

General rules for the SoCLib hardware components

1. Naming conventions
 1. Namespaces
 2. Variables
2. VCI initiators and targets indexation
3. Endianness

Naming conventions

Namespaces

SystemC being build upon C++, we use the C++ namespace constructs, to create unambiguous names. SoCLib defines the following namespaces:

- `soclib`
- `soclib::common`
- `soclib::caba`
- `soclib::tlmt`

Variables

The following conventions have been defined :

- All component port names should be prefixed with `p_`
- All component register names should be prefixed with `r_`
- All component member variable names should be prefixed with `m_`

VCI initiators and targets indexation

In a VCI-based architecture, all initiators and targets must be indexed. Initiators and targets have different address spaces.

- The target index is used by interconnect components to route the VCI command packets : the target index is decoded from VCI ADDRESS MSBs.
- The initiator index is used by the interconnect components to route the VCI response packets : the initiator index is the VCI RSRCID.

Indexes can be :

- a simple scalar index, in case of a *flat* interconnect.
- a composite index, in case of a *clusterised* architecture, using a two-level (or more) hierarchical interconnect.

`include/common/int_tab.h` defines an utility class storing a list of indexes.

All indexes must be declared as `IntTab?s`.

Endianness

All SoCLib targets components are little-endian. In case of write, the bytes positions are fully controlled by the VCI BE bits :

- LSBs of the VCI ADDRESS are ignored, and the VCI ADDRESS is only used to select a VCI cell (a word in memory).
- Bytes are selected by the VCI BE field, and the BE[0] bit is always associated to the Byte 0 of the VCI WDATA field (ie WDATA[7:0]).