can specify new interfaces to the data path in the processor that can be used to interface with other RTL and processor blocks in the SOC. These interfaces – Ports and Queues – are instantiations of general purpose I/O pins and FIFO interfaces on the processor that can be accessed directly by operations/ instructions without using load/store instructions.
- **Automatically generated, pre-verified processor RTL and software tool chain:** The designer-defined extensions and configuration options selected by the designer are taken as input by the Xtensa Processor Generator to automatically generate pre-verified RTL for the processor implementation, along with the entire software tool chain including compilers, debuggers, and simulators (cycle-accurate and fast functional). The designer can thus focus on application development instead of focusing on how to create an application-specific processor or how to create a complete software tool chain to support modifications they make to the processor.

Xtensa LX2 as a RTL Alternative

RTL verification has become the most resource and time expensive aspect of SOC design. The Xtensa LX2 processor offers this and several other distinct advantages as an alternative to using hardwired RTL blocks in complex SOCs. Some of these advantages are:

- **Lower verification effort and time:** Designing hardwired RTL blocks has become more about verification than about design. Design teams typically spend twice the number of resources and person months on verification than on design. Design changes made late in the project cycle are often limited by the verification effort. Furthermore, whereas 90% of the area of hardwired RTL blocks lies in the data path and only 10% in the control logic, most of the bugs are found (perhaps 90%) in the control logic. The ability to extend the Xtensa LX2 processor using TIE enables designers to create the same data path inside the Xtensa processor as they would in a hardwired RTL block. Yet no control FSM need be generated and verified by the designer; instead the control logic is expressed as software – instructions that execute on the processor. It is easier to verify TIE extensions made to the Xtensa LX2 processor than it is to verify the corresponding RTL data path, since only the input-output relationship or functional behavior of the operations specified in TIE have to be verified. The TIE Compiler and Xtensa Processor Generator take care of converting the TIE specification into data path elements in the processor pipeline and implementing the control, decode, and bypass logic in the processor control units. Thus, the designer does not have to verify RTL anymore, just the TIE behavioral specification.

- **Lower power because of finer clock gating:** Since the Xtensa LX2 processor is automatically generated by the Xtensa Processor Generator, the Generator statically analyses pipeline activity and does aggressive clock gating on each functional unit and flip-flop in the processor RTL. To do an equivalent amount of clock gating in hardwired RTL blocks would be impractical due to the time and resources it would take and more importantly the burden it would add to the verification team on verifying that the clock gating is done correctly.
- **Reuse of the same hardware for multiple tasks:** Complex SOCs consist of millions of gates of logic and are designed to perform multiple tasks. Often these multiple tasks do not need to be performed at the same time. This provides an opportunity for multiple tasks to share the same hardware units. Processors are particularly amenable to enabling this sort of sharing. Designers can specify in TIE a data path that consists of a set of execution units that can be used by the multiple tasks and use the programmability of the processor to determine which tasks are executed. For example, a designer can build a video engine that can be used to implement a range of video codecs like H.264, MPEG-4, VC-1, etc.

- **Flexibility to upgrade algorithms post-silicon:** Using a task-optimized Xtensa LX2 processor to implement an algorithm lets the designer implement modifications, enhancements, and tweaks to the algorithm after the SOC has taped out. For example, half-toning algorithms in printers are a subject for continuous research and, therefore, are good candidates for implementation using an Xtensa LX2 processor.

Rapid Design Exploration

The Xtensa Processor Generator and the Xtensa Xplorer™ development environment assist SOC designers in creating tailored, application-specific embedded processors quickly and reliably.

Designers with existing application software code 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 I/O ports of nearly unlimited complexity to stream data into the new execution unit datapaths, add new instructions to the processor, and write simple C programs to test the new processor - all from within the integrated Xtensa Xplorer design environment.

