

VciLocalCrossbar

1) Functional Description

This hardware component can be used as a local interconnect in a two-levels hierarchical interconnect to interconnect a limited number of VCI initiators and targets in a local sub-system. The associated sub-system is identified by a cluster index. It contains two independant crossbars for VCI commands and VCI responses.

- The number of VCI initiators is a parameter that should not be larger than 4.
- The number of VCI targets is a parameter that should not be larger than 4.

The VciLocalCrossbar component has two dedicated VCI ports (initiator and target) to connect the local subsystem to the global VCI interconnect.

When several initiators try to reach the same target, the arbitration policy is round-robin.

As any VCI advanced compliant interconnect, this component uses the MSB bits of the VCI ADDRESS field to route the command packets to the proper target, thanks to a routing table, implemented as a ROM. This routing table is build by the constructor from the informations stored in the [mapping table](#). It uses the VCI RSRCID field to route the response packet to the initiator.

2) Component definition & usage

source:trunk/soclib/module/network_component/vci_local_crossbar/caba/metadata/vci_local_crossbar.sd

See [SoclibCc/VciParameters](#)

```
Uses( 'vci_local_crossbar', **vci_parameters )
```

3) CABA Implementation

CABA sources

- interface :
[source:trunk/soclib/soclib/module/network_component/vci_local_crossbar/caba/source/include/vci_local_crossbar.h?](#)
- implementation :
[source:trunk/soclib/soclib/module/network_component/vci_local_crossbar/caba/source/src/vci_local_crossbar.cpp?](#)

CABA Constructor parameters

```
VciLocalCrossbar(  
    sc_module_name, // instance name  
    const soclib::common::MappingTable &mt, // mapping table  
    const size_t cluster_index, // global cluster index  
    const soclib::dd::common::IntTab &cluster_index, // global cluster index  
    const size_t nb_initiator, // number of VCI initiators  
    const size_t nb_target, // number of VCI targets  
    const size_t default_target ); // default target index
```

CABA Ports

- **p_resetrn** : Global system reset
- **p_clk** : Global system clock
- **p_to_initiator[i]** : Ports to VCI initiators
- **p_to_target[i]** : Ports to VCI targets
- **p_initiator_to_up** : Initiator VCI port to micro-network
- **p_target_to_up** : Target VCI port to micro-network

4) TLM-DT implementation

TLM-DT sources

- interface :
[source:trunk/soclib/soclib/module/network_component/vci_local_crossbar/tlmdt/source/include/vci_local_crossbar.h](#)
- implementation :
[source:trunk/soclib/soclib/module/network_component/vci_local_crossbar/tlmdt/source/src/vci_local_crossbar.cpp?](#)

TLM-DT Constructor parameters

```
VciLocalCrossbar(  
    sc_module_name name, // instance name  
    const soclib::common::MappingTable &mt, // mapping table  
    const soclib::common::IntTab &initiator_index, // global initiator index  
    const soclib::common::IntTab &target_index, // global target index  
    size_t nb_initiator, // number of VCI initiators  
    size_t nb_target ); // number of VCI targets
```

TLM-DT Ports

- **p_to_initiator[i]** : ports to VCI initiators
- **p_to_target[i]** : ports to VCI targets
- **p_initiator_to_up** : VCI initiator port to micro-network
- **p_target_to_up** : VCI target port to micro-network