Processor Functional Description

This hardware component is a Mips-32 processor core.

This ISS uses the ISS2 API and can be wrapped in a CABA or TLM-T Wrapper.

It implements all instructions defined in the MIPS32 architecture specification, with the following limitations:

- The floating point instructions are supported, FPU exception detection is partially implemented
- The Mips virtual memory instructions are not supported. The MMU is implemented as an external TLB (SoCLib generic MMU) in the VciVcacheWrapper? component.

Both little-endian and big-endian implementations are available.

Component definition & implementation

- source:trunk/soclib/soclib/iss/mips32/metadata/mips32.sd?
- source:trunk/soclib/soclib/iss/mips32/include/mips32.h?
- source:trunk/soclib/soclib/iss/mips32/src/mips32.cpp?
- source:trunk/soclib/soclib/iss/mips32/src/mips32_cp0.cpp?
- source:trunk/soclib/soclib/iss/mips32/src/mips32_fpu.cpp?
- source:trunk/soclib/soclib/iss/mips32/src/mips32_hazard.cpp?
- source:trunk/soclib/soclib/iss/mips32/src/mips32_instructions.cpp?
- source:trunk/soclib/soclib/iss/mips32/src/mips32_load_store.cpp?
- source:trunk/soclib/soclib/iss/mips32/src/mips32_run.cpp?
- source:trunk/soclib/soclib/iss/mips32/src/mips32_special.cpp?
- source:trunk/soclib/soclib/iss/mips32/src/mips32_special2cpp
- source:trunk/soclib/soclib/iss/mips32/src/mips32_special3.cpp?

Template parameters

This component has no template parameters.

Interrupts

Mips defines 6 interrupts lines. The handling and prioritization of the interrupts is deferred to software.

Ports

None, it is to the wrapper to provide them.

Debug and traces

When built with SOCLIB_MODULE_DEBUG defined (for example by calling soclib-cc with "-b common:mips32 -b common:mips32el"), some debug messages are available, controlled with the ISS2 set_debug_mask() method. By default all messages are enabled; see source:trunk/soclib/soclib/iss/mips32/include/mips32.h? for possible values.