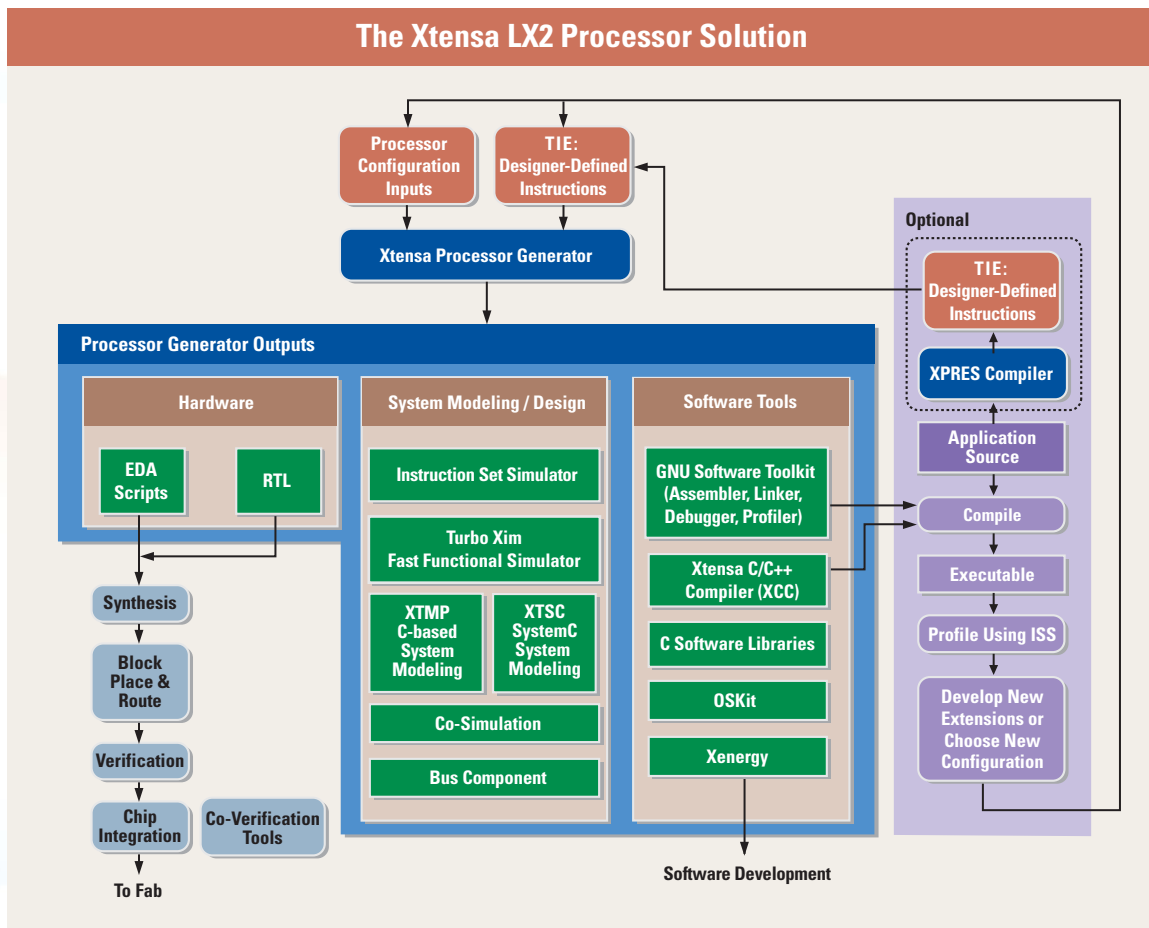


Figure 2: Every instance of the Xtensa LX2 processor includes comprehensive hardware, software and system modeling deliverables - all automatically generated by the Tensilica processor generator.

Highly Configurable Base Processor Feature Set

Designers can quickly configure many Xtensa LX2 core options using a check-box GUI in the Xtensa Xplorer design environment. These options include:

Execution Unit and ISA Options

- Multipliers, 32- or 16-bit
- DSP engines
 - Single 16-bit MAC
 - 4-MAC SIMD Vectra™ LX DSP engine
 - HiFi 2 Audio engine
- 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
- Choice of 1 or 2 general purpose load/store units
 - Enables classic X-Y style DSP configurations
- Big-Endian/Little-Endian byte ordering
- On-chip debug port
- Trace port signals
- Up to 32 interrupts
- Designer-defined queues/ports
 - Up to 1024 logical ports
 - Up to 1024 pins per port

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 4Mbytes each
- Optional parity or ECC for all local memories

Design Support

- Pipeline-modeled, cycle-accurate instruction set simulator
- High-speed (40X-80X) instruction-accurate simulator for software development
- Xtensa Modeling Protocol (XTMP) for multi-CPU simulation
- 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
- Hardware/software co-verification model for Mentor Graphics’ Seamless

Processor Extensions

Accelerating Processor Performance to Custom Logic Design Levels

The TIE language is used to describe new instructions, new registers and execution units, and new I/O ports that are then automatically added to the Xtensa LX2 processor. TIE is a Verilog-like language used to describe 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 LX2 processor and the complete tool chain that incorporates the new TIE instructions.

