VciTargetTransactor

1) Functional Description

All transactions received on the TLM-DT port are registered in a 2 dimensions buffer, indexed by both the VCI SRCID, and VCI TRDID fields. Each entry in this Pending-Transaction_Buffer contains a transaction status, a transaction time, and a pointer on the transaction payload. As any TLM-DT target, the VciTargetTransactor component has a purely reactive behaviour : the behavior() thread is running only when a command is received, and keeps running until the last response to the last pending transaction has been transmited. When it is running, it is descheduled twice per simulated cycle to drive the CK signal on the CABA interface. Regarding the VCI commands received on the TLMDT port, the nb_transport_fw() interface method register the transaction in the Pending_Transaction_Buffer. The behaviour() thread will then send (cycle per cycle) the VCI command on the CABA port. Regarding the VCI responses received on the CABA port, the behaviour() thread updates the payload (cycle per cycle). When the transaction is completed, the corresponding entry in the Pending_transaction_buffer is cleared, and the nd_transport_bw() method is called on the TLM-DT port.

🔟

If the VCI target can send interrupts, the transactor provides an optional service of translation : When the behaviour() thread is running (following a VCI command, or NULL message reception) it polls the CABA IRQ port, and calls the nb_transport_fw() method on the IRQ TLM-DT port when the CABA value is modified.

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_tlmdt>
```

sources

- interface :
  source:trunk/soclib/soclib/module/network_component/vci_target_transactor/tlmdt/source/include/vci_target_transactor
- implementation :
  source:trunk/soclib/soclib/module/network_component/vci_target_transactor/tlmdt/source/src/vci_target_transactor.c

Constructor parameters

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

Ports

- sc_out<bool> **p_resetn** : hardware reset
- sc_out<bool> **p_clk** : clock
- sc_in<bool> ***p_irq_target** : IRQ target ports (from CABA to transactor)
- VciInitiator<vci_param_caba> **p_vci_initiator** : VCI initiator port (from transactor to CABA)

- **p_vci_target** : VCI target port (from TLM-DT platform to transactor)
- ***p_irq_initiator** : IRQ initiator ports (from transactor to TLM-DT platform)