

Processor Functional Description

This hardware component is a lm32 (lattice mico32) processor core.

This ISS uses the [ISS2 API](#) and can be wrapped in a CABA or TLM-T [Wrapper](#).

- [gdb server](#) support is under development.

It implements all instructions defined in the lm32 architecture specification.

Component definition & implementation

- [source:trunk/soclib/soclib/iss/lm32/metadata/lm32.sd?](#)
- [source:trunk/soclib/soclib/iss/lm32/include/lm32.h?](#)
- [source:trunk/soclib/soclib/iss/lm32/src/lm32.cpp?](#)
- [source:trunk/soclib/soclib/iss/lm32/src/lm32_isa.cpp?](#)
- [source:trunk/soclib/soclib/iss/lm32/src/lm32_load_store.cpp?](#)
- [source:trunk/soclib/soclib/iss/lm32/src/lm32_debug.cpp?](#)

Interrupts

LM32 architecture supports up to 32 external interrupt lines. This ISS implements all these interrupt lines by default.

Ports

None, it is to the wrapper to provide them.

Notes

MMU support

The lm32 provided by lattice does not have an mmu.

Compiling programs for lm32 with SoCLib

Before compiling a program for the lm32 with the SoCLib framework you will need to define some system variables (usually on the `~/soclib/soft_compilers.conf`) needed to find the lm32 compiler. Below you have an example:

```
sparc_CC_PREFIX = lm32-elf-  
sparc_CFLAGS = -O2 -g -mmultiply-enabled -mdivide-enabled -msign-extend-enabled -mbarrel-shift-e  
sparc_LDFLAGS = -nostdlib
```