FLIX (VLIW) Architecture

Highly Parallel Implementations

The Xtensa LX2 implements Tensilica’s FLIX architecture. FLIX is a configuration option that makes the Xtensa LX2 configuration into a VLIW processor and allows designer-defined instructions to consist of multiple, independent operations bundled into a compact 32-bit or 64-bit instruction word. Wide 32/64 bit FLIX instruction formats are seamlessly and modelessly intermixed with the base Xtensa ISA’s existing 16/24 bit instructions – there is no mode switch penalty to utilize a FLIX instruction.

The FLIX architecture allows the implementation of highly parallel processors with a range of 2 to 15 parallel execution units. Thus Xtensa LX2 can deliver the ultra-high performance characteristic of specialty ultra-wide instruction word processors, without the negative code size implications typically found in such VLIW or ULIW solutions. In fact, Xtensa LX2 processors with FLIX can often deliver higher performance and smaller code size at the same time. This performance increase comes with very little overhead – adding only 2,000 gates to the size of the processor for instruction decode and control.

GPIO (Ports) and FIFO Interfaces (Queues)

Unlimited I/O Bandwidth

The Xtensa LX2 processor brings another fundamental breakthrough in embedded processor design – designer-defined GPIO pins (called Ports) and FIFO interfaces (called Queues).

TIE Ports are created using simple one-line declarations [see Figure 3] in a TIE file. These new Ports are used as direct connections to other logic within an SOC or to other Xtensa LX2 processors.

A designer also can specify FIFO interfaces on an Xtensa LX2 processor by declaring TIE Queues. TIE Queues enable Xtensa LX2 processors to have virtually unlimited I/O bandwidth. TIE input Queues present a familiar pop/empty/data interface to the external logic, while TIE output Queues present a similar push/full/data interface.

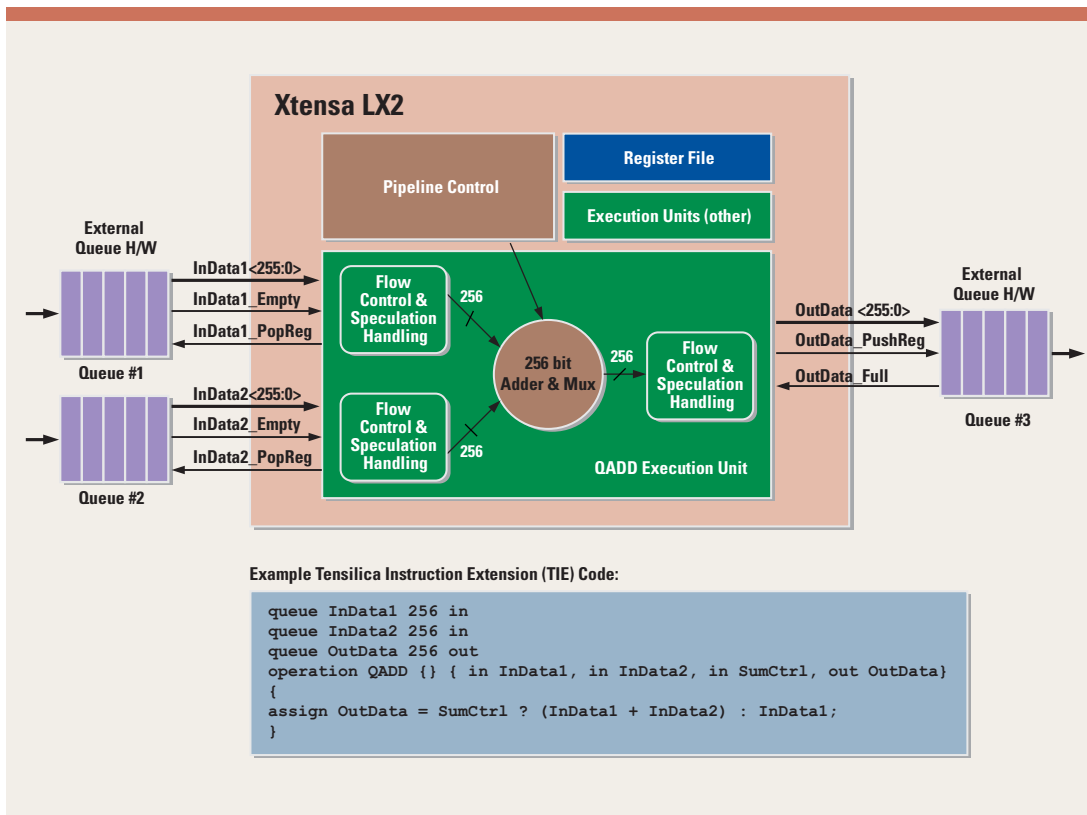


Figure 3: An example of designer-defined FIFO interfaces (TIE Queues) and an execution unit that performs 2 operand inputs, one operand output, and a 256-bit Add on every clock cycle.

