

General rules for the SoCLib hardware components

1. A) Naming conventions
2. B) Component indexation
3. C) Endianness

A) 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_`

B) Component 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 : those components decode the VCI address MSB bits to get the target index and route the VCI command packets.
- The initiator index is used by the interconnect components to route the VCI response packets : the initiator index is the VCI SRCID value.

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 hierarchical interconnect. Each component is identified by two scalars: `cluster_index`, `local_index`.

The `common/int_tab.h` file defines an utility class storing a list of indexes : All indexes must be declared as `IntTabs`.

C) Endianness

All SoCLib targets components respect the little-endianness convention. In case of write, the bytes transfers are fully controlled by the VCI BE bits :

- If the VCI word is larger than one byte, the LSB bits of the VCI ADDRESS are ignored, and the VCI ADDRESS is only used to address the selected VCI word.
- the selected bytes are specified the VCI BE field, and the BE0 bit is always associated to the Byte 0 of the VCI WDATA field.
- The BE1 bit is associated