

# VciInitiatorTransactor

## 1) Functional Description

All VCI commands received on the CABA interface are registered in a buffer indexed by the TRDID field. Each entry in this Pending\_Transaction\_Buffer contains a transaction status, a transaction time, and enough space to store a complete VCI TLM-DT transaction payload. As for any TLM-DT initiator, the behavior() sc\_thread associated to the VciInitiatorTransactor is permanently running. It is descheduled twice per simulated cycle to drive the CK signal on the CABA interface. It implement the same look-ahead mechanism as any TLM-DT initiator to send NULL messages with a bounded period when there is no regular (Read or Write) VCI transactions. Regarding the VCI command, the behavior() thread read the command signals (cycle per cycle), on the CABA ports, and stores a complete transaction payload in the Pending\_Transaction\_Buffer. When the transaction is completed, the behavior() thread calls the nb\_transport\_fw() method on the TLM-DT port.



Regarding the VCI response, the interface nb\_transport\_bw() method updates the status of the proper entry in the Pending\_Transaction\_Buffer when a response is received on the TLM-DT port. This response is then transmitted (cycle per cycle) by the behaviour() thread on the CABA ports. If the VCI initiator can receive interrupts, the transactor provides an optional service of translation : When an IRQ is received on the TLM-DT IRQ port (by the interface nb\_transport\_fw() method) this information is registered, and the IRQ is transmitted on the IRQ CABA port by the behaviour() thread.

## 2) Implementation

### template parameters

This component have two template parameters, defining respectively the width of the CABA and TLM-DT VCI parameters.

```
template<typename vci_param_caba, typename vci_param_tlm_dt>
```

### sources

- interface :  
[source:trunk/soclib/soclib/module/network\\_component/vci\\_initiator\\_transactor/tlm\\_dt/source/include/vci\\_initiator\\_tra](#)
- implementation :  
[source:trunk/soclib/soclib/module/network\\_component/vci\\_initiator\\_transactor/tlm\\_dt/source/src/vci\\_initiator\\_transac](#)

### Constructor parameters

```
VciInitiatorTransactor(  
    sc_module_name name, // Instance name  
    size_t n_irq;        // optional : number of interruption ports (default = 0)
```

### Ports

- sc\_out<bool> **p\_resetrn** : hardware reset
- sc\_out<bool> **p\_clk** : clock
- sc\_out<bool> **\*p\_irq\_initiator** : IRQ initiator ports (from transactor to CABA initiator)

- VciTarget<vci\_param\_caba> **p\_vci\_target** : VCI target port (from CABA initiator to transactor)
- **p\_vci\_initiator** : VCI initiator port (from transactor to TLM-DT platform)
- **\*p\_irq\_target** : IRQ target ports (from TLM-DT platform to transactor)