All interactions with the Xtensa LX2 processor pipeline, including pipeline control and speculation handling, are automatically implemented by the processor generator. As figure 3 illustrates, the SOC designer need only specify a handful of commands to create a high-bandwidth set of I/O Queues and execution units that operate on those Queues.

A maximum of 1024 logical Ports, each consisting of up to 1024 pins, can be added to each configuration of an Xtensa LX2 processor. TIE Ports and Queues can be utilized every clock cycle without the use of load/store instructions – providing virtually unlimited I/O bandwidth for an embedded Xtensa LX2 core.

Lookup interfaces

Unlimited local memories

TIE Lookup interfaces enable designers to connect RAMs or an external device to the Xtensa LX2 processors that can be accessed directly from the data path without using load and store instructions. These interfaces are useful for connecting RAMs for doing table lookups, for example in networking applications, or for connecting long latency hardware computation units. An example is shown in Figure 4.

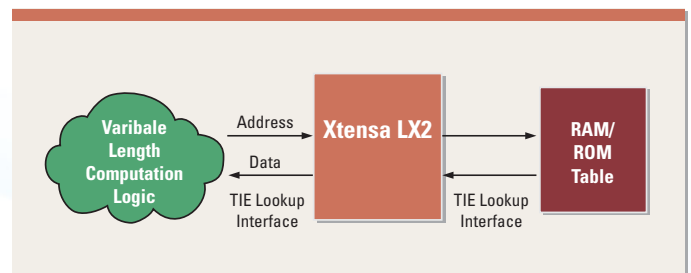


Figure 4: Examples of connecting the Xtensa LX2 processor to external RAM/ROMs and variable-length computation logic blocks using designer-defined Lookup interfaces.

Performance and I/O of Hard-wired Logic, Without Hand-coded State Machines

The combination of TIE Ports and Queues with parallel FLIX execution units implemented in an Xtensa LX2 processor allows SOC designers to achieve virtually the same levels of inter-block I/O bandwidth and intra-block computational parallelism as with hard-wired logic designed with traditional RTL design methodologies. But unlike RTL-based design, configured and extended Xtensa LX2 processors are pre-verified by the Xtensa Processor Generator, and do not require hard-

wired implementation of complex state machines. Instead of state machines, the complex datapaths added to Xtensa LX2 cores are sequenced/controlled by the instruction stream of the Xtensa LX2 processor. That means the “control logic” is fully software programmable and debuggable - reducing verification time and risk for the entire SOC.

7-stage Pipeline for Better Interface to On-Chip Memories

To address the growing speed disparity between standard cell logic and memories (memory access speeds have not scaled as well as logic in the migration from 180 nm to 130 nm and now 90 nm and 65nm processes), the Xtensa LX2 processor features a configurable pipeline. Designers can select a configuration option for a 7-stage pipeline that adds two additional clock cycles for memory access if required by the application. While the Xtensa LX2 processor’s standard 5-stage pipeline is very efficient for many applications, designers employing large local memories or specialized low-power memories with longer access times will find advantages in moving to a longer pipeline, resulting in a higher system clock frequency.

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 into 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.

Tensilica’s Xenergy™ energy estimation tool can be used by designers to optimize both the Xtensa LX2 configuration and TIE instructions plus it can be used to tune the software application for energy.

Clock gating is a very effective power reduction technique that shuts down power 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 LX2 processor including functions conceived of and created by the designer. This automation gives the Xtensa LX2 a significant advantage over RTL design, where manual, error-prone post-layout tuning of clock circuits is often required.

Accessing local memories is one of the highest power-consuming activities an embedded processor must perform. Tensilica has designed the Xtensa LX2 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 LX2 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 LX2 processor is designed to use power very efficiently. The minimum configuration of the Xtensa LX2 processor dissipates less than 48 $\mu\text{W}/\text{MHz}$ in a 90nm, overdrive (GT), worst-case 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 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 LX2 processors; it will correct single-bit errors and detect double-bit errors. Error correction is extremely important in storage, networking, and automotive applications where a single-bit error can be catastrophic.

Comprehensive and Automated Software Tool Support

Every Xtensa LX2 processor is automatically generated with a complete set of software development and modeling tools tailored to the exact Xtensa LX2 configuration. 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.

The Xtensa LX2 Tools suite includes the high-performance vectorizing Xtensa C/C++ compiler; the XPRES Compiler for automatically generating processor configurations from standard C/C++ code; a GNU-based assembler/linker/debugger/and profiler; the pipeline-modeled, cycle-accurate Xtensa Instruction Set Simulator; the fast functional TurboXim™ simulator, the Xtensa Modeling Protocol system simulation API, and the Xtensa SystemC API. And all of the Xtensa development tools are tightly integrated within the Xtensa Explorer development environment [see Figure 5], the only tool that integrates software development, processor optimization and multiple-processor SOC architecture tools into one common design environment.

See the Processor Developer’s Toolkit product brief and the Software Developer’s Toolkit product brief for more information on the Xtensa tools.

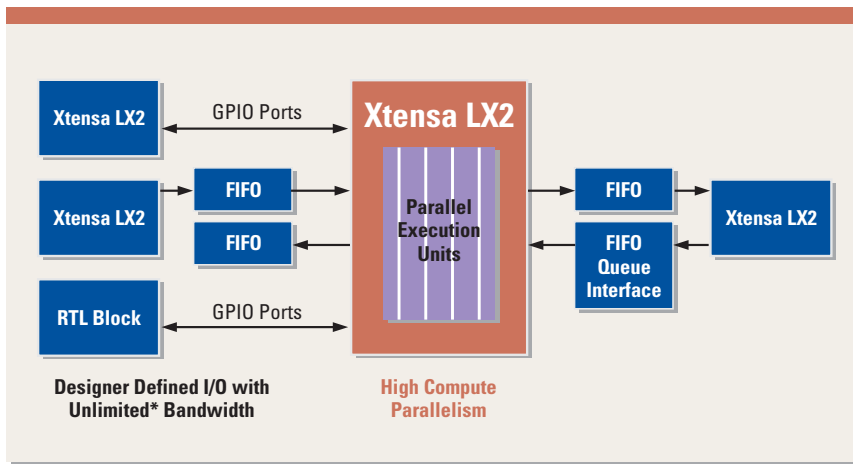


Figure 5: Illustration of high-performance streaming data design using multiple designer-defined TIE Ports (GPIO) and Queues (FIFO Interfaces).

Pipeline Viewer shows instruction flow of disassembled code

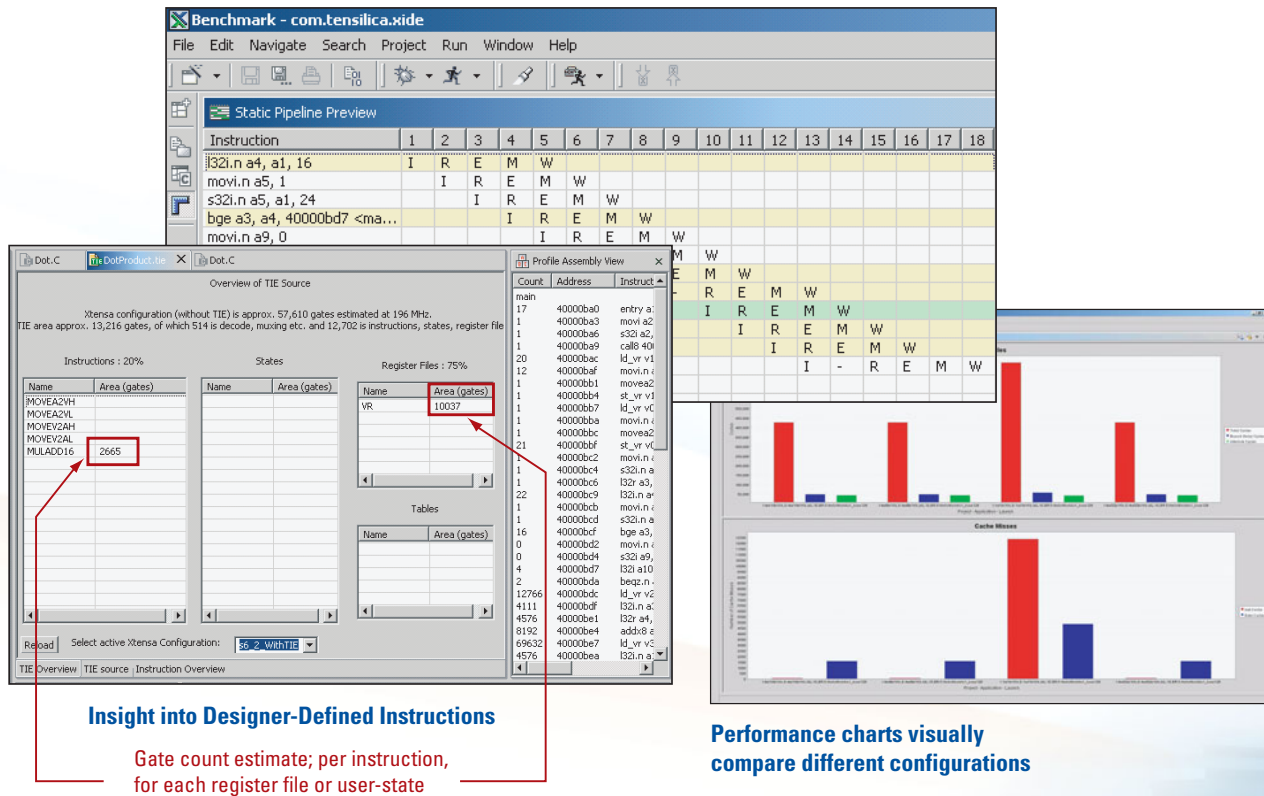


Figure 6: The Xtensa Xplorer development environment provides powerful visualization tools to assist in the configuration of an Xtensa LX2 processor.

Feature Summary

Tensilica's processor generator technology creates a complete RTL hardware description of a customized Xtensa LX2 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 multiplier
- 16-bit multiplier and MAC
- 4-way SIMD Vectra LX DSP engine
- HiFi2 Audio engine
- Floating point unit

Configurable Interface Options

- Designer-defined I/O ports up to 1M wires
- 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
- Optional second data Load/Store unit

Sea of Processors / Multiple-Processor design style support

- System modeling capability: optional Xtensa Modeling Protocol (XTMP) and Xtensa SystemC (XTSC) simulation environments
- 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
- Formal verification support for designer-defined functionality

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 simulator for software development
- Multiple-processor simulation with XTMP option
- Xtensa SystemC (XTSC) transaction-level modeling (TLM) support
- 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 LX2 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)
- Mentor Graphic's Nucleus Plus, Express Logic's ThreadX, Micrium Technologies' uC/OS-II, MontaVista Software's Linux Professional Edition, and Sophia Systems' μITRON operating systems

XPRES Compiler Support

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

Processor Architecture

- High-performance 32-bit RISC with designer-selectable 5-stage or 7-stage pipeline.
- Optional FLIX (VLIW) designer-defined, highly parallel instruction extensions. Parallelism of designer-defined execution units: 2-wide to 15-wide.

Instruction Set

- Xtensa ISA with compact 16-bit and 24-bit base instruction set and optional designer-defined 32-bit or 64-bit FLIX (VLIW) instructions.

	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, Area-optimized netlist. Typical operating conditions.	90 nm GT process, Area-optimized netlist. Typical operating conditions.
RTOS-ready base configuration, 5-stage	0.26mm ²	350 MHz	600 MHz	76 μ W/MHz	94 μ W/MHz
Performance optimized task engine, 7-stage, no PIF	0.26mm ²	350 MHz	650 MHz	47 μ W/MHz	59 μ W/MHz
Minimum configuration	0.16mm ²	350 MHz	600 MHz	38 μ W/MHz	48 μ W/MHz



Tensilica, Inc.

3255-6 Scott Boulevard, Santa Clara, CA 954054-3013, USA
 Tel. 408-986-8000 • Fax. 408-986-8919 • Website: www.tensilica.com

© 2007, Tensilica, Inc. Tensilica and Xtensa are registered trademarks of Tensilica, Inc. The following terms are trademarks of Tensilica, Inc.: FLIX, OSKit, Sea of Processors, TurboXim, Vectra, Xenergy, Xplorer, and XPRES. All other registered trademarks are the property of their respective owners.