

# VciVgmn Functional Description



This hardware component is a generic micro-network respecting the VCI advanced protocol. It supports several simultaneous transaction, and behaves as two fully independent packet switched network for VCI commands and VCI responses.

When several initiators try to reach the same target, the arbitration policy is round-robin. It gives the system designer a generic "communication black-box" with a parametrized number of VCI initiator ports (NB\_INITIATOR), and a parameterized number of VCI target ports (NB\_TARGET). It can be used to build a "flat" interconnect, where all VCI initiators and targets are identified by a single index:

- The VCI targets must be indexed from 0 to (NB\_TARGET - 1).
- The VCI initiators must be indexed from 0 to (NB\_INITIATOR - 1).

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.

This component has two "structural" parameters, that can be used to fit the behaviour of a specific physical micro-network:

- The MIN\_LATENCY parameter is a number of cycles that defines the latency of an empty network.
- The FIFO\_DEPTH parameter can be increased to improve the saturation threshold.

## VciVgmn CABA Implementation

The caba implementation is in

- source:trunk/soclib/systemc/include/caba/interconnect/vci\_vgmn.h
- source:trunk/soclib/systemc/src/caba/interconnect/vci\_vgmn.cc

## Template parameters

```
template<typename vci_param>
```

## Constructor parameters

```
VciVgmn(  
    sc_module_name name,    // Component Name  
    const soclib::common::MappingTable &mt);    // MappingTable
```

## Ports

- sc\_in<bool> **p\_resetn** : Global system reset
- sc\_in<bool> **p\_clk** : Global system clock
- soclib::caba::VciTarget<vci\_param> **p\_from\_initiator[NB\_INITIATOR]** : Ports from VCI initiators

- `soclib::caba::VciInitiator<vci_param> p_to_target[NB_TARGET]` : Ports to VCI targets

## Implementation notes

Each micro network (Command and Response packets interconnects) is implemented within the same templated code. This code is instantiated twice, once for each network.



There is one `InputRouter` object per input port, it handles incoming packets and forwards them to the right `OutputPortQueue`.

There is one `OutputPortQueue` per output port, handling as many fifos as input ports, and one delay line. It handles putting packets on output port.