L31

Product brief

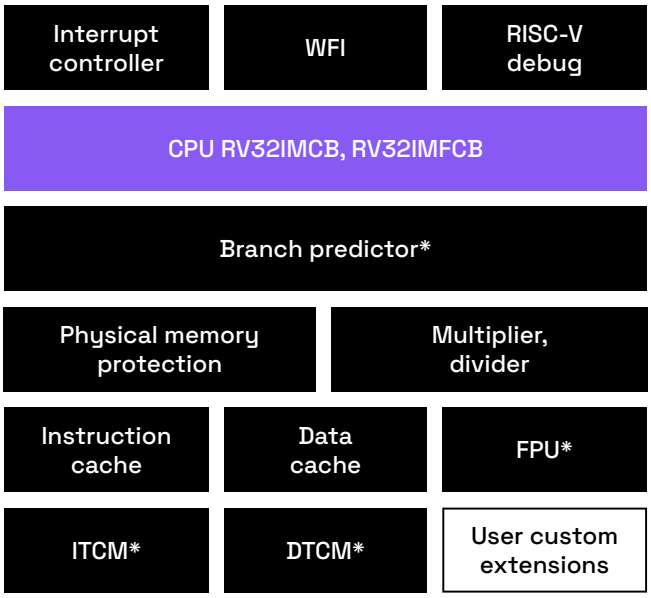
Codasip

Overview

The Codasip® L31 core is a low-power, embedded RISC-V core in the 3 Series. Providing an ideal balance between performance and power consumption, it enables a wide range of embedded applications.

The L31 with FPU option enables innovation in applications that require hardware floating-point capabilities such as signal processing.

Codasip L31



*optional

The L31 core can be licensed either as an off-the-shelf (RTL and software toolchain) or fully customizable processor (CodAL).

Benefits

Highly configurable

- Flexible design for use in a broad range of markets

Designed for differentiation

- RISC-V core designed in CodAL™ and fully customizable if the CodAL architecture source is licensed

Optimized for AI/ML workloads

- With TensorFlow Lite support for Microcontrollers

Applications

Video and image processing
Smart sensors
IoT devices

Deliverables

- Hardware Development Kit
- Software Development Kit
- FPGA evaluation platform
- Basic set of applications
- Documentation

Specifications

Core

- 3-stage, in-order pipeline
- RV32IMCB & RV32IMFCB ISA
- 32-bit registers

Branch predictor

- Optional
- Increased single-thread performance

Power

- Clock gating
- Configurable reset
- WFI

Caches

- L1 instruction and data
- Configurable L1 cache size, associativity and cache line size
- Optional

Multiplier, divider

- Multiplier implementation: Parallel
- Divider implementation: Sequential

Debug

- Standard RISC-V debug
- 4 JTAG pins
- 2-8 breakpoints & watchpoints

Tightly coupled memory

- Instruction and data TCMs
- Customizable size up to 2MB
- AHB-Lite TCM secondary port

Memory protection

- Optional MPU with 2/4/8/16 regions
- Physical memory attributes with 16 regions
- Machine and User privilege modes

Interrupts

- Interrupt controller
- Standard RISC-V CLINT implementation
- Up to 128 interrupts
- NMI

Interfaces

- 2x AHB-Lite

FPU

- Available as option

Performance

- Performance monitors

Customization

L31 is designed using Codasip Studio™

- Easy customization in CodAL

Add new instructions or data types accelerating your algorithm

- Studio automatically generates the compiler aware of such extensions

Connect your existing accelerator directly to the processor

- Studio generates the RTL with the